A KNOWLEDGE-BASED INTELLIGENT SYSTEM FOR SURFACE TEXTURE (VIRTUALSURF)

By

YAN WANG

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

Department of Computing & Engineering
The University of Huddersfield

July 2008

ACKNOWLEDGEMENTS

I would like to thank my supervisors for their inspiration and valuable contributions to my work. In particular sincere thanks are directed to my supervisor Professor Xiangqian (Jane) Jiang for her patience, guidance and professional support.

Special thanks are devoted to Prof. Paul Scott for his continued encouragement, guidance and invaluable suggestions during this work. Words really cannot express how grateful I am.

I would also like to thank the following people for their advice and technical assistance throughout the duration of this project:

Professor Liam Blunt
Dr. Leigh Brown
Dr. Paul Bill
Helene Pickles
All of my colleagues and friends in the Centre for Precision Technologies, The University of Huddersfield.

Finally I would like to thank my Family who have always been a source of great inspiration, motivation and calm as and when required. The encouragement and support from my parents and my fiancé Can has been invaluable and to them I give my sincere and heartfelt thanks.

ABSTRACT

The presented thesis documents the investigation and development of the mathematical foundations for a novel knowledge-based system for surface texture (VitualSurf system). This is the first time that this type of novel knowledge-based system has been tried on surface texture knowledge. It is important to realize that surface texture knowledge, based on new generation Geometrical Product Specification (GPS) system, are considered to be too theoretical, abstract, complex and over-elaborate. Also it is not easy for industry to understand and implement them efficiently in a short time.

The VirtualSurf has been developed to link surface function, specification through manufacture and verification, and provide a universal platform for engineers in industry, making it easier for them to understand and use the latest surface texture knowledge. The intelligent knowledge-base should be capable of incorporating knowledge from multiple sources (standards, books, experts, etc), adding new knowledge from these sources and still remain a coherent reliable system. In this research, an object-relationship data model is developed to represent surface texture knowledge. The object-relationship data model generalises the relational and object orientated data models. It has both the flexibility of structures for entities and also good mathematical foundations, based on category theory, that ensures the knowledge-base remains a coherent and reliable system as new knowledge is added.

This prototype system leaves much potential for further work. Based on the framework and data models developed in this thesis, the system will be developed into implemental software, either acting as a good training tool for new and less experienced engineers or further connecting with other analysis software, CAD software (design), surface instrument software (measurement) etc, and finally applied in production industries.

CONTENTS

LIST OF FIGURES

LIST OF TABLES

Chapter 1

Introduction

1.1 Motivation

A geometrical product is a manufactured component having shape, dimensional, form and surface properties. Geometrical Product Specification (GPS) impacts all products in terms of these properties, from macro- to nano-scale components, from top-down manufacturing to bottom-up processes [1]. It applies to high-tech, defence, aerospace, automotive, electronics, computing, MEMS, biomedical, domestic and most manufactured products.

GPS is the means of communication in which designers, production engineers and metrologists exchange unambiguous information concerning product GPS requirements [1]. Because of this GPS documentation may be regarded as the basis of a binding legal contract. In such a market, GPS is the only stable means of communication. Consequently, incorrect and ambiguous definitions of GPS-requirements constitute high economical risks to industry and are liable to be subject to disputes between companies. Thus the understanding and implementation of the GPS system is very important to industry.

In order to optimise resources through the scientific and economic management of the variability of all production processes, the next generation GPS system [2&3] has been shown to be a revolutionary breakthrough. However its wide acceptance and application in industry has been a great problem. The next generations of GPS standards are considered to be too theoretical, abstract, complex and over-elaborate. It is proving very difficult for industry to understand and operate them effectively. This point is especially true for small and medium businesses, where resources are not available to interpret and implement GPS correctly.

It is widely recognized that a software tool for GPS implementation needs to be developed to solve the problems. Therefore, a designer, an engineer, or a manufacturer does not need to be an expert in GPS system, having in mind all the complex standards. During the last few years, a few software tools have been developed, but they are almost according to old technical standards, providing no functional content and dealing only with a few technical specifications [1]. A PC-based knowledge system, Interactive Surface Modeling (ISM) developed by Chalmers University of Technology [4] connected the functional demands with the characterization and the manufacturing of the surface together, but since it relies on the relational database, it lacks a fundamental platform for general surface texture knowledge.

In this thesis, the mathematical foundations for a knowledge-based VirtualSurf system are developed. It is envisaged that the VirtualSurf system will overcome the above mentioned problems for surface texture, which is a critical part of the next generation of GPS, and act as a next generation smart standard for surface texture. It aims to advance considerably the current state-of-the-art by creating a virtual knowledge-based intelligent system to solve difficulties in surface texture specification and verification [5]. The VirtualSurf system will be used by designers, production engineers and metrologists as a tool to provide a common language for understanding surface texture.

1.2 Objectives and Approaches

A knowledge-based system is developed which will provide expert knowledge of surface texture to link surface function, specification of micro- and nano-geometry through manufacture, and verification. The intelligent knowledge-base should be capable of incorporating knowledge from multiple sources (standards, books, experts, etc), adding new knowledge from these sources and still remain a coherent reliable system. The system should provide a universal platform for engineers in industry, making it easier for them to understand and use the latest surface texture knowledge.

The thesis comprises a number of component studies in order to fulfil the above aim, these studies comprise:

1. An investigation of the different knowledge entities/structures within surface texture and finding suitable categories that describes them. The knowledge-bases cover: function, specification, manufacture and verification of surface texture knowledge for the profile design and metrology; including: functional requirements, parameter selection, filtration, manufacturing methods, instrumentation, and measurement parameters.
2. Suitable data structures for knowledge representation will be investigated, including different data models: relational data model, object-oriented model, functional model, pattern languages, fuzzy logic, etc. An intensional framework,

using category theory will be applied to define the relationships between the previously identified entities/structures.

3. A specification knowledge-base will be developed, which can be used to provide correct and unambiguous geometrical specifications (the technical specification) relevant to the functional design intent; A verification knowledge-base will be developed to provide appropriate measurement procedures and equipment appropriate to the technical specifications; A function knowledge-base will then be developed to specify the functional design intent of the surface; And a manufacturing knowledge-base will be developed to choose a suitable production process to manufacture surfaces according to the technical specifications;
4. The relationships and structures in knowledge-bases above will be identified and represented using category theory in order to transform them into a computer readable form for an intelligent system to make inferences.
5. Two case studies will illustrate the synthesise of four knowledge-bases into one integrated knowledge-based system.

1.3 Novel work

The novel work in this thesis is applying category theory [6-9] in the following aspects:

- Representation of a stable measurement procedure. The stability of measurement procedures is very important for design and metrology of a product [10-11]. Category theory is used to describe and define the stability of a measurement procedure, which ensures the consistency of knowledge acquisition for knowledge-based system.
- Representation of the knowledge stored in the knowledge-based system. The category model based on category theory [12-16] generates different data models and provides a fundamental formula for knowledge representation in this thesis.
- Representation of the data refinement process [17]. Data refinement is used to convert a simplified abstract data model into a complex implementable data model. During the refinement processes, the initial knowledge-base is refined to produce a high-performance system. Category theory is applied to secure the consistency between simplified abstract models and complex implementable models.
- Application of the developed categorical structures to identify and model the knowledge structures found in the surface texture literature and building a virtual knowledge-based intelligent system for surface texture.

1.4 Structure of the thesis

The various knowledge sources will be investigated in this thesis, not only from the engineering field, but also from computer science and mathematics. Chapter 2 introduces the background of knowledge-based system (KBS). Knowledge representation is the key procedure to the design of a KBS. Different data models for knowledge representation are then investigated and compared. A new data model – category model is introduced in Chapter 3, which is the mathematical foundation in this thesis and applied to represent surface texture knowledge.

Chapter 4 begins to discuss about the GPS system and surface texture general model. Different structures for separate knowledge-bases in the general model are investigated, providing a guideline for refinement processes in next few chapters. Chapters 5 to 8 are focused on the main design of knowledge-based system: specification knowledge-base refinement, verification knowledge-base refinement, function knowledge-base refinement and manufacture knowledge-base refinement respectively. Detailed refinement procedures are described and then knowledge representation models have been provided together with interfaces of the system. Two case studies are discussed in Chapter 9 which provide examples and illustrate the implementation of the integrated Virtualsurf system.

Finally, Chapter 10 provides a summary, conclusions and some future works.

References

[1] X. Jiang (2004), *A Knowledge-Based Intelligent System for Engineering and Bio-medical Engineering Surface Texture (VirtualSurf)*, University Research Project

[2] ISO/TR 14638 (1995), *Geometrical Product Specifications (GPS) –Masterplan,* International Organisation for Standardisation, Geneva

[3] ISO TC/213 (2001), *Vision Statement*, http://isotc213.ds.dk

[4] Robert Ohlsson (1996), *A Topographic Study of Functional Surfaces*, Thesis for the Degree of Doctor of Philosophy, Chalmers University of Technology, ISBN 91-7197-351-6

[5] ISO/TC213 WG16N084 (2003): *Masterplan for the uniform of surface texture for profile and areal standards*

[6] William Lawvere & Stephen H. Schanual (1997), *Conceptual Mathematics, A first introduction to categories*, Cambridge University Press, ISBN 0-521-47817-0

[7] Benjamin C. Pierce (1991), *Basic Category Theory for Computer Scientists*, MIT press, ISBN 0-262-66071-7

[8] Michael Barr & Charles Wells (1995), *Category Theory for Computing Science 2nd*

ed, Prentice Hall, ISBN 0-13-323809-1

[9] J. Lambek and P. J. Scott (1986), *Introduction to higher order categorical logic, Cambridge University Press,* ISBN 0-521-24665-2

[10] Paul J Scott (2006), *The case of surface texture parameter RSm*, Measurement Science and Technology, 17, 559-564

[11] Paul J Scott (2004), *Pattern analysis and metrology: the extraction of stable features from observable measurements*, Proc.R.Soc.Lond.A (2004) 460, 2845-2864

[12] Stefan Stanezyk (1993), *Theory and Practice of Relational Databases*, UCL Press Ltd, ISBN 1-85728-232-9

[13] John G. Hughes (1991), *Object-Orientated Databases*, Prentice Hall, ISBN 0-13-629874-5

[14] M.Z. Bleyberg (1992), *A Categorial View of Databases*, Computing and Information Sciences Department, Kanas State University, http://www.cis.ksu.edu/~maria/papers/ara.ps

[15] N.N Rossiter & D.A Nelson (1994), *The Categorical Product Data Model as Formalism for Object-Relational Databases*, Technical Report 505, Department of Computing Science, University of Newcastle, 1995

[16] D.A Nelson & N.N Rossiter (1995), *Prototyping a Categorical Database in P/FDM*, Technical Report 532, Department of Computing Science, University of Newcastle, 1995

[17] Robert M. Colomb (2001), *Category-Theoretic Fibration as an Abstraction Mechanism in Information Systems*, Acta Informatica 38, 1–44 (2001) c Springer-Verlag

Chapter 2

Knowledge-based system

2.1 Introduction

It is widely recognized that "Knowledge-based systems (KBSs) are tools for building applications that draw logical inferences from their stored knowledge of the problem domain" [1]. In around sixty years' development history, KBSs already has several applications, such as modern database and expert systems. All of them have the same KBS framework but also have different emphasis among the components. For example, the database has the advantage of storing large amounts of data information, while the expert system is good at producing inferences.

In general, a KBS has three main components: knowledge-base, inference engine and end-user interface. In the design of a KBS, knowledge acquisition and representation of a knowledge-base is the first and one of the most important procedures. The latter section of this chapter then focuses on introduction of different data models for knowledge representation, including traditional data models, the fuzzy model and the pattern language.

2.2 Knowledge-based System

KBS does not have a long development history (1943), but it already has several branches and applications, such as Artificial intelligence (AI), Expert System (ES), Decision Support System (DSS) and so on. Since the use of KBSs is increasing and expanding their boundaries, there has been no uniform classification of them until now, the figure 2.1 provides the view to classify KBSs in this thesis. In figure 2.1, Database System can be seen as a limited Knowledge-based System [2]. DSS and AI both are types of KBS, AI is the study of computers doing tasks that would be considered to require intelligence if a human did them. DSS assists in decision making and problem solving, and is extensively used in business and industry [3]. ES is the best example of a commercial success of AI, it is a computer program that draws upon the knowledge of human experts captured in a knowledge-base to solve problems that normally require human expertise [3].

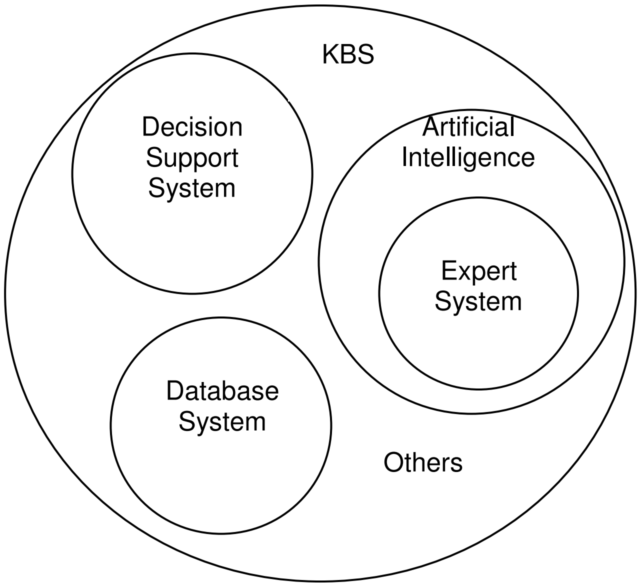


Figure 2.1 Classifications of KBS

Databases are now very commonly used in many commercial areas. A database is a shared collection of structured data, designed to meet the needs of an organization's information system [4]. In order to convey a designer's understanding of the information of the enterprise, a high-level data model is used in the design to represent the data information in databases. A data model is a general architecture for describing data, relationships between data and constraints on the data. It normally includes three essential components: the first, a structural part, including a set of rules which are used to construct the database; the second, a manipulative part, i.e. the operations that are allowed on the data; and the third, a set of integrity rules, which ensures the accuracy of the data [4].

Expert system is another technology devoted to representing real world aspects. It came from the development of AI (artificial intelligence). The expert system is a computer system that draws upon knowledge from human experts in a particular domain to solve problems that normally require human expertise in that domain; it must be capable of solving problems directly [5]. Similar to a database system in that an expert system contains a knowledge base, but it also represents another kind of knowledge—namely rules, which is a major component of the expert system. Rules can be used to infer new instances of the objects or new instances of a relationship from previous objects.

In the early days, database and expert systems were developed to represent different aspects of the real world. Database systems had the ability to store large amounts of structured data, together with sophisticated data management facilities; while expert system aimed to store rules and had the ability to produce inferences. It is clear that the combination of these two technologies would benefit both of them: with expert systems contributing to database systems in areas such as providing a useful reasoning ability and database systems contributing to expert systems in giving them the ability to access large collections of facts [2].

Knowledge-based systems are tools for building applications that draw logical inferences from their stored knowledge of the problem domain [1]. Both database systems and expert systems are particular kinds of knowledge-based systems and have the same general framework, see figure 2.2. The main components of the basic structure of a KBS include [3]: Knowledge-base, inference engine and end-user interface. The knowledge-base consists of facts, rules, heuristics, and other relevant information, and is used by the inference mechanism to provide expert opinion and other useful resources for the users through an interface. This interface must be user friendly to allow the user to easily use the systems.

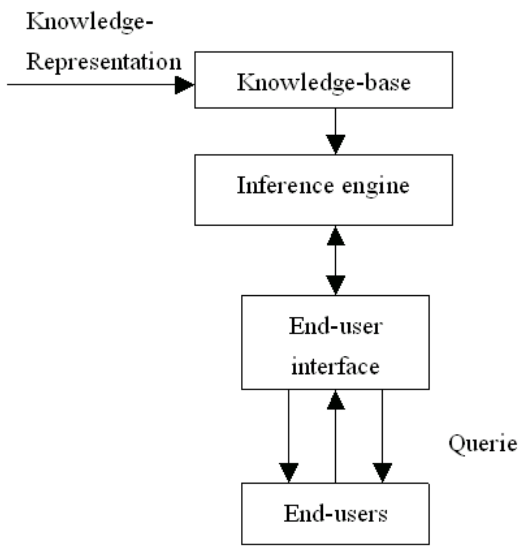


Figure 2.2 Basic Structure of a KBS

In this thesis emphasis will be on the knowledge-base, particularly at knowledge representation, which is an important step for managing the knowledge. Knowledge representation refers to data models, which is similar to data representation in traditional databases. The difference between them is that knowledge is more than data; it is data plus information necessary to make inferences and reach conclusions [3], including facts, theories, heuristics, relationships, attributes, observations, definitions and so on. Nowadays, some current popular database management systems, like Oracle, SQL server already have the ability to represent knowledge and also use programming language to make inferences, and they all use the traditional data representation model – relational data model. The details are introduced in the next section.

2.3 Knowledge representation

Knowledge-base requires special data models for the knowledge representation. There are some popular data models for data (knowledge) representation at present. They all have some advantages and disadvantages. There are two rules to evaluate a data model for certain knowledge representation: 1) naturally represent the knowledge; 2) needs to be easy to search and modify the knowledge.

2.3.1 Different data models

The following discusses different data models for knowledge representation.

2.3.1.1 Traditional data models (record-based data models)

Traditional data models include: network data model [4], hierarchical data model [4] and the relational data model [4]. They are all record-based logical data models.

Network model

In the network model, data is represented as collections of records and relationships are represented by sets [4]. Figure 2.3 shows an example graph structure of a network model, in which records appear as nodes and sets appear as edges. The set type "major of" links the owner record type "department" together with the member record type "student".

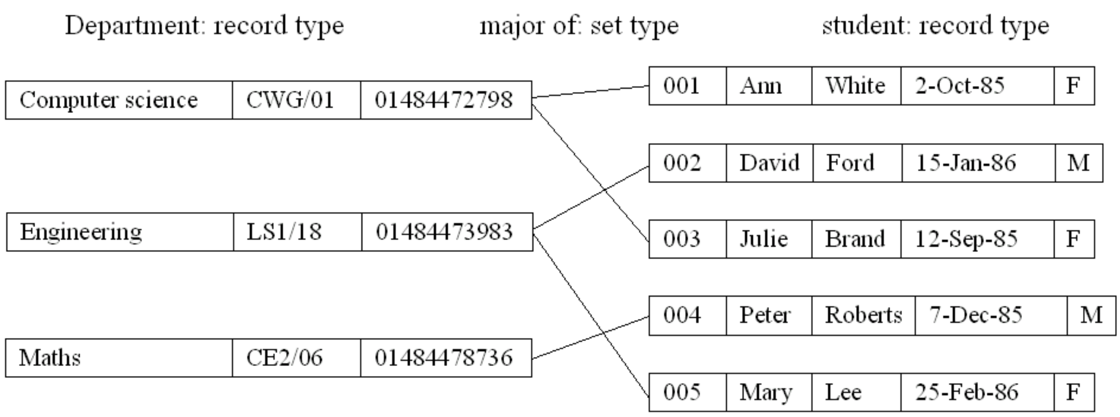


Figure 2.3 An example of a network schema

The hierarchical model

The hierarchical data model is a restricted network model, in which each node is allowed to have only one parent [4], see figure 2.4, it is a general tree structure. Again, data is represented as collections of records and relationships are represented by sets.

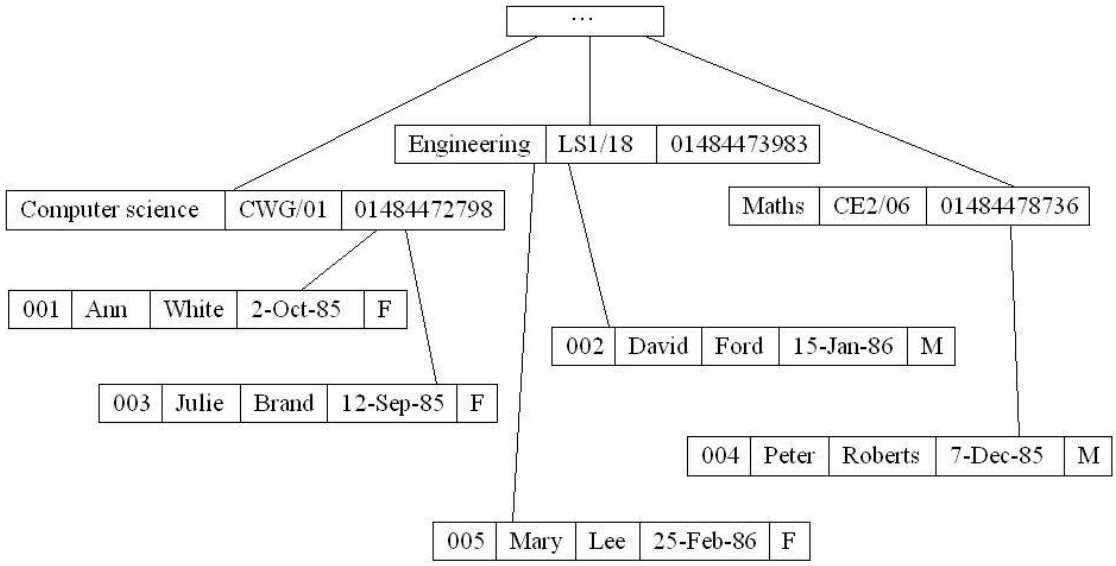


Figure 2.4 An example of a hierarchical schema

The relational model

The relational model is based on mathematical relations, in which data and relations are both represented as tables [4]. Table 2.1 and Table 2.2 give a typical relational data model [6].

Location	Lecturer	Department	Degree	Subject	Year	Students
Building A	Norman	Maths	BSc	Software Eng	1	90
Building A	Norman	Maths	BSc	Software Eng	2	50
Building A	Norman	Maths	BSc	Software Eng	3	30
Building A	Norman	Maths	BSc	Discrete Maths	1	90
Building A	Norman	Maths	BSc	Discrete Maths	2	70
Building B	Peter	Maths	BSc	Software Eng	1	90
Building B	Peter	Maths	BSc	Software Eng	2	50
Building B	Peter	Maths	BSc	Software Eng	3	30
Building B	Gillian	Computing	BSc	Discrete Maths	1	90
Building B	Gillian	Computing	BSc	Discrete Maths	2	70

Table 2.1 The relation *teaching*

Relation R1	
Location	Lecturer
Building A	Norman
Building B	Peter
Building B	Gillian

Relation R2	
Lecturer	Department
Norman	Maths
Peter	Maths
Gillian	Computing

Relation R3	
Subject	Degree
Software Eng	BSc
Discrete Maths	BSc

Relation R4	
Lecturer	Subject
Norman	Discrete Maths
Norman	Software Eng
Peter	Software Eng
Gillian	Discrete Maths

Relation R5		
Subject	Year	Students
Software Eng	1	90
Software Eng	2	50
Software Eng	3	30
Discrete Maths	1	90
Discrete Maths	2	70

Table 2.2 A set of simple relations

The relational data model is based on the mathematical theory of sets and relations. In this model, data relationships are represented by tables, each horizontal row describes a record (tuple) and each column describes one of the attributes (data fields) of the record, as illustrated in table 2.1[3]. The relation *teaching* can be viewed as a model of a particular view of the faculty's teaching system. Table 2.2 shows a better model representing a set of simple relations that captures the same information as Table 2.1; there is no redundant information here. A number of mathematical operations can then be applied to the relations themselves.

The first two data models were mainly used in early database systems; they still require the users to know how to access the physical database. Now the majority of commercial database systems are based on the relational data model, which provides data independence.

2.3.1.2 Object-based data models

The main drawback of the record-based data models is that they lack the abilities to specify the constraints on the data. In order to facilitate the representation of the real world, object-based data models are used in the design, such as entity relationship data model and some semantical data models: functional data model, object-oriented data model etc.

The entity-relationship model

The entity – relationship model is a high-level conceptual data model, and is commonly used as a basis of the logical data model design for the database system [4]. It comprises the entity, the relationship types and the attributes. Figure 2.5 shows an example of an entity-relationship data model.

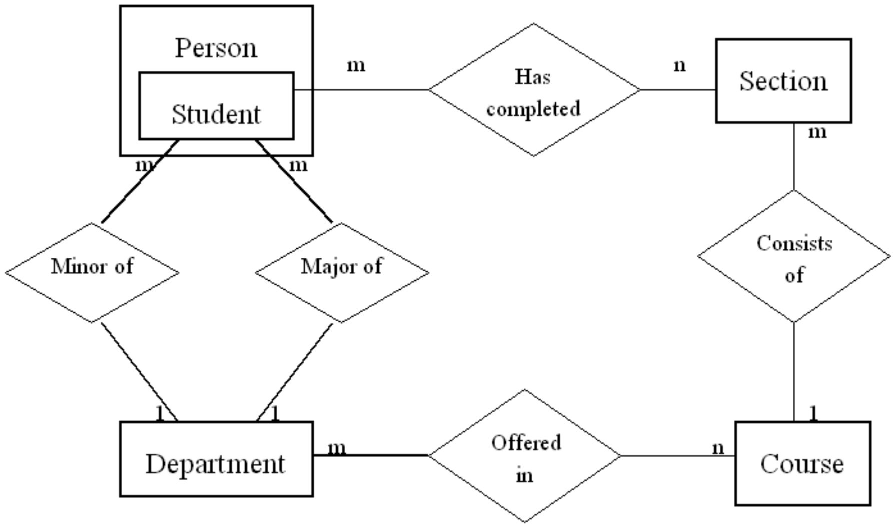
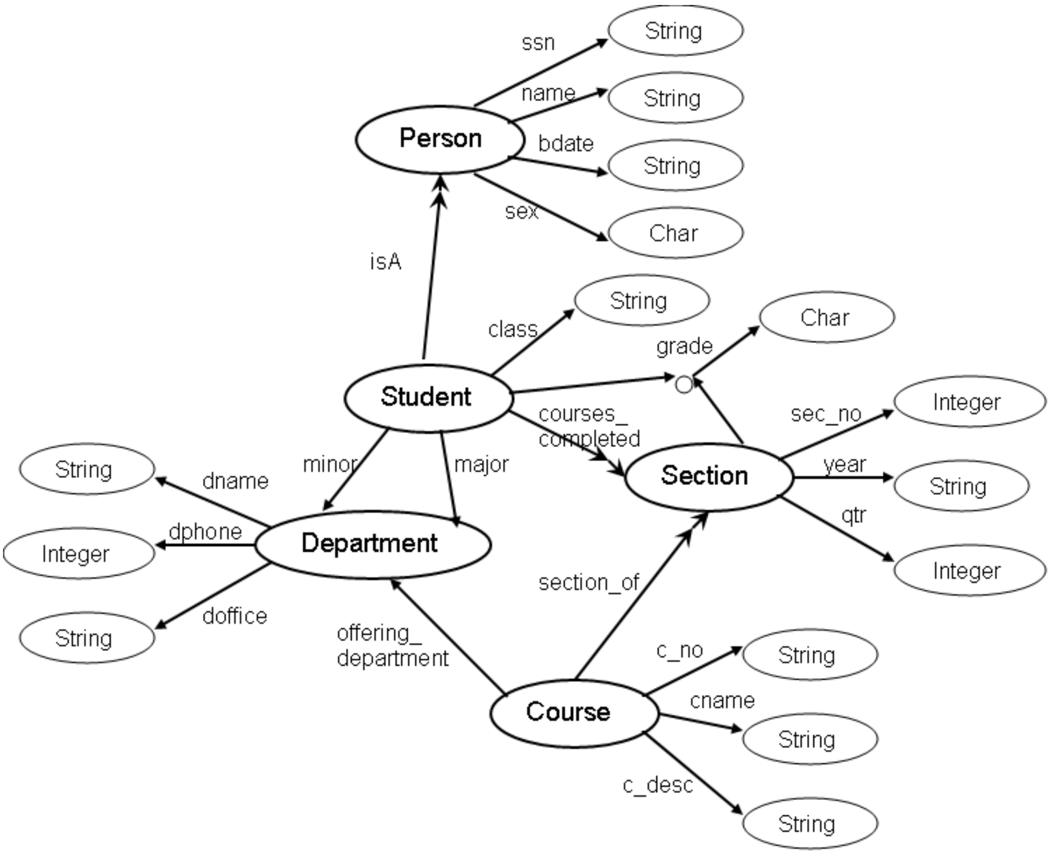


Figure 2.5 An example of an entity-relationship data model

The functional model

The functional data model is based on entities and functions, and also provides a natural language DAPLEX to give a more natural representation [7]. DAPLEX is a data definition and manipulation language for database systems, attempting to provide a database system interface which allows the users to more directly model the way they think about the problems they are trying to solve [7]. Therefore, the Functional data model has been proposed as a suitable formal and practical basis for object-oriented databases.

Figure 2.6 gives an example of a functional data model with the DAPLEX schema. It includes five entities: Person, Student, Department, Section and Course. The name of each arrow represents the name of the function with each entity, and the entity type at the head of each arrow represents the function's type. The single headed arrows are single-valued functions, i.e. return one entity, and the double headed arrows are multi-valued functions, i.e. return a set of entities. It is no longer required to create extra tables as in the relational model, multi-valued functions allow a many-to-many, or one-to-many relation to be defined [7].



declare Person() ⇒> ENTITY
declare ssn(Person) ⇒ STRING
declare name(Person) ⇒ STRING
declare bdate(Person) ⇒ STRING
declare sex(Person) ⇒ CHAR

declare Student() ⇒> Person
declare class(Student) ⇒ STRING
declare minor(Student) ⇒ Department
declare major(Student) ⇒ Department
declare courses_completed(Student) ⇒> Section

FOR EACH Student SUCH THAT dname(minor(Student)) = 'Maths'
PRINT name(Student)

Figure 2.6 An example of a functional schema

The object-oriented model

The object-oriented data model receives more and more attention these days with the development of the object-oriented programming [4]. It's a (logical) data model that captures the semantics of objects supported in object-oriented programming. It allows the real world to be modeled more closely, as it supports the building of complex objects, which are more natural and realistic representation of real-world objects.

Figure 2.7 gives an example of object-oriented schema. A relation or table in a relational data model can be considered to be analogous to a class in an object-oriented data model, such as "Department". A tuple is similar to the attributes of a class, such as attributes "dname", "doffice" and "dphone". There are no behaviours represented in the relational data model, which can be defined by a set of operations in the object-oriented data model.

```
interface Department{                    // define interface for Department
/* define attributes */
     attribute string dname;
     attribute string doffice;
     attribute string dphone;
/* define operations */
   void add _department( );
}
```

Figure 2.7 An example of object-oriented schema

There are several advantages of object-oriented databases [4]:

1) Enhanced modeling capabilities. The object, which encapsulated both state and behaviour naturally represents real world objects. It provides higher performance management of objects, and enables better management of the complex interrelationships between objects.
2) Removal of impedance mismatch[1] .It eliminates many inconveniences that occur in mapping a declarative language such as SQL and an imperative language such as "C".
3) Support for long duration transactions.

---

[1] **Impedance mismatch** describes an inadequate or excessive ability of one system to accommodate input from another. It occurs when object-oriented programming applies to relational schema, i.e., Mapping objects used in an application to tables stored in a relational database. Mapping objects to tables and vice versa creates a performance disadvantage when you have complex data.

The main drawback of the object-orientation database is that there is no universally agreed data model for an object-oriented DBMS, and most models lack a theoretical foundation.

Expert system – production rules

There are three representation formalisms used in conventional expert systems to represent domain knowledge [8]: Production rules, structured objects and predicate logic. Production rules are certainly the most prevalent of the three. Expert system designed using the production rule formalism is said to be a production system. Production rules, also known as "condition-action" rules or "if-then" rules, consist of condition and action pairs of the form "IF condition THEN action", i.e., if there exists conditions 1 to n, then perform actions 1 till n, see figure 2.8.

```
IF condition_1
   AND condition_2
   AND ...
   AND condition_n
THEN action_1
   AND action_2
   AND ...
   AND action_n
```

Figure 2.8 Production rules

The production rules define a set of allowed transformations which move a problem from its initial statement to its solution. The following are some examples of production rules, see figure 2.9.

```
Rule 1:   IF Mary majors computer_science
                AND Mary is second_year
          THEN Mary studying_week 25_hrs

Rule 2:   IF Peter majors computer_science
                AND Peter minors enginnering
                AND Peter is first_year
          THEN Peter studying_week 35_hrs
```

Figure 2.9 Sample production rules

The inference engine in the expert system will match the facts stored against the conditions or IF parts of the rule in the knowledge-base, and then forward the data stored to the goal to be solved.

Traditional expert systems can store few facts, when more facts need to be accessed, the reaction and efficiency of the system will decrease rapidly.

Comparison of different models

The popular methods of the knowledge representation are discussed above, which are relational data model, object-oriented data model and production rules. Although they are most prevalent methods today, they all have their own main drawback.

The dominant paradigm for data models is the "relational data model" [9]. However, the attributes of simplicity, including the minimality (i.e., a small number of data constructs) and nonredundancy make the relational model unrepresentive of the way humans model the world. Therefore, they are not sufficient for some new applications, such as engineering databases, e.g. CAD (computer-aided design), multimedia databases etc. Those applications require new features of the database [4], including:

1) Complex objects: A complex object in the real world contains other objects.
2) Behavioural data: Distinct objects may need to respond in different ways to the same command. This capability is provided by the methods of OODBs and by the rule base of Knowledge based systems.
3) Meta knowledge: General rules about the application rather than specific tuples (i.e., data about data) form an important part of expert databases.
4) Long duration transactions: some applications involve human interaction with the data, which lead to long transactions.

In these situations, the object-oriented data model [10] is appropriate to meet the requirements [4]. The object in the model includes not only the attributes that describe the state but also the actions associated with the object, and its behaviour. The object is said to encapsulate both state and behaviour. But unfortunately this data model lacks a universal formal basis and mathematical foundations to ensure the database remains a coherent and reliable system as new knowledge is added.

Another data representation method - Production rules of expert system have good inference abilities, but lack the ability to store abundant knowledge.

Table 2.3 gives the comparison results of these data models. Although the relational data model has good mathematical foundations, the knowledge must take the structure of a table, making it very inflexible and unnatural when applied to real world problems. The object – orientated data model has the flexibility for knowledge to take any structure but lacks a universal formal basis and mathematical foundations. And the production rules lack the ability to store abundant knowledge.

	Universal formal basis	Mathematical foundations	Flexibility of structures	Abundant knowledge
Relational Model	√	√		√
Object-oriented Model			√	√
Production rules	√	√		

Table 2.3 Comparison of data models

2.3.2 Fuzzy logic

There is a problem of knowledge representation that the conventional systems have never handled satisfactorily: the ability to handle uncertain or incomplete information. In order to represent vagueness or uncertainty, fuzzy logic is developed, with a continuous range of possibilities from 0 for impossible to 1.0 for certainty [11].

Now the fuzzy logic is applied more and more to the database systems, much work has been done in fuzzy conceptual data modeling, namely fuzzy modeling. Some of them are extended to the entity-relationship models and some are extended to the object-oriented models.

Fuzzy sets

Classical set theory is based on two-valued logic, which says *a* must either be in set A or in set not-A, there is no situation that is both *a* in set A and *a* in set not-A. A fuzzy set is a set without a crisp, clearly defined boundary. It can contain elements with only a partial degree of membership. A fuzzy set is therefore a function, f, from an appropriate domain to the interval [0,1], where f(x) =0 denotes that x is not a member of the set, f(x) = 1 denotes that x is definitely a member, and all other values denote degrees of membership [8].

For example, consider a set S is the results of six students in an exam.

S = {50, 48, 65, 70, 85, 95}, which is a classical set.

Then consider the set of good results, there is not a clear boundary here. Fuzzy set can be applied. Define the functions g(40)=0, g(100)=1, and some histogram of monotonically increasing values between the two. Thus the good results set can be obtained as follows:

Good-results = {(50, 0.17), (48, 0.13), (65, 0.42), (70, 0.5), (85, 0.75), (95, 0.92)}

Fuzzy logic

Classical set theory is governed by a two-valued logic, while a fuzzy set theory can be represented to a many-valued logic in which propositions such as good-results have a value which is a real number between 0 and 1[8].

In fuzzy logic, the truth of any statement becomes a matter of degree. Fuzzy logic deals with conjuction by taking the minimum value of the disjuncts, thus

$f_{(F/G)}(x) = \min(f_F(x), f_G(x))$

e.g. good-results(50) $\wedge\neg$good-results(50) = 0.17

Fuzzy logic deals with disjuction by taking the maximum value of the disjuncts, thus

$f_{(F/G)}(x) = \max(f_F(x), f_G(x))$

e.g. good-results(95)v $\neg$good-results(95) = 0.92

Fuzzy object-oriented model

Fuzzy concepts are considered at different levels of the object-oriented model [12]:

a) Attribute values: the probability measures defined within the [0,1]-interval can be used to express the explicit uncertainty that affects an attribute value.
b) Relationships among objects: a new attribute can be used to stand for the belief in the corresponding aggregation, and use the appropriate truth scales for dealing with explicit uncertainty, i.e., add an extra attribute to the class to express the importance or strength in the connection.
c) Class extents: it only needs to add an extra attribute to the class and extend in a fuzzy way. The domain of this attribute could be the interval [0,1].
d) Inheritance relationships: use static variables of the suitable scales to express these connection degrees.
e) Definition of the type of a class: This new way of considering the type definition can be easily modeled over a traditional object-oriented model, using the concept of 1-ramified hierarchy of classes. A 1-ramified hierarchy of classes is defined as a series of classes C1, …, Ci−1,Ci, Ci+1, …, Cn

verifying the following properties:

- For any $i \in 1..n - 1$, Sub{Ci} = {Ci+1} (Sub{Ci} stands for the set of subclasses of Ci).
- For any $i \in 2..n$, Sup{Ci} = {Ci-1} (Sup{Ci} stands for the set of superclasses of Ci).
- A finite sequence of values αi exists, associated with the hierarchy, such that $\alpha 1 = 1$, $\alpha n > 0$, and $\alpha i > \alpha i+1$.

Each class of the hierarchy is used to represent an α-cut of the type being defined.

Fuzzy expert system

Fuzzy concepts are also used in the expert system, namely fuzzy expert system or fuzzy inference system. Fuzzy inference is the process of formulating the mapping from a given input to an output using fuzzy logic.

The following is the process to interpret if-then rules with fuzzy logic:

a) Fuzzy inputs: Resolve all fuzzy statements in the inputs to a degree of membership between 0 and 1. If there is only one part to the input, this is the degree of support for the rule.

b) Apply fuzzy operator to multiple part inputs: If there are multiple parts to the inputs, apply fuzzy logic operators and resolve the inputs to a single number between 0 and 1. This is the degree of support for the rule.

c) Apply implication method: Use the degree of support for the entire rule to shape the output fuzzy set. The consequent of a fuzzy rule assigns an entire fuzzy set to the output. This fuzzy set is represented by a membership function that is chosen to indicate the qualities of the consequent. If the input is only partially true, (i.e., is assigned a value less than 1), then the output fuzzy set is truncated according to the implication method.

2.3.3 Pattern language

This research also implies the pattern language, in order to provide an open platform for structured and complex knowledge. Both fuzzy sets and pattern language can be applied to the problem which has not a specific solution. Fuzzy sets consider the probability and generate a function to represent the solutions. Whereas pattern language gives several solutions under different contexts and forces, and then users can choose the most suitable one matching the current circumstance.

The term 'Pattern language' was initially introduced in Christopher Alexander's book 'A Pattern Language: Towns, Buildings, Construction' [13], which is used to refer to common problems of civil and architectural design, from how towns should be laid out to where windows should be placed in a room.

The idea was soon expanded into popular diverse fields, such as computer science patterns used in software engineering, interaction design patterns in human computer interaction and pedagogical patterns in education.

Pattern

'Each pattern describes a problem that occurs over and over again in our environment and then describes the core of the solution to that problem in such a way that you can use this solution a million times over without ever doing it the same way twice. ' [13]

A single design pattern in pattern language is that a common problem (or decision) in a design process, together with its best solution. Each pattern has a name, a descriptive entry, and some cross-references, much like a dictionary entry.

A pattern language

Just as words must have grammatical and semantic relationships to each other in order to make a spoken language useful, design patterns must be related to each other in order to form a pattern language. The patterns should be organized into a logical or naturally intuitive structure. Each pattern should indicate its relationship to other patterns and to the language as a whole [14].

A pattern language is a linked structure of patterns within a particular domain. It is characterized by

a) Naming the common problems in a field of interest
b) Describing the key characteristics of effective solutions for meeting some stated goals
c) Helping the designer move from problem to problem in a logical way
d) Allowing for many different paths through the design process

A pattern language is not a kind of programming language, it is a piece of literature that describes an architecture, a design, a framework or other structure.

Pattern form

A pattern is applied to document expertise. The typical pattern form includes the following items [15]:

Name: A word or simple phrase to describe the pattern.

Context: When to apply the problem.

Problem: A statement of the problem.

Solution: A solution to the problem. Many problems have more than one solution. The success of a solution is affected by the context or circumstances in which the problem exists.

Force: The often contradictory considerations to be considered when choosing a solution to a problem. Each solution considers certain forces. It optimizes some and may totally ignore others. The relative importance of the forces is determined by the context.

2.4 Summary

The knowledge-based system has been presented in this chapter, including the definition, the classification and the general structure. Some typical applications of KBS are introduced, such as database, expert system, DSS etc. The Knowledge-based System intended to be developed in this research is a new application of KBS, which generalizes both Database System and Expert System. Unlike the DSS, it's not designed to do decision making itself, it is used to give some suggestions or remind the user what needs to be done and how to do certain procedures. It is like a library or a help document covering all the useful knowledge of surface texture, and also has a friendly user interface to allow the users to make enquires and easily get what they want to know.

One of the most important design procedures of a KBS - knowledge representation has been outlined, which is important for managing the knowledge in knowledge-base. Different data models for knowledge representation have been introduced and compared. The dominant data model is the relational data model, which has good mathematical foundations, but lacks the flexibility to model the real world. The other, increasingly popular, the object – orientated data model has the flexibility for knowledge to take any structure but lacks a universal formal basis and mathematical foundations. On these bases, a new data model is to be proposed in this research in order to satisfy the requirements, which not only has a good mathematical foundation, but also has flexible structures for abundant knowledge.

References

[1] D.M. Hembry (1990), *Knowledge-based systems in the AD/cycle environment*, IBM Systems Journel, 29(2), 274-286

[2] Paul Beynon-Davies (1991), *Expert Database Systems: A Gentle Introduction,* McGRAW-HILL Book Company (UK) Limited, ISBN 0-07-707240-5

[3] D. Partridge and K.M.Hussain (1995), *Knowledge Based Information Systems,* McGRAW-HILL Book Company Europe, ISBN 0-07-707624-9

[4] Thomas Connolly and Carolyn Begg (1998), *Database Systems: A Practical Approach to Design, Implementation, and Management,* Addison Wesley, ISBN 0-201-34287-1

[5] J L Alty and M J Coombs (1984), *Expert systems: Concepts and examples,* NCC publications, ISBN 0-85012-399-2

[6] Norman Fenton and Gillian Hill (1993), *Systems Construction and Analysis: A Mathematical and Logical Framework,* McGRAW-HILL Book Company Europe, ISBN 0-07-707431-9

[7] Shipman (1981), *The Functional Data Model and the Data language DAPLEX,* Computer Corporation of America, ACM Transactions on Database Systems, 6(1981) 140-173

[8] Peter Jackson (1999), *Introduction to expert systems*, Addison Wesley Longman Limited, ISBN 0-201-87686-8

[9] Stefan Stanezyk (1993), *Theory and Practice of Relational Databases*, UCL Press Ltd, ISBN 1-85728-232-9

[10] John G. Hughes (1991), *Object-Orientated Databases*, Prentice Hall, ISBN 0-13-629874-5

[11] John F. Sowa (2000), *Knowledge Representation: Logical, Philosophical, and Computational Foundations*, Thomson Learning, ISBN 0 534-94965-7

[12] Zongmin Ma (2005), *Advances in Fuzzy Object-Oriented Databases: Modeling and Applications,* Idea Group Publishing, ISBN 1591403847

[13] Christopher Alexander (1977), *A Pattern Language: Towns, Buildings, Construction*, New York: Oxford University Press

[14] *Pattern language*, Wikipedia, the free encyclopedia, http://en.wikipedia.org/wiki/Pattern_language

[15] Linda Rising (1998), *The patterns handbook: techniques, strategies, and applications,* Cambridge University Press, ISBN 0-521-64818-1

Chapter 3

Category Model for Knowledge-based system

3.1 Introduction

Category theory is a general mathematical theory that deals in an abstract way with mathematical structures and relationships between them [1]. The basic concepts of category theory include categories, functors, natural transformations etc. In this thesis, category theory is used to resolve three important issues in the design of a knowledge-based system.

The first application of category theory is in the representation of a stable measurement procedure. Measurement can be found everywhere in daily lives, including product manufacturing. The stability of measurement procedures is very important for design and metrology of a product. Category theory is used to describe and define the stability of a measurement procedure.

The second application of category theory is the representation of the knowledge stored in the knowledge-based system. The category model based on category theory generates different data models and provides a fundamental formula for knowledge representation in this thesis.

The third application of category theory is the representation of the data refinement process. Data refinement is used to convert a simplified abstract data model into a complex implementable data model. During the refinement processes, the initial knowledge-base is refined to produce a high-performance system. Category theory, particularly the concepts of fibration and adjointness, is applied to secure the consistency between simplified abstract models and complex implementable models.

3.2 Measurement theory applies Category theory

Measurement theory is a foundation for measurement in both natural and social sciences. The degree to which things are measured can be regarded as the major difference between a "well-developed" science such as physics and some of the less "well-developed" sciences such as psychology or sociology [2]. For over 100 years philosophers, physicists, mathematicians and social scientists have pursued the definition or analysis of the concept of measurement.

3.2.1 Three theories of measurement

Different interpretations of measurement can lead to different models. There have been three main theories of measurements: representational measurement theory which is the dominant current measurement paradigm, operational measurement theory and classical measurement theory [3].

3.2.1.1 Representational measurement theory

The representational measurement theory consists of the following [3]:

1) An empirical relational system (ERS) including a set of objects with the relations between them, each of objects has one or more common attributes.
2) A numerical relational system (NRS) comprising numbers and the relationships between them. The numbers form the values of a variable.
3) A set of mappings from ERS to NRS, in such a way that the relationships between objects are matched by relationships between numbers.

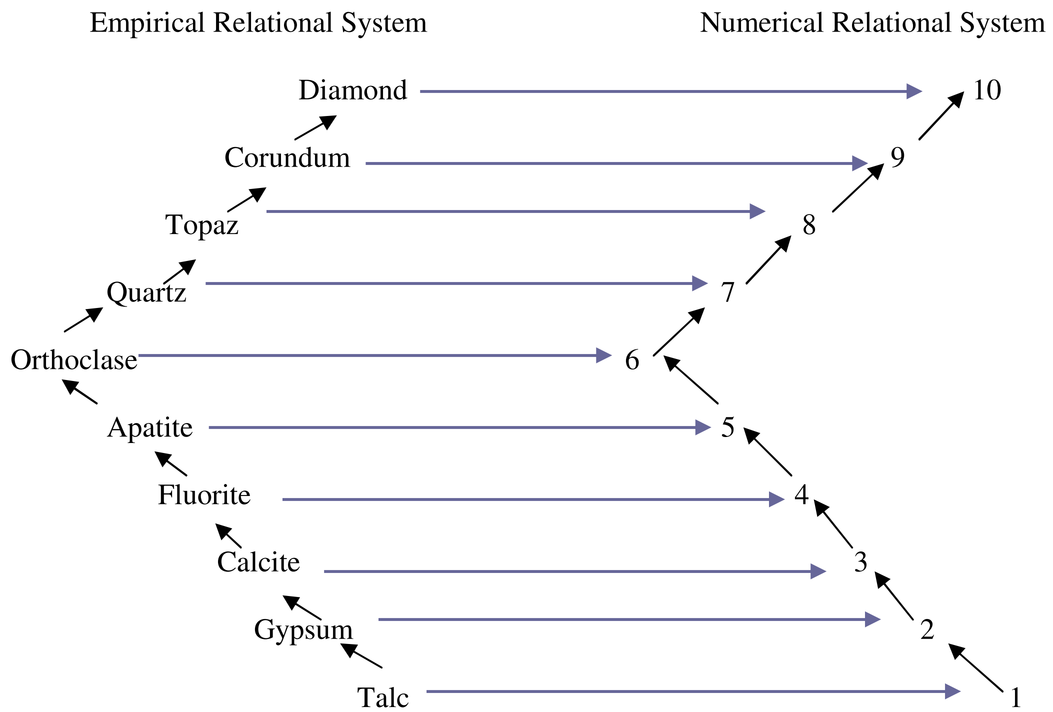


Figure 3.1 Mohs hardness scale for minerals

For example the Mohs hardness scale for minerals in figure 3.1 illustrates these ideas: the numbers in NRS match the hardness properties in ERS. It is obvious that Diamond is the hardest as it is designated by the number 10; therefore, it is harder than Corundum which is designated by the number 9, and Corundum is harder than Topaz etc. Talc is the least hard as it is designated by the number 1.

3.2.1.2 Operational measurement theory

Operationalism defines scientific concepts in terms of the operations used to identify or measure them. Thus, an attribute is defined by its measuring procedure, no more and no less, and has no ‘real’ existence beyond that. In operationalism the attribute and the variable are one and the same. This approach thus defines a measurement as any precisely specified operation that yields a number. It follows that, to be useful, the numerical assignment procedure has to be well defined. Arbitrariness in the procedure will reflect itself in ambiguity in the results [3]. Operationalism survives to this day with considerable influence in the social and behavioral sciences (especially psychology), where the methodological war cry to "operationalize your variables!" [4].

3.2.1.3 Classical measurement theory

According to classical measurement theory, measurement addresses the question of how much of a particular attribute an object has and thus only refers to attributes that are quantitative, i.e. an attribute whose values satisfy ordinal and additive relationships.

Classical measurement theory involves the discovery of the relationship between different quantities of the given attribute. The key word here is “discovery”.

3.2.1.4 Comparison of three measurement theories

Representational measurement theory seeks to represent or model empirical relationships, and so is based on a mapping from an assumed underlying reality; while in operational measurement theory, things start with the measurement procedure. Operationalism avoids assuming an underlying reality and so is fundamentally different from representationalism.

Representational measurement theory assigns numbers to objects to model their relationships, and operational measurement theory assigns numbers according to some consistent measurement procedure, classical measurement theory discovers pre-existing relationships.

It is interesting to consider the dramatic international rise in IQ scores in light of these three theories of measurement [5-6]: Has intelligence increased, or just the test scores? The representational theory makes clear the importance of this distinction. Because the structure of the representational theory is a set of mappings from ERS to NRS, in such a way that the relationships between objects are matched by relationships between numbers, the objects here are clearly distinguished between intelligence and test scores.

The operational theory makes the question impossible to ask, since there is no such thing as "intelligence" as distinct from scores on intelligence tests. The assessment of intelligence is as such measuring what is in the black box. The approach in operationalism defines a measurement as any precisely specified operation that yields a number. However, this number can not always exactly reflect human intelligence. The classical theory allows the question, "What do IQ tests really measure?" [5]. It is the procedure to discover the relationship between different quantities of the given attribute, such as intelligence, but it is difficult to see how intelligence can be regarded as having a magnitude susceptible to classical measurement by Michell's definition [7].

3.2.2 Stable measurement procedure

The representational measurement theory will be used to define the stability of the measurement procedure [8-9]. The measurement consists of the following under the representational measurement theory:

1) An empirical relational system (ERS) including a set of objects on which a measurand is defined together with the relations between measurands.
2) A numerical relational system (NRS) comprising numbers (derived values) and the relationships between them.
3) A set of mappings, called the measurement procedure, from ERS to NRS, in such a way that the relationships between measurands are matched by relationships between numbers.

It is considered here that when a measuring procedure is mathematically stable a “small” difference in the derived values implies a “small” difference in the measurand. Relationships between measurement values should reflect functional significant properties between the measurands or else the measurement has little practical meaning [8].

The following theorems and corollaries are proved in Scott 2004 [9] which provides a stability corollary that can be used to show when a measurement procedure is stable:

In topology open sets can be used to define ‘small’ differences between points. Using this concept the stability condition can be restated as follows. Define topologies on the space of measurands and the space of derived values such that the inverse image of every open set of the derived values is an open set of the measurands. This is the topological definition of a continuous mapping. Hence the stability condition is just a continuous mapping from the measurands to the derived values. It is necessary to define topologies on the spaces of measurands and measurement values to apply this definition [9].

Notes:

Topology: Topology is the study of geometrical structures. It builds on set theory, considering both sets of points and families of sets [10].

Let X be any set and let T be a family of subsets of X. Then T is a topology on X if

- Both the empty set and X are elements of T.
- Any union of elements of T is an element of T.
- Any intersection of finitely many elements of T is an element of T.

All sets in T are called open sets.

Open sets: In topology and related fields of mathematics, a set U is called open if, intuitively speaking, you can "wiggle" or "change" any point x in U by a small amount in any direction and still be inside U. Or, in other words, if x is surrounded only by elements of U — it cannot be on the edge of U [11].

As a typical example, consider the open interval (0,10) consisting of all real numbers x with $0 < x < 10$. Here, the topology is the usual topology on the real line. If you "wiggle" such an x a little bit (but not too much), then the wiggled version will still be a number between 0 and 10. Therefore, the interval (0,10) is open. However, the interval (0,10] consisting of all numbers x with $0 < x \leq 10$ is not open; if you take $x = 10$ and move even the tiniest bit in the positive direction, you will be outside of (0,10].

Inverse image: The inverse image of a set $B \subseteq Y$ under f is the subset of X defined by $f^{-1}[B] = \{x \in X \mid f(x) \in B\}$.

Partial pre-order: A partial pre-order is a binary relation *R* over a set *P* which is reflexive and transitive, i.e., for all *a*, *b*, and *c* in *P*:

- *aRa* (reflexivity);
- If *aRb* and *bRc* then *aRc* (transitivity).

An example of partial pre-order is the set of subsets of a given set (its power set) ordered by inclusion, see figure 3.2.

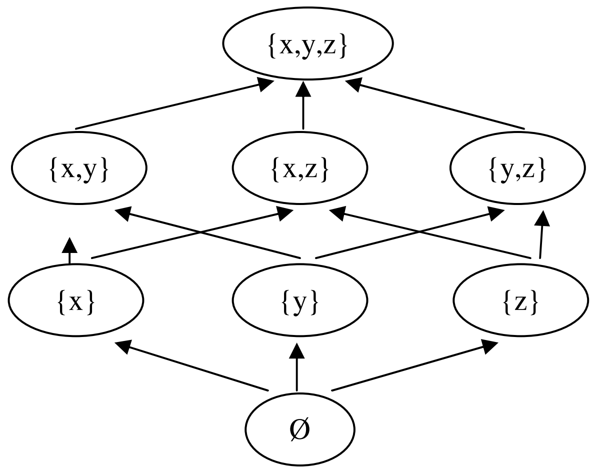


Figure 3.2 The set of subsets of {x,y,z}, ordered by inclusion

Finite sets: Finite sets are sets that has a finite number of members. The number of elements in a finite set A is denoted by n(A).

Example:

A = {0, 2, 4, 6, 8, …, 100}

C = {x : x is an integer, 1 < x < 10}

An infinite set is a set which is not finite.

Example:

T = {x : x is a triangle}

N is the set of natural numbers

The surface to be measured is a continuous surface, which includes infinite number of points. A straight line being extracted to do the evaluation has an infinite number of points but can be reconstructed by just two points i.e. has finite information content. The number of points does not always equal the information content (it is always more or equal to the information content). Since it is not possible for a physical device to handle a measurement with infinite information content. The finite sets of information content are being extracted to characterize the surface characteristics.

Continuous function: A continuous function is a function for which, intuitively, small changes in the input result in small changes in the output. In figure 3.3, R and T are both set of real numbers, there is a function f that maps R to T, and suppose c is an element of R. The function f is said to be continuous at the point c if the following holds: For any number $\varepsilon > 0$, however small, there exists some number $\delta > 0$ such that for all x in the domain R with $c - \delta < x < c + \delta$, the value of f(x) in T satisfies:

$$f(c) - \varepsilon < f(x) < f(c) + \varepsilon.$$

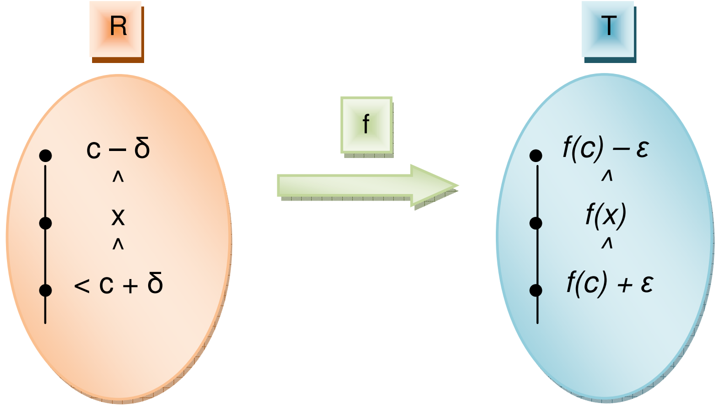


Figure 3.3 Example of continuous function

For example, a square root is a continuous function. A number r is a square root of a number x, such that $r = \sqrt{x}$. For x $\geq$ 0, any small change in the input x results in small change in the output r, see figure 3.4.

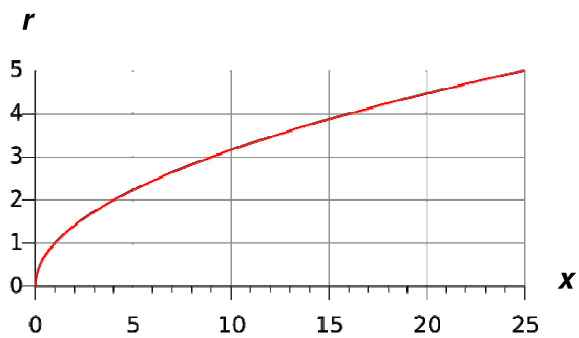


Figure 3.4 Example of a square root function

The following theorem gives the relationship between topologies and partial pre-orders.

Theorem 1

Let L be a finite set. Then there is a one-to-one correspondence between the topologies on L, and the partial pre-orders (i.e. reflexive and transitive relations) on L by using the following constructions.

Construction 1

Let T be a topology on L. Define the relation R by the rule that xRy if every open set containing x also contains y. it is trivial that R is reflexive and transitive; that is, R is a partial pre-order.

Construction 2

Let R be a partial pre-order on L. Call a subset U of L open if, whenever $x \in U$, we have $R(x) \subseteq U$, where $R(x) = \{y : xRy\}$.

Proof: See Cameron 1994 [12].

Theorem 2

Let L and M be sets with partial pre-orders defined.

Let $\psi: L \rightarrow M$ define an increasing mapping from L to M.

That is to say, $xRy \Rightarrow \psi(x)R\ \psi(y)$ with $x, y \in L$.

Using constructions 2 to define topologies on *L* and *M*, the inverse images of the open sets of *M* are then open sets in *L*.

Proof: see Scott 2004 [9].

Theorem 3

Let L an d M be sets with partial pre-orders defined. Use construction 2 to define topologies on L and M.

Let $\psi: L \rightarrow M$ define a continuous function from L to M. that is to say, the inverse mappings of the open sets of M are open sets in L. then xRy implies $\psi(x)R\ \psi(y)$ with $x, y \in L$.

Proof: see Scott 2004 [9].

Theorems 1 to 3 lead to the following corollaries.

Corollary 1

Finite sets of measurands and derived values with partial pre-orders and increasing mappings map one-to-one onto finite topologies with continuous functions.

Corollary 2 - Stability corollary

If for a measurement procedure the relational structures of the measurand and the derived values are both partial pre-orders and the mapping between them are also increasing mappings then the measurement procedure is stable.

3.2.3 Category theory applied to Measurement theory

In this section of the thesis, the relationship between category theory and the representative theory of measurement is developed. Basic category theory concepts: category, functor and natural transformation, are applied to provide a framework for representational measurement.

3.2.3.1 Category

A category consists of a class of objects and a class of arrows such that classes of arrows are associated with every pair of objects (could be the empty class). (See section 3.3.2 and the full description of category theory see the reference Benjamin C. Pierce [1])

It is mentioned in the last section that in order to satisfy the Stability corollary, both ERS and NRS for a measurement procedure should be partial pre-orders, which have the properties of reflexive and transitive. Since a category is a collection of objects and arrows between these objects, and the arrows in a category have two properties – reflexive and transitive, which are the same as the properties of partial pre-orders, ERS and NRS in the partial pre-order would be regarded as categories.

Reflexive (identity map): A map in which the domain and the codomain are the same set A, and for each a in A, f(a)=a.

$$A \xrightarrow{I_A} A$$

Figure 3.5 Reflexive of the category

Transitive (Composition of maps): two maps are combined to obtain a third map.
For example, category ERS consists of two arrows f and g: together with a third arrow g ∘ f the composition of f and g.

$$\text{ERS:}\quad \begin{array}{c} A \xrightarrow{f} B \xrightarrow{g} C \\ \xrightarrow[g \circ f]{\hspace{6em}} \end{array}$$

$$f\!: A \to B \quad g\!: B \to C$$

$$g \circ f\!: \ \mathrm{A} \to C$$

Figure 3.6 Transitive of the category

3.2.3.2 Functor

A functor is a special type of mapping between categories in category theory: its objects are categories and its arrows are certain structure-preserving maps between categories [1].

Let C and E be categories. A functor P from C to E is a mapping that

- Associates to each object X ∈ C an object P(X) ∈ E,

- Associates to each morphism u: $X \to Y \in C$, a morphism $P(u): P(X) \to P(Y) \in E$

Such that the following two properties hold:

- $P(id_X) = id_{P(X)}$ for every object $X \in C$
- $P(v \circ u) = P(v) \circ P(u)$ for all morphisms $u: X \to Y$ and $v: Y \to Z$.

That is, functors must preserve identity morphisms and composition of morphisms. It is obviously that increasing mapping can be represented by the category concept functor. For example, figure 3.7 shows a constant functor: The functor $C \to E$ is one which maps every object of C (X and Y) to a fixed object A in E and every morphism in C ($1_Y \circ w$ and v) to the identity morphism 1_A on A.

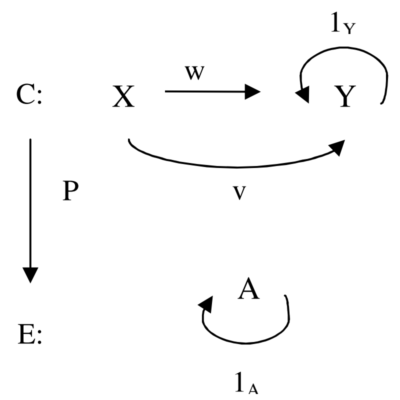


Figure 3.7 The constant functor

Therefore, the stability corollary is possible to be represented by category concepts:

If for a measurement procedure the relational structures of the measurand and the derived values are both partial pre-order categories and the mapping between them is functor then the measurement procedure is stable.

3.2.3.3 Natural transformation

A natural transformation is a relation between two functors [1]. Calibration refers to the process of determining the relation between the output of a measuring instrument and the value of the input quantity or attribute, a measurement standard. Calibration is often regarded as including the process of adjusting the output or indication on a measurement instrument to agree with value of the applied standard, within a specified accuracy.

Figure 3.8 shows the calibration process, the output value of the specimen is obtained after several measurement processes, such as partition, filtration, etc. And then the output value is compared with the specified value of the specimen. Each measurement process is a functor mapping from the measurand to the measured value after the measurement. The relations between these functors constitute the natural transformation.

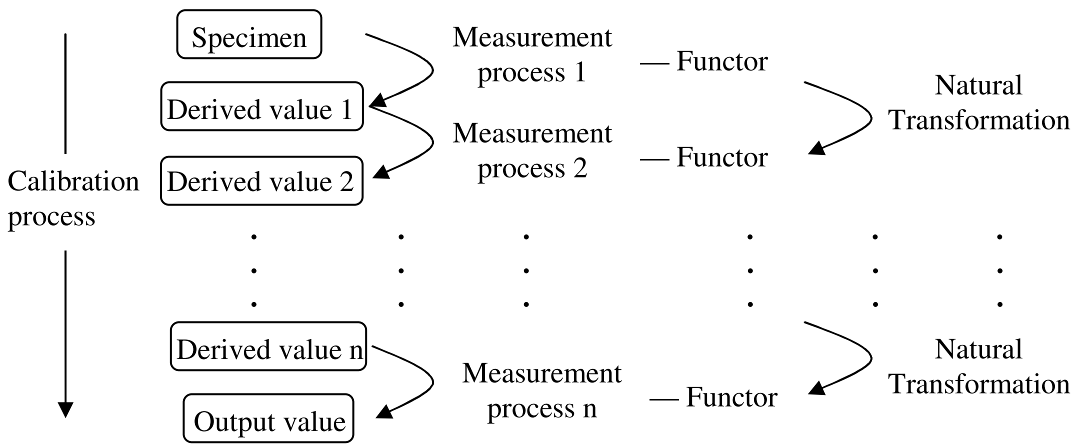


Figure 3.8 The calibration process – Natural transformation

Table 3.1 gives a conclusion and shows the relation between category concepts and the concepts of representational measurement theory:

Category theory concept	Description	Representational measurement theory concept
Category	Collection of objects and arrows	Relational system
Functor	Structure preserving mapping between categories	Structure preserving mapping between relational systems
Natural Transformation	Mapping between Functors	Calibration

Table 3.1 Category theory concepts and representational measurement theory concept

3.2.4 Application in GPS (Geometrical Product Specification and Verification) system

The development of stable measurement procedures definition in category theory is useful in providing guidance for the management of features in observable data. And therefore ensures the consistency of knowledge acquisition for knowledge-based system.

The operations in both specification and verification of GPS system (see Chapter 4) are proved to be stable by the stability corollary above, as there are successive functors between them as illustrated in figure 3.9. For example, for the procedure partition, the relational structures of the measurand in object skin model and the derived values after partition from the skin model are both partial pre-order categories, and the mapping between them is functor, therefore the procedure partition is stable.

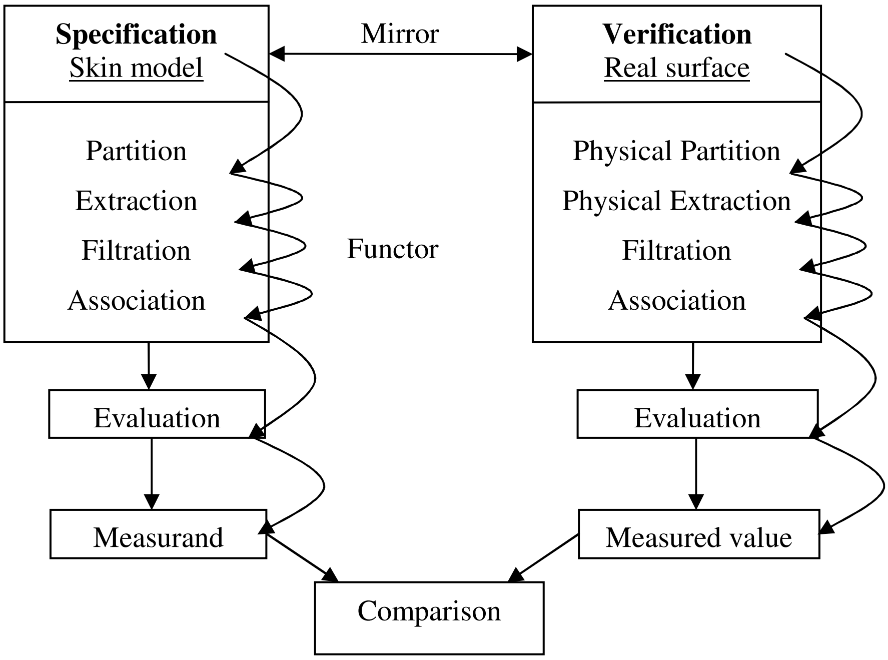
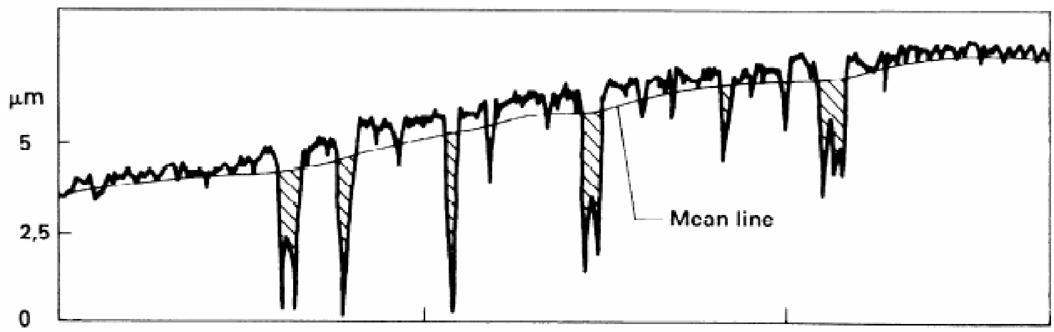


Figure 3.9 Stable measurement operations

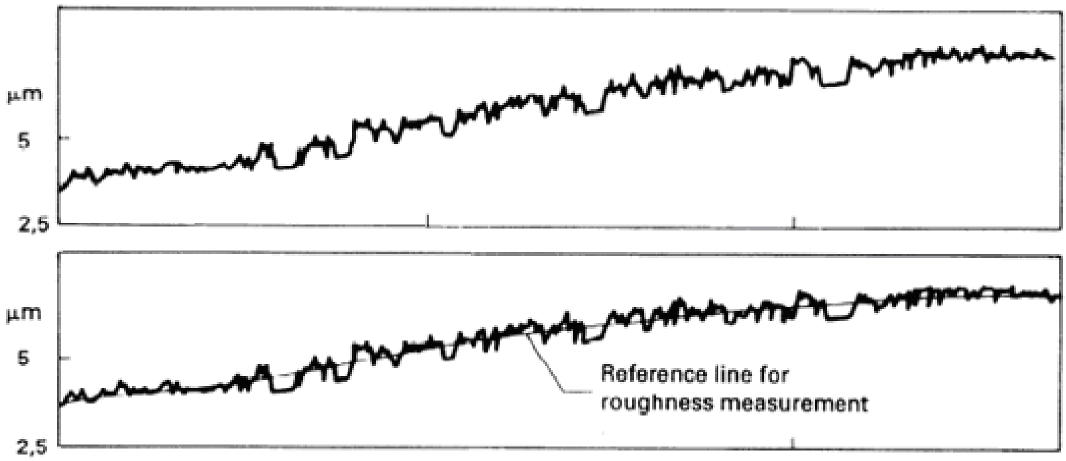
Figure 3.10 a)-d) shows a filtering process which is aimed to obtain the roughness profile for the surfaces having stratified functional properties (the original pictures are obtained from reference ISO 13565-1 [13]). Figure a) shows the unfiltered primary profile with the first mean line determined by a preliminary filtering of the primary profile with the phase correct filter. All valley portions which lie below this mean line are removed. In these places the primary profile is replaced by the curve of the mean line. Figure b) shows the unfiltered primary profile after suppression of valleys. The same filter is used again on this profile and then the second mean line (reference line) is obtained which is used to perform the assessment of profile parameter. In figure c) the reference line is transferred to the original primary profile and the roughness profile is obtained from the difference between the primary profile and the reference line, showing in figure d).

This filtration process is carried out in the above stages in order to determine the roughness profile for further evaluation. For example the roughness parameter *Rk*. *Rk* represents the depth of the roughness core profile, which is the roughness profile excluding the protruding peaks and deep valleys [14]. The information of distances between every measured point and the reference line is needed to calculate the parameter *Rk*. Figure c) shows the distances between every measured point and the reference line before the filtration, the distances are constituted a partial – order category, which has the smallest distance and the biggest distance. The objects in the category are the different distances, and the arrows are from the smaller distance to the next much smaller distance. In the same way, the distances between every measured point and the reference line after the filtration are constituted a partial – order category as well, as shown in figure d).

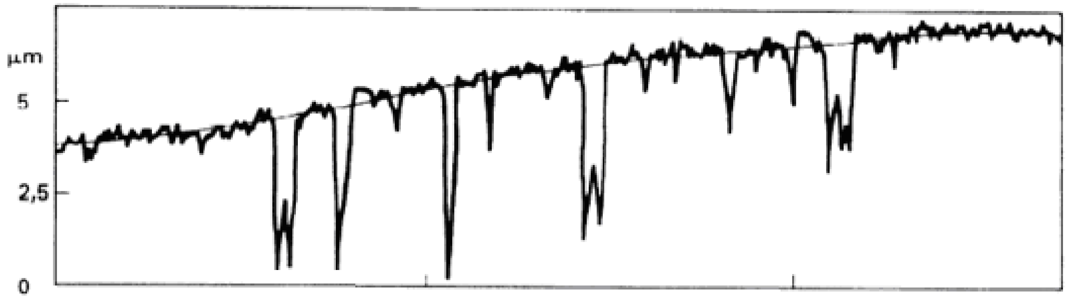
The mapping from the above two stages – before and after filtration satisfies the functor definition, as it is a mapping between two categories, and the relational structures of those two categories are preserved. Therefore, the procedure filtration is a stable measurement procedure according to the stability corollary: the relational structures of the roughness profile before and after the filtration processes are both partial – order categories and the mapping between them is a functor.



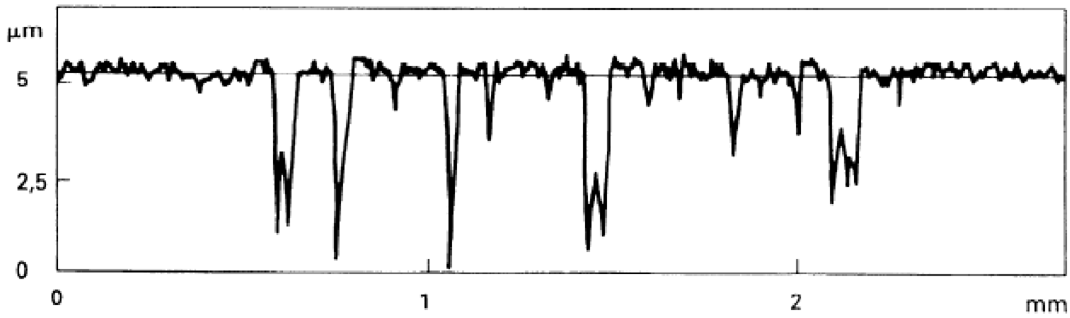
a) Unfiltered primary profile (valleys shown hatched)



b) Unfiltered primary profile after suppression of valleys



c) Position of the reference line in the primary profile



d) The roughness profile

Figure 3.10 Filtering processes [13]

3.3 Category model approach in knowledge representation

Different data models for knowledge representation have been discussed in chapter 2, and a new data model is to be proposed in this research in order to satisfy the requirements. Category model is then introduced and applied, which generalises the relational and object-oriented data models.

3.3.1 Introduction of Category model approach

Category theory provides a formal basis and abstracts from all types of the representation. It has the ability to combine diagrammatic formalisms as in geometry with symbolic notation as in algebra [15]. The category model, based on category theory, is a highly abstract data model, which can bring together different data models and provide a common structure for describing data. It has both the flexibility of structures for entities and has good mathematical foundations that ensure the knowledge-base remains a coherent and reliable system as new knowledge is added.

The fundamental constructs in category theory are objects and arrows between objects, which is similar to the entities and functions in the functional data model (see Chapter 2). Because the functional data model provides what is probably the most natural query language DAPLEX, based on function composition, it is intended that a manipulation language can be developed based on DAPLEX, thereby providing a conceptually natural query language for the category model.

3.3.2 Category Theory

Category theory is a general mathematical theory of structures and systems of structures and mappings that preserve structures. It allows one to see, among other things, how structures of different kinds are related to one another as well as the universal components of a family of structures of a given kind. The theory is philosophically relevant in more than one way. For one thing, it is considered by many as being an alternative to set theory as a foundation for mathematics.

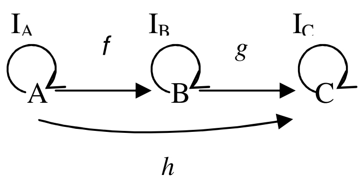
Category theory is a very young subject, appearing almost out of nowhere in 1945 in Eilenberg & Mac Lane's paper entitled “General Theory of Natural Equivalences”. Much fundamental work was completed in the 1950’s, 60’ & 70’s. From the 1980s to this day, category theory has found new applications. It now has many applications in theoretical computer science where it has firm roots and contributes, among other things, to the development of the semantics of programming and the development of new logical systems. Further category theory is being used in the research and design of next generation databases.

The following describe some basic terms in the category theory which will be used in the category model (for a full description please see the reference Benjamin C Pierce [1]).

3.3.2.1 Category

As defined in section 3.2.3, the two fundamental entities in Category Theory are arrows and objects. A category consists of a class of objects and a class of arrows such that classes of arrows are associated with every pair of objects (could be the empty class). The arrows are associative under composition and every object has an identity arrow. For example, category ***P*** consists of three objects A, B and C together with three arrows *f*, *g and h* and the associated identity arrows of A, B & C I_A, I_B & I_C respectively, see figure 3.11 [1].

***P*:**



$f: A \rightarrow B \quad g: B \rightarrow C \quad h: A \rightarrow C$

Such that h = g ○ f

Figure 3.11 Category *P*

3.3.2.2 Products

The product in category theory is the generalization of the mathematical concept of product, e.g. Cartesian product A x B. A product of two objects A and B is an object A x B, together with two projection arrows π_1: A x B → A and π_2: A x B → B, such that for any object C and pair of arrows *f*: C → A and *g*: C → B there is exactly one mediating arrow < f, g>: C → A × B making the diagram commute — that is, such that π_1 ○< f, g> = f and π_2 ○< f, g> = g, see figure 3.12. (See Benjamin 1991 [1] for more details)

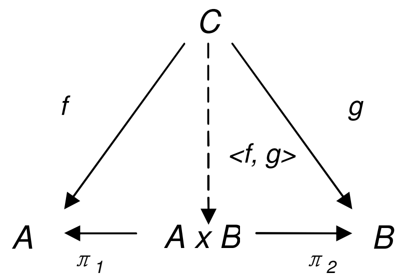


Figure 3.12 Product cone

Figure 3.13 is an example of a product. Here an arrow means "is a subset of". The product of two sets A, B then becomes the intersection of the two sets, i.e. A ∩ B is the product of the object A and B, since A ∩ B ⊆A & A ∩ B ⊆B, and A ∩ B is the largest set to have these properties. Thus, if there is another set C with the same properties, then C ⊆A & C ⊆ B, and so C ⊆A ∩ B completing the definition of a product.

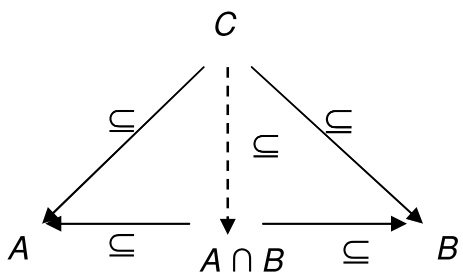


Figure 3.13 An example of a product

3.3.2.3 Pullbacks

A pullback of the pair of arrows *f*: A → C and *g*: B → C is an object P and a pair of arrows g': P → A and f': P → B such that $f \circ g' = g \circ f'$, and if i: X → A and j: X → B are such that $f \circ i = g \circ j$, then there is a unique k: X → P such that $i = g' \circ k$ and $j = f' \circ k$. (See Benjamin 1991 [1] for more details)

Figure 3.14a gives a basic structure that connects two entities A and B with two arrows f: A → C and g: B → C, the pullback structure identifies the relationships between A and B as defined in the basic structure, see figure 3.14b, object P represents the relationships between A and B.

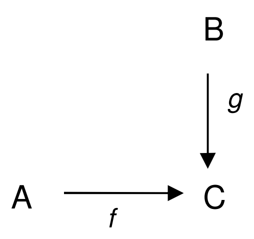


Figure 3.14a Basic structure: connects entities A & B

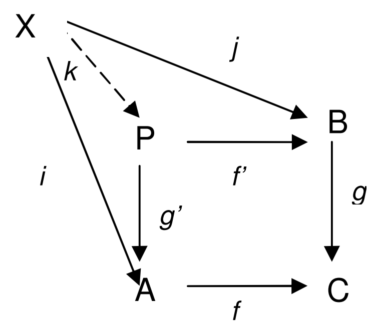


Figure 3.14b The pullback structure: finds relationships between entities A & B contained within the basic structure

Figure 3.14 Pullback

The following is an example. If A and B are subsets of the set C, then figure 3.15 is an illustration of the associated pullback structure [1], here the arrows are the same as in figure 3.13, A ∩ B is the product of the object A and B, thus, if there is another set X, X ⊆ A & X ⊆ B, then X ⊆ A ∩ B, A ∩ B is the largest set with this property.

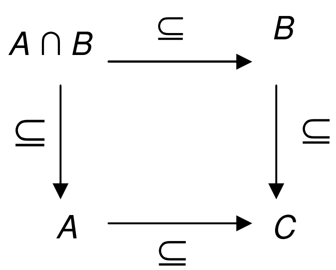


Figure 3.15 An example of a pullback

3.3.3 Category model

The category model is based on category theory described above. It generates both the relational model and object-oriented data models (see Chapter 2). Since the fundamental constructs in category theory are objects and arrows between objects, which is similar to the entities and functions in the functional data model. And the functional data model provides what is probably the most natural query language DAPLEX [16], based on function composition, it is intended that a manipulation language can be developed based on DAPLEX, thereby providing a conceptually natural query language for the category model. The biggest advantage of the category model is its stronger framework in the categorical approach for multi-level constraints, for inter-object relationships and for functional dependencies.

The table 3.2 below compares some important issues of the relational model, functional model and category model [15].

	Relational model	**Functional model**	**Category model**
Inter-object associations	Join on primary/foreign keys	Functions	Pullbacks
Intra-object associations	Functional dependencies	N/A	Arrows within category
Integrity constraints	Referential/Entity integrity	Single/Multi valued functions	Arrow types, object types
Keys	Primary/foreign key definitions	N/A	Initial objects
Relational join	Relational algebra	Function composing entities	Pullback/product

Table 3.2 Comparison of data models

3.3.3.1 Relationships in Category Model

In category theory, the E-R model can be represented by pullbacks, as in figure 3.16. Object P represents the relationships between object A and B.

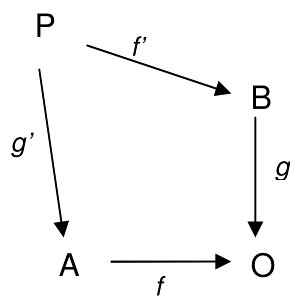


Figure 3.16 Relationship as a Pullback

Figure 3.17 gives an example of how to generate the table relations into a category model – pullback. P and Q represent different entities, M is a structure used to store all the possible relations and extra information between P and Q, N is the restricted product relationship, i.e. the table relations between entity P and Q. A tick "√"in the table means a relationship between the two objects in P and Q, which is generated into a pullback structure in the category model. For example, there is a tick between “A” and “1” in the table, thus, in the category model, there is a pullback structure connecting them together, “A1” is the product of “A” and “1”, as indicated by the dotted arrows in figure 3.17. If there is no tick between two objects in the table, then there is no pullback structure in the category model between them and vice versa.

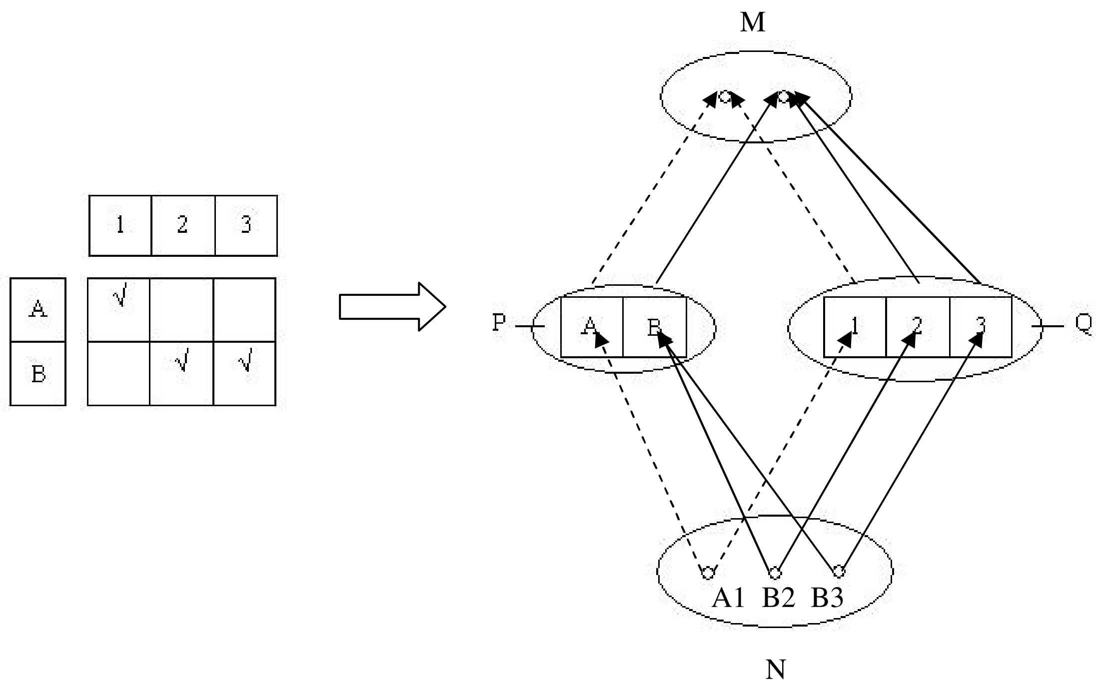


Figure 3.17 A simple Table Relationship put into an equivalent Pullback Structure

3.3.3.2 Comparison of category model examples with other data models

The relational data model

Table 3.3 and table 3.4 give an example of relational database *studying*. Table 3.3 includes the integral and redundant information contained in the database. And table 3.4 shows a normalised relational data model representing a set of simple relations that captures the same information as table 3.3.

It can be seen that in order to represent the one-to-many relationship in table 3.3, extra table ‘Relation R3 Major’ has to be defined. This situation is applied to many-to-many relationships as well. It is one of the main drawbacks with the relational model.

ssn	name	bdate	sex	dname	dphone	doffice
001	David	25-02-1980	Male	Computer science	3468	1/23

002	Mary	18-04-1981	Female	Computer science	3468	1/23
003	Fred	02-11-1984	Male	Computer science	3468	1/23
004	Peter	18-09-1982	Male	Engineering	2587	2/15
005	Andrew	06-07-1981	Male	Engineering	2587	2/15

Table 3.3 The relation *studying*

Relation R1 Student			
ssn	name	bdate	sex
001	David	25-02-1980	Male
002	Mary	18-04-1981	Female
003	Fred	02-11-1984	Male
004	Peter	18-09-1982	Male
005	Andrew	06-07-1981	Male

Relation R3 Major	
ssn	dname
001	Computer science
002	Computer science
003	Computer science
004	Engineering
005	Engineering

Relation R2 Department		
dname	dphone	doffice
Computer science	3468	1/23
Engineering	2587	2/15

Table 3.4 A set of simple relations

The functional model

Figure 3.18 shows the equivalent functional model of *studying*. It differs from the relational model, as it is no longer required to create extra tables to manage the one-to-many and many-to-many relationships, see the corresponding data definition statement in the figure 3.18 [15].

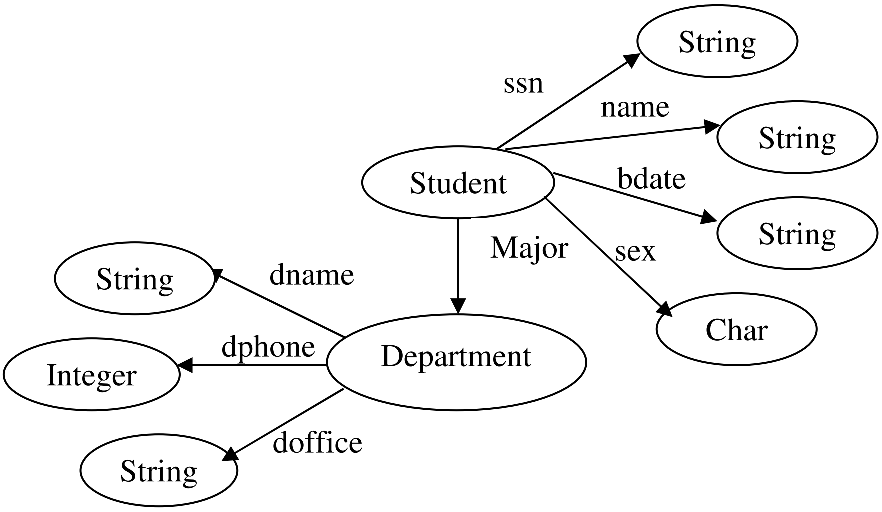
It is mentioned before that the functional data model provides what is probably the most natural query language DAPLEX. For example:

Print the names of all students who have majored in the department of Engineering:

FOR EACH Student SUCH THAT dname(major(Student)) = 'Engineering'

PRINT name(Student)

The functional model avoids the drawback of the relational model and also gives a more natural representation in the way human models the world. However, from the figure 3.18, the flatness of the model would become a problem for abundant knowledge and complex relations. In that case, the functional model would be hard to recognise and maintain.




```
declare Student() ⇒> ENTITY
declare ssn(Student) ⇒ STRING
declare name(Student) ⇒ STRING
declare bdate(Student) ⇒ STRING
declare sex(Student) ⇒ STRING
declare major(Student) ⇒ Department

declare Department() ⇒> ENTITY
declare dname(Department) ⇒STRING
declare dphone(Department) ⇒ INTEGER
declare doffice(Department) ⇒ STRING
```

Figure 3.18 The functional diagram of *studying* with corresponding data definition statements

The category model

The category model of *studying* is given in figure 3.19 [15]. It includes two objects: Student and Department, and with their own attributes. 'ssn' and 'dname' are both the key attributes and represented by initial object[2] in category theory. The relationship between Student and Department is represented in Pullback structure. p0 is a structure used to store all the possible relations and extra information between Student and Department, ssn $\times_{p0}$dname is the restricted product relationship, i.e. the table relations 'major' between entity Student and Department. The arrows are typed according to the functionality and membership classes attributed to the relationship.

[2] Initial object: an object O in category C is called an initial object of C if, for every object A in C, there is exactly one arrow from O to A.

The arrows in the category model are similar with the functions in the functional model. Moreover, the category model clearly distinguishes between inter and intra object relationships, which makes the diagram more readable and manageable.

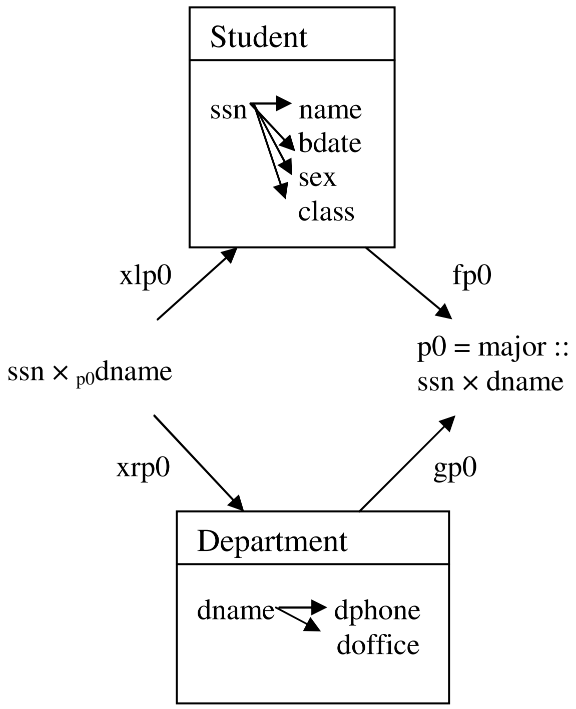


Figure 3.19 The category model of *studying*

3.3.3.3 Generate production rules to category model

Figure 3.20 gives an example of how to represent production rules using pullback structure. The production rule is used to resolve the problem that output the studying hours per week for a student majors in a certain department and a certain year. This particular rule is generated into an equivalent pullback structure.

P1 is the product of three categories "Student", "Department" and "Section", where *studying_week :: ssn*dname*year* is the name and type of the product, $ssn*_{p1}dname*_{p1}sec\#$ is the restricted product, *xlp1* ,xmp1 and *xnp1* are the projections of the product into the initial objects of the "Student", "Department" and "Section" categories respectively, and *fp1,gp1 and hp1* are functions injecting the initial objects into the pool of values of the product.

This example shows the inference ability of the category model, which would be applied as part of the inference engine within the knowledge-based system.

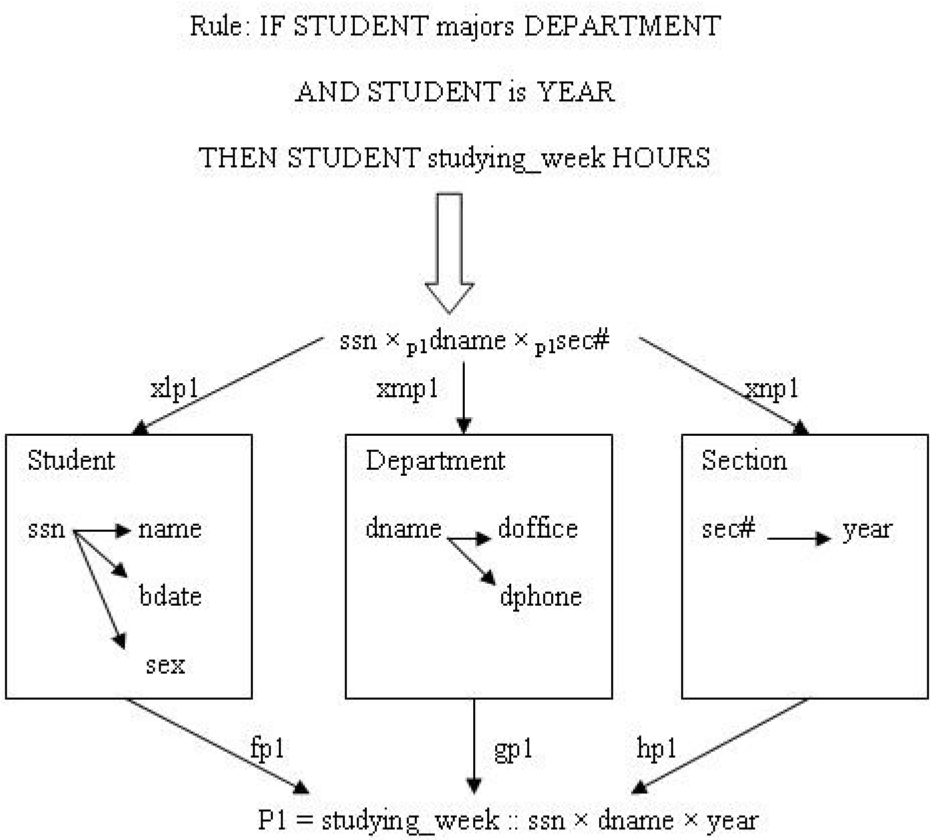


Figure 3.20 Production Rules put into an equivalent Pullback Structure

3.3.3.4 The queries for the category model

According to the DAPLEX queries of the functional model, the natural queries of the category model would be developed. An example is given as follows [15]:

Print the names of all students who have majored in the Engineering department:

[Hom-sets are defined as a set of arrows, each arrow being specified by its name or its source and target.

P0 is the pullback category of the relationship p0, including all arrows in the pullback, the product p0 and the restricted product, and the initial objects of the categories that the pullback is between.]

A → P0

Hom set = xlp0

Objects = ssn $\times_{p0}$dname | dname = ‘Engineering’

B → A

Hom set = { }

Objects = ssn

$C \rightarrow$ Student

Hom set = ssn $\rightarrow$ name

Objects = ssn, name | ssn $\in$ B

3.4 Model refinement applies Category theory

Model refinement is used to convert a simplified (Coarse-grained) abstract (high-level) data model into a complex implementable (low-level) data model. Figure 3.21 gives an illustration of data refinement. On platform E there exists the abstract entities A and B and the abstract relationship between them; while on platform C there exists the concrete data structures X and Y and the relationships between them. P is the abstract procedure for the refinement process.

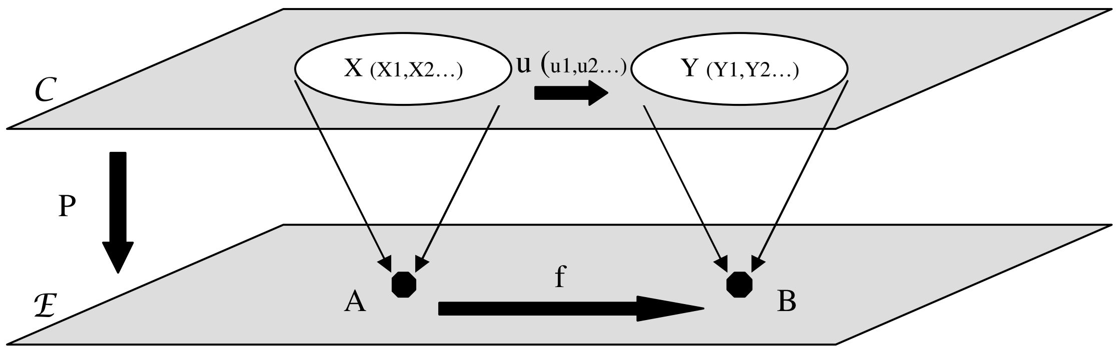


Figure 3.21 Illustration of Model Refinement

Starting with simplified models of the knowledge-base, more complex models can be developed and checked by the process of refinement to ensure consistency of both models. In this way a hierarchy of consistent models can be built from the simple to the complex, as figure 3.21 shows. Simplified abstract models are used to represent general frameworks of a knowledge-based system, which might involve many entities and relationships when it is refined. Abstract model may not be able to incorporate all the necessary information, such as relationships, rules, etc, which would be a great problem for programming. It is necessary to carry out successive refinements of abstract model to implementable models, which contains all the entities, relationships and rules needed for a knowledge-based system. This complex model includes necessary information for software engineers to realise the integrated system.

Model refinement is used in the design of knowledge-based systems. One of the most difficult problems in the development of knowledge-based systems is the construction of the underlying knowledge-base. As a result, the rate of progress in developing knowledge-based systems is directly related to the speed with which knowledge-bases can be assembled [17]. Therefore, the development of knowledge acquisition plays an important role in the system design. Normally knowledge acquisition can be divided into two stages: an initial stage knowledge extraction, in which a knowledge engineer extracts a rough set of data information from the references, books, experts and all kinds of resources, and the second stage knowledge base refinement, in which the initial knowledge base is refined to produce a high-performance system [18].

3.4.1 Requirements for refinement

In order to secure the consistency between simplified abstract model and complex implementable models, there are nine requirements that should be matched according to Colomb 2001[19].

Assume that simplified abstract model is named E and a particular complex implementable model is named C, P is the abstract procedure for the refinement process from E to C. A and B are the entities in E and f is a relation in E between A and B, X and Y are the entities in C and u is the relation in C between X and Y.

Requirement 1:

"R1a – every entity in the implementable model is above an entity in the simplified abstract model;

R1b – every entity in the simplified abstract model has at least one entity in the complex implementable model above it (refinement)".

For example, consider figure 3.19, entity X in C is above the entity A in E and entity Y in C is above the entity B in E (i.e. P(X)=A and P(Y)=B)

Requirement 2:

"R2 – If there is a relationship in the complex implementable model between two entities X and Y, then there is a relationship in the abstract model between the entities A and B, where X is above A and Y is above B. Moreover, transitive relationships in the complex implementable model are preserved in the abstract model."

For example, the relationship u between X and Y in C is above relationship f between A and B in E (i.e. P(u)=f).

Requirement 3:

"R3 – If there is a relationship in the abstract model, there must be at least one relationship in the complex implementable model above it."

The relationship u in C may incorporate several different relations between X and Y, such as u1, u2 and etc, and all of which are above the relationship f between A and B in E.

Requirement 4:

"R4 – If there is a relationship in the abstract model, and there is a derived relationship obtained by composing a second (connecting) relationship with the first, then there must be a relationship in the complex implementable model above the derived relationship which is derived form relationships above the first and connecting relationships."

$$
\begin{tikzcd}
\mathcal{C}: & X \arrow[r, "u"] \arrow[rr, bend left, "w"] \arrow[d, "P"'] & Y \arrow[r, "v"] \arrow[d] & Z \arrow[d] \\
\mathcal{E}: & A \arrow[r, "f"] \arrow[rr, bend right, "h"'] & B \arrow[r, "g"] & C
\end{tikzcd}
$$

Figure 3.22 Illustration of Requirement 4

Consider the models in figure 3.22. In the abstract model E, there is a relationship f between A and B and a relationship between B and C, thus, there is a derived relationship h between A and C. X, Y and Z in the complex implementable model C are above A, B and C respectively, and u and v are above the relationships f and g in E. Therefore, there must be a derived relationship w between X and Z above the derived relationship h (i.e. P(w) = h).

Requirement 5:

"R5 – If a set of relationships in the abstract model satisfies the principle of consistent dependency, then every set of relationships above them must also."

R5 is a consequence of R4. Consider the models in figure 3.23, in the abstract model E, there are two equivalent indirect relationships from A to C via B1 and Via B2, which both satisfy the principle of consistent dependency (Dependencies are consistent providing object is consistent (and vice-versa)). Therefore, in the complex implementable model C, there are two corresponding derived relationships above each of them w1 and w2 from X to Z via Y1 and Y2.

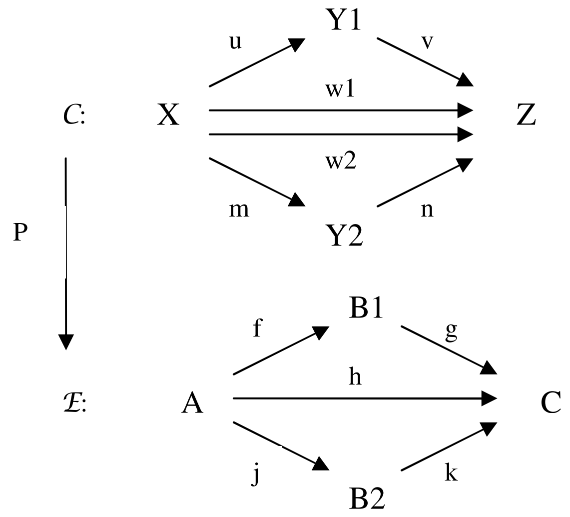


Figure 3.23 Illustration of Requirement 5

Requirement 6:

"R6 – The solutions to R4 must be equivalent."

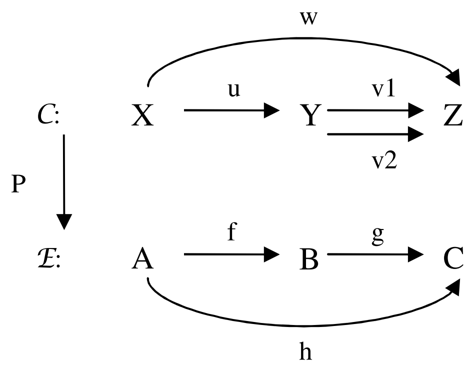


Figure 3.24 Illustration of Requirement 6

Consider the models in figure 3.24, there are two distinct relationships connecting Y and Z in C, which achieves the same derived relationship w between X and Z. According to requirement 6, these two relationships should be equivalent.

Requirement 7:

"R7B – Every entity above B must participate in a relationship above f.

R7F – Every entity above A must participate in a relationship above f."

Consider figure 3.21, the entities X1, X2, X3… in C which are above A in E, participate in the relationships u1, u2, u3… which is above the relationship f, with the entities Y1, Y2, Y3…which are above B in E.

Requirement 8:
"R8 – At least one complex implementable entity can be selected above each abstract model entity such that they participate in the same pattern of relationships as the abstract model."

Requirement 9:
"R9 (informal) – Every entity in the implementation model participates appropriately in such a pattern."

The above nine requirements secure the consistency between the abstract model and complex implementable models. It is necessary to consider and match these requirements when designing and developing the knowledge-base refinement. These requirements can be concluded into four words as follows [20]:

a) Integrity (requirement 1-3): every entity in the implementable model is above an entity in the abstract model, every relationship in the implementable model is above a relationship in the abstract model, and every entity and relationship in the abstract model is refined.
b) Composition (requirement 4-6): every composition of relationships in the abstract model is refined by a composition of refinements of the relationships composed.
c) Completeness (requirement 7): every entity refining a abstract model entity is the target of a relationship refining a relationship whose target is the abstract model entity.
d) Pattern reservation (requirement 8-9): every entity and relationship in the implementable model is part of a pattern with the same structure as the abstract model.

3.4.2 Category model for data refinement process

There have been much research work on data (model) refinement using category theory during the past ten years, such as the approaches mentioned in Naumann 2001 [21]: how natural transformations model data refinement for an imperative language, a particular class of non-higher order systems including modern database systems, and higher order functional and imperative languages.

The Colomb 2001 [18] adopts category theory - fibration as the abstraction mechanism, and introduces how the fibration model can be used for data refinement for data models in information systems. This is a more appropriate way to explain model refinement of knowledge-bases using category theory. This thesis applies the method being developed in Colomb 2001 [19], which provides an abstract formulism for the refinement process of Virtualsurf knowledge-based system.

3.4.2.1 Category theory applied to refinement process

The requirements for refinement have been stated in last section. This section then converts these requirements into the language of Category theory [19].

Since a category is a collection of objects and arrows between these objects, C and E would be regarded as categories. For example in figure 3.25, category E consists of two arrows f and g: [15].

$$E\text{:}\quad A \xrightarrow{f} B \xrightarrow{g} C$$

$$f\text{: } A \rightarrow B \quad g\text{: } B \rightarrow C$$

Figure 3.25 A category example

The refinement process P between C and E satisfies the functor concept in category theory. A functor is a special type of mapping between categories in category theory.

Let C and E be categories. A functor P from C to E is a mapping that

- Associates to each object $X \in C$ an object $P(X) \in E$,
- Associates to each morphism $u: X \rightarrow Y \in C$, a morphism $P(u): P(X) \rightarrow P(Y) \in E$

Such that the following two properties hold:

- $P(id_X) = id_{F(X)}$ for every object $X \in C$
- $P(v \circ u) = P(v) \circ P(u)$ for all morphisms $u: X \rightarrow Y$ and $v: Y \rightarrow Z$.

That is, functors must preserve identity morphisms and composition of morphisms.

Fibration is a special form of functor. In the category of small categories Cat, a functor $P: C \rightarrow E$ from a category C to a category E is a **fibration** if and only if for every object Y of C and every map f: A → P(Y) in E there exists a cartesian morphism u: X → Y in C above f (i.e. P(u) = f). In this case, C is also said to be a category fibred over E or a fibred category, see figure 3.26.

$$\begin{tikzcd} \mathcal{C}: & X \arrow[r, "u"] \arrow[d, "P"'] & Y \arrow[d] \\ \mathcal{E}: & A \arrow[r, "f"] & P(Y) \end{tikzcd}$$

Figure 3.26 A fibration

Cartesian [22]: Let P: $\mathcal{C} \to \mathcal{E}$ be a functor between small categories, let f: C → D be an arrow of E, and let P(Y) = D. An arrow u: X → Y of C is **cartesian** for f and Y if P(u) = f; For any arrow v: Z → Y of $\mathcal{C}$ and any arrow h: P(Z) → C of $\mathcal{E}$ for which f ∘ h = P(v), there is a unique w: Z → X in $\mathcal{C}$ such that u ∘ w = v and P(w) = h, see figure 3.27.

$$\begin{tikzcd} & & Z \arrow[dl, "w"'] \arrow[d, "v"] \\ \mathcal{C}: \arrow[ddd, "P"'] & X \arrow[r, "u"] & Y \\ & & P(Z) \arrow[dl, "h"'] \arrow[d, "P(v)"] \\ \mathcal{E}: & C \arrow[r, "f"] & D = P(Y) \end{tikzcd}$$

Cartesian arrow: u

There is a unique w in $\mathcal{C}$:
u ∘ w = v
and P(w) = h

Figure 3.27 Cartesian arrow

The refinement requirements can be implemented using those category theoretic concepts described above.

It has shown in Colomb 2001 [19] that since both the abstract model E and complex implementable model C are categories, and refinement process P is a functor, requirements 1-3 (Integrity) are automatically satisfied for the definitions of a category and a functor. Further requirements 4-7 (Composition and Completeness) would also be satisfied if functor P is a fibration.

3.4.2.2 Fibration checking rules

It is necessary to check whether the refinement process P between C and E is a fibration, in order to prove the consistent mapping between the simplified abstract model and the complex implementable model.

The following is the Checking rules for a fibration as given in Colomb 2001 [19]:

"To check whether P: C → E is a fibration,

F1: check every arrow in E

F2: for each arrow f: C → B in E, check for each object Y above B that there is a Cartesian arrow c above f whose target is Y

To check whether an arrow c: Z → Y is Cartesian,

C1: check the composite arrow fh for every h in E with target (h) = source (f)

C2: for each fh we must check all the arrows v above fh with the same target as c, there must be at least one such arrow

C3: for each v, there must be a unique w above h such that cw = v. break this condition into two

C3E: a w exists

C3U: the w is unique"

3.4.2.3 Proposition on fibration functors

The following are several useful propositions about fibration as developed in Colomb 2001 [19]. The propositions are helpful while checking and deciding whether the functor is a fibration in practise. Assume the functor P: C → E is a fibration, C is defined as its finer category and E as its coarser category.

"Proposition 1: The constant functor is a fibration."

The functor whose codomain (coarser category) is the category **1**(one object only with only the identity arrow) is called the constant functor [19].

Proof, see figure 3.28, P is the functor between C and E, A is the only object in E, and 1A is the identity arrow of A:

F1: all the arrows in E are 1_A

F2: Y is the object above A in C, and 1_Y is the identity arrow of Y

C1: f = 1_A, and h = 1_A, the composite arrow fh in E is $1_A 1_A = 1_A$

C2: the arrow c in C equals arrow 1_Y, and there exists arrow v: X → Y with the same target Y as c. Since P(X) = A, P(v) = 1_A, there is at least one such arrow: v = 1_Y
C3E: since cw = v, c = 1_Y => w = v, there exists a w
C3U: if there is another w' above h = 1_A, then cw' = v => w = v, thus w' = w, the w is unique.

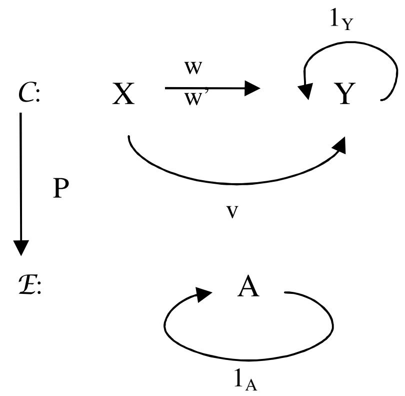


Figure 3.28 The constant functor

"Proposition 2: The identity functor is a fibration."
It is suggested that the complex implementable model can be partially abstracted, and leave the rest unchanged.

As mentioned in section 3.4.2.2, it is necessary to check every arrow in E in order to check a fibration. It is convenient to classify such arrows into three classes and consider the propositions below:
"A1: the identity arrows
A2: the non-identity arrows, where C1 is satisfied by the identity on the source.
A3: the non-identity arrows, where C1 is satisfied by a non-identity arrow."

"Proposition 3: Checking A1 can never lead to a functor failing to be a fibration."
The following propositions are used to check A2. Assume f: C → D is in E and Y is above D, C(Y) is the subcategory above C whose objects are the source of arrows whose target is Y.

"Proposition 4: If A2 is satisfied, C(Y) is not empty for all Y above D."

"Proposition 5: If p: X → Y is a Cartesian arrow above f, then X is a terminal object of C(Y)."

An object M of a category C is terminal if for every N in objects of C, $\exists !\ f: N \to M$, see figure 3.29, M is the terminal object.

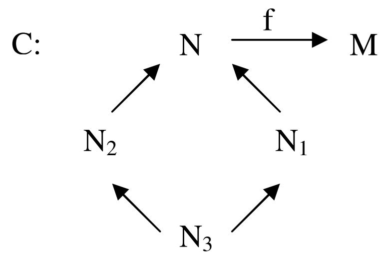


Figure 3.29 Terminal object

“Proposition 6: If there are two parallel arrows p, p’: $X \to Y$ and p is a Cartesian arrow, then C(Y) has a non-identity endomorphism.
An endomorphism is an arrow whose source and target are the same object. For example, An identity arrow is an endomorphism.

“Proposition 7: If (i) C(Y) has a terminal object X, and (ii) there are no parallel arrows from X’ to Y where X’ in C(Y), then p: $X \to Y$ satisfies the condition A2.”

“Corollary 8: If P is the identity functor on subcategory C’ of C, then every arrow in C’ satisfies A2.”

“Corollary 9: If the arrows above f are derived from an arrow p whose source is a terminal object of the subcategory above C and whose target is an initial object of the subcategory above D, A2 is satisfied.”

An object M of a category C is initial if for every N in objects of C, $\exists !f: M \to N$, see figure 3.30, M is the initial object.

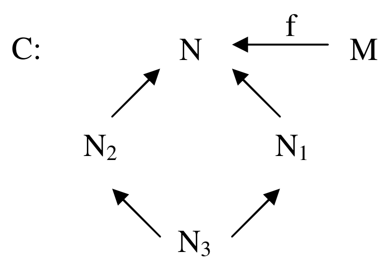


Figure 3.30 Initial object

“Proposition 10: If each arrow of a commuting diagram in E passes A2, and there are no non-identity endomorphisms in C, then if a commuting diagram in C passes C3E it also passes C3U.”

3.4.2.4 Refinement guidelines

The methods and Propositions described above can be concluded into five guidelines for the refinement of data models [20]:

Guideline 1: Freedom within a fibre. The fibration does not itself place any restrictions on the refined model fragments within a fibre. This is because the constant functor is always a fibration (Proposition 1).

Guideline 2: Connections between fibers must be unitary (requirements integrity and composition). A key element in the definition of a fibration is that there must be a Cartesian relationship above every relationship in the abstract model for each entity above the target.

Guideline 3: every entity above an entity in the abstract model which is the target of a relationship must be the target of a relationship in the implementation model satisfying guideline 2 (requirement completeness).

Guideline 4: if there is a derived relationship in the abstract model, relationships above it must be derived from the Cartesian relationships of guideline 2.

Guideline 5: if there are diverging relationships in the abstract model, then there must be relationships in the implementable model above them whose centres are aligned (requirement pattern preservation).

3.4.2.5 Adjoint and the refinement requirements

The fibration has been discussed in the last few sections and proved to satisfy the refinement requirements 1-7, but is not sufficient for requirements 8-9 (Pattern preservation). The concept of adjointness is used in this thesis to solve the problems and provide an easier explanation for the refinement requirements. Adjoint functors are used to establish duality relationships between two spaces. In this thesis, it provides the refinement of complex implemetable models to simplified abstract models. In mathematics, adjoint functors are pairs of functors which stand in a particular relationship with one another [22].

An illustration of adjoint functors is shown in figure 3.31, which consists of:

- A pair of categories C and E
- A pair of functors F: E → C and G: C → E
- A natural transformation $\eta(X)$: $I_E \xrightarrow{\cdot} (G \circ F)$
- A natural transformation $e(Y)$: $(F \circ G) \xrightarrow{\cdot} I_C$

Note : $\xrightarrow{\cdot}$ natural transformation

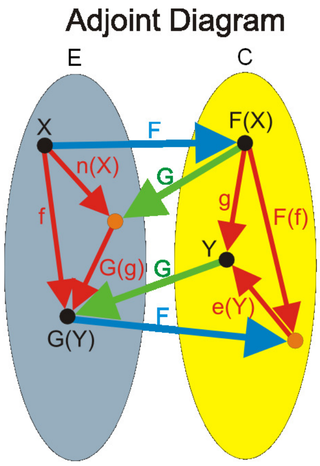


Figure 3.31 Adjoint diagram

Such that for each E-object X and E-arrow f: $X \rightarrow G(Y)$, there is a unique C-arrow g: $F(X) \rightarrow Y$ such that the following triangle commutes, i.e.: $f = G(g) \circ \eta(X)$, see figure 3.32:

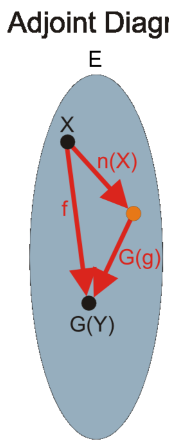


Figure 3.32 Adjoint condition 1

In the same way, for each C-object Y and C-arrow g: $F(X) \rightarrow Y$, there is a unique E-arrow f: $X \rightarrow G(Y)$ such that the following triangle commutes, i.e.: $g = e(Y) \circ F(f)$, see figure 3.33:

(F, G) is an adjoint pair of functors; F is the left adjoint of G and G is the right adjoint of F [23].

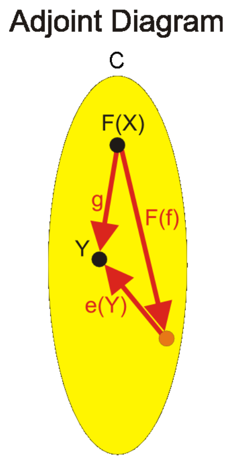


Figure 3.33 Adjoint condition 2

An example of adjoint functors is shown in figure 3.34, which consists of:

- A pair of categories C and E with the usual ordering: E is the set of integers Int = (X,≤), and C is the set of Real numbers Real = (Y,≤).
- A pair of functors:

Inclusion F: E → C, which means E is a subset of C, i.e. for each element x of E, F(x) = x;
Ceiling function G ([r]): C → E, which takes each r∈ Y to the smallest integer greater than or equal to r — that is, r ≤ r' implies that [r] ≤ [r'], where "≤" in each case stands for an arrow.

- Such that for each E-object [r] and E-arrow ≤: [r] → [r'], there is a unique C-arrow ≤: r → r', such that the following triangle commutes, i.e.: ≤ = ≤ ∘ ≤.
- In the same way, for each C-object r and C-arrow ≤: r → r', there is a unique E-arrow ≤: [r] → [r'] such that the following triangle commutes, i.e.: ≤ = ≤ ∘ ≤.

(*F*, *G*) is an adjoint pair of functors; *F* is the left adjoint of *G* and *G* is the right adjoint of *F*.

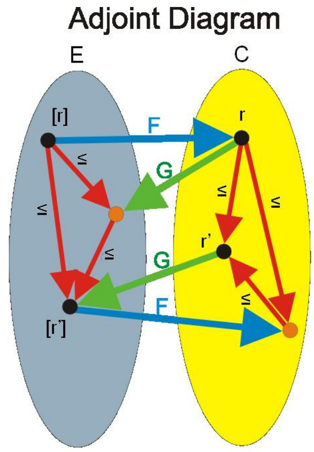


Figure 3.34 Example of adjoint functor

Define category E is the simplified abstract model, and category C is the complex implementable model, functor F is the refinement process, functor G is the abstract procedure for the refinement process. The refinement requirements 1-9 are here proved and restated using the concept of adjoint functors as follows:

Requirements 1 – 3 are automatically satisfied by the functor definition.

Requirement 4: see figure 3.31, in the abstract model E, there is a derived relationship $G(g) \circ \eta(X)$, while in the implementable model C, there exists a first relationship I_E above $G(g)$, and a second relationship g above $G(g)$:

$$G(g) \circ \eta(X) = f,$$
$$g \circ I_E = g = e(Y) \circ F(f)$$

Therefore, the derived relationship $g \circ I_E$ in category C would above $G(g) \circ \eta(X)$ in category E, iff $e(Y) = I_Y$. Then $e(Y) \circ F(f) = F(f)$, and is above the derived relationship f. Figure 3.35 gives the refinement adjoint diagram, which requires that $e(Y) = I_Y$.

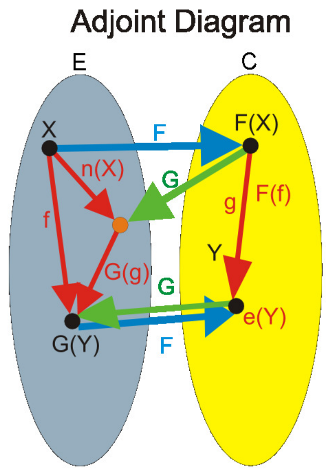


Figure 3.35 Refinement Adjoint

Requirements 5 – 7 are satisfied because of *g* is the unique according to the adjoint definition.

Requirement 8 and 9: from figure 3.32 and figure 3.33, the entity X participates in the pattern of relationship that $f = G(g) \circ \eta(X)$, there exists a unique *g* in category C to allow the entity F(X), which is above X, to participate in the same pattern of relationship that $g = e(Y) \circ F(f)$, the same as the entity Y in category C.

A special adjointness – refinement adjoint functors in figure 3.34 are proved to satisfy all the 9 requirements for the refinement processes. Since adjoint has the transitive properties [23], i.e., functor F and G between category E and C are a pair of adjoint functors, functor H and I between category C and M are also a pair of adjoint functors, then functor H ∘ F and G ∘ I between category E and M are a pair of adjoint functors, see figure 3.36.Therefore, the adjoint functors would secure the transition of the refinements processes.

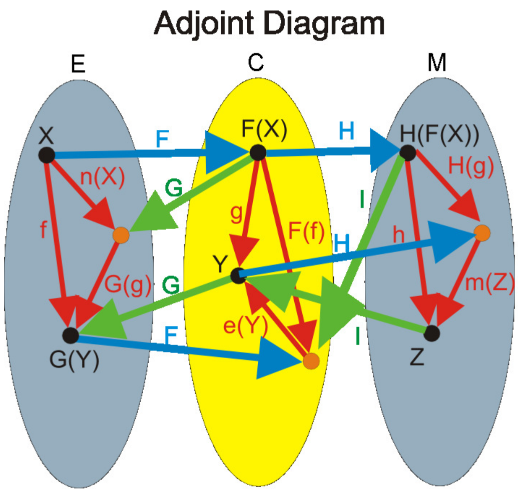


Figure 3.36 Transition of adjoints

3.4.2.6 Example

This section gives an example to illustrate the refinement process in the knowledge-base refinement.

A special adjointness – refinement adjoint functors in figure 3.37 are proved to satisfy all the 9 requirements for the refinement processes, which requires that e(Y) = IY , i.e. e(Y) is always an identity arrow.

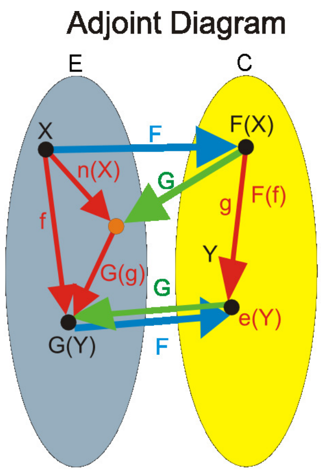


Figure 3.37 Refinement Adjoint

This is refinement because this adjoint functor partitions the space E (all the elements of E that map onto the same object in C) and each partition is then refined down to a single element in C. The Category in C preserves the relationships between the partitions in E and thus is a simplified model of the complex model in space E.

Figure 3.38 gives an abstract model representing a simplified Specification knowledge-base, which includes three operations: Partition, Extraction and Filtration. The Extraction process determines the Filtration process, and the Measurand determines both the Extraction process and the Filtration process. Figure 3.39 gives the complex model which is refined from the abstract model, with added detailed and necessary information.

These two models form a pair of adjoint functors, with funtor F from abstract model to complex model is a refinement, and functor G from complex model to abstract model is a projection. P#, E#, F# and M# in the complex model are initial objects with unique arrows to the related objects, and mapping to the corresponding entities in the abstract model. These mappings satisfy the definition of the refinement adjoints and therefore, the refinement processes are proved to be consistence.

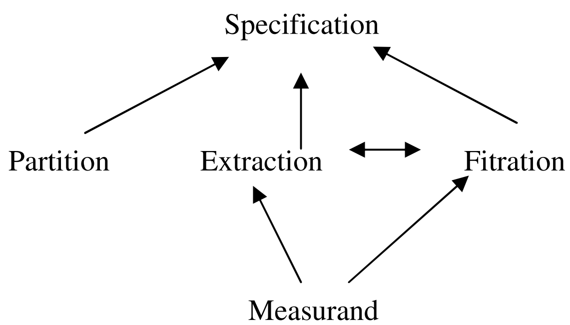


Figure 3.38 Abstract model of the Specification

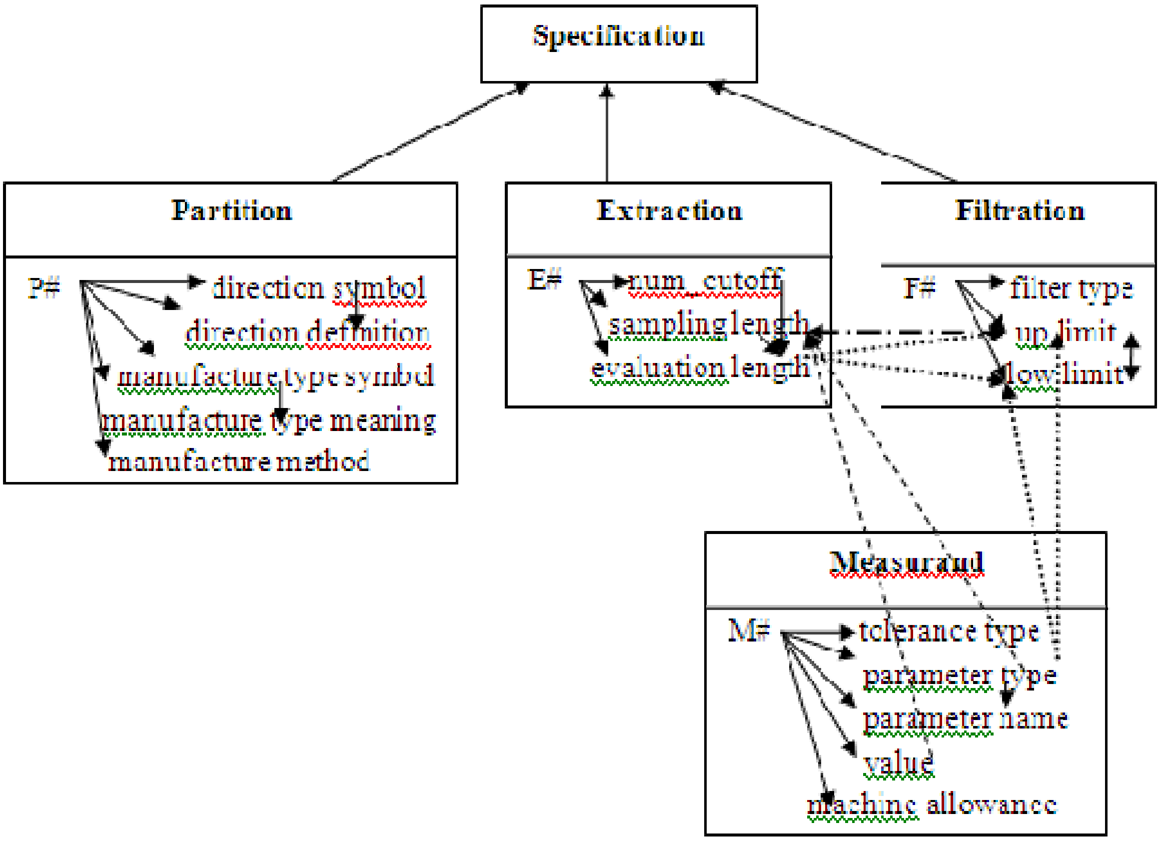


Figure 3.39 Concrete model of the Specification

3.5 Summary

Three main applications of Category theory have been developed in this chapter, and will be applied in this research.

Firstly, the category theory is applied to ensure the stability of measurement procedures. Scott 2004 [9] provides a stability corollary of measurement procedures using representational measurement theory. Based on that, in this thesis, category theory has been researched and developed to provide a mathematical framework for representational measurement. In that way, the category concept is applied to represent the stability corollary and useful to provide a guidance for the stability of measurement procedures.

Secondly, The category model based on category theory is generated to represent the knowledge with complex and different structures. By comparing with relational data model, functional model and production rules, the category model has been found to be the most suitable formula for knowledge representation of the knowledge-based system intended to be developed in this thesis, as it has both the flexibility of structures and good mathematical foundations.

And finally, the method of applying category theory – fibration and adjoint functors to the knowledge acquisition of knowledge-based system has been described in detail. During knowledge acquisition procedures, the initial abstract knowledge-base will be refined to produce a high-performance system. The most important thing among these refinement processes is the consistency between data models. The fibration process and adjoint functors are proved to satisfy refinement requirements, and can be used to carry out the knowledge refinement, gradually expanding an abstract and simplified model into a more concrete model and keeping the consistency of the models. An example is then given to illustrate the implementation of knowledge refinement in this research.

References

[1] Benjamin C. Pierce (1991), *Basic Category Theory for Computer Scientists*, MIT press, ISBN 0-262-66071-7

[2] Fred S. Roberts (1979), *Measurement theory with applications to decision making*, and the social sciences, Encyclopaedia of mathematics and its applications, vol.7 Addison Wesley, ISBN 0-201-13506-x

[3] D. J. Hand (1996), *Statistics and the theory of measurement*, J.R.Stat.Soc.A 159(3), 445-492

[4] Robert Klee (1997), *Introduction to the Philosophy of Science: Cutting Nature at Its Seams,* NY: Oxford University Press

[5] J.R. Flynn (1987), *Massive IQ gains in 14 nations: What IQ tests really measure*, Psychological Bulletin, 101, 171-191

[6] U. Neisser (1997), *Rising scores on intelligence tests*, American Scientist, 85, 440-447

[7] J. Michell (1986), *Measurement scales and statistics: a clash of paradigms*, Psychological Bulletin, 100, 398-407

[8] Paul J Scott (2006), *The case of surface texture parameter RSm*, Measurement Science and Technology, 17, 559-564

[9] Paul J Scott (2004), *Pattern analysis and metrology: the extraction of stable features from observable measurements*, Proc.R.Soc.Lond.A (2004) 460, 2845-2864

[10] http://en.wikipedia.org/wiki/Topology#Mathematical_definition

[11] http://en.wikipedia.org/wiki/Open_set

[12] P.J. Cameron (1994), *Combinatorics: topics, techniques, algorithms*, Cambridge University Press.

[13] ISO 13565-1 (1996), *Geometrical Product Specifications (GPS) – Surface texture: Profile method; Surfaces having stratified functional properties – Part 1: Filtering and general measurement conditions*

[14] 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*

[15] N.N Rossiter & D.A Nelson (1994), *The Categorical Product Data Model as Formalism for Object-Relational Databases*, Technical Report 505, Department of Computing Science, University of Newcastle

[16] Shipman (1981), *The Functional Data Model and the Data language DAPLEX,* Computer Corporation of America, ACM Transactions on Database Systems, 6(1981) 140-173

[17] Dirk Ourston and Raymond J. Mooney (1994), *Theory refinement combining analytical and empirical methods*, Artificial Intelligence 66 (1994) 273-309

[18] A. Ginsberg, S.M. Weiss and P. Politakis (1988), *Automatic knowledge based refinement for classification systems*, *Artif. Intell.* 35 (1988) 197-226

[19] Robert M. Colomb (2001), *Category-Theoretic Fibration as an Abstraction Mechanism in Information Systems*, Acta Informatica 38, 1–44 (2001) c Springer-Verlag

[20] Robert M. Colomb & C.N.G Dampney (2005), *An approach to ontology for institutional facts in the semantic web*, Information and Software Technology 47(2005) 775-783

[21] Michael Johnson, David Naumann and John Power (2001), *Category Theoretic Models of Data Refinement*, Electronic Notes in Theoretical Computer Science 45

[22] Michael Barr and Charles Wells (1990), *Category Theory for Computing Science*, Prentice Hall, ISBN 0-13-120486-6

[23] J. Lambek and P. J. Scott (1986), *Introduction to higher order categorical logic*, Cambridge University Press, ISBN 0 521 35653 9

Chapter 4

GPS – based Surface texture KBS

4.1 Introduction

A geometrical product is a manufactured component having shape, dimensional, form and surface properties. Tolerancing and metrology are primary tools for specifying, assessing, and controlling geometric variability in design and manufacture of a product [1].

Before 1996 there were three individual Technical Committees: TC3 (Limits and Fits), TC10/SC5 (Geometric Dimensioning & Tolerancing) and TC57 (Metrology and Properties of Surfaces) within ISO (International Organisation for Standardisation) for standardizing tolerances and related metrological practices of geometrical products [2]. These three TCs identified gaps and contradictions in the chain of standards that were issued by these three committees for dealing with dimensional and geometrical tolerance specifications and their verifications using metrological instruments, systems and procedures [3].

In recent years, the introduction of new technology – notably Coordinate Measuring Machines (CMMs) and CAD systems – exposed gaps, ambiguities, and inconsistencies in current practice. These findings triggered a wave of effort to formalize tolerancing and metrology, by 'mathematizing' the standards. To harmonise the standardisation of design, manufacture and metrology of geometrical products, a new era based on mathematically defined techniques and standards began [1]. The technical committee ISO/TC 213 was created on 16 June 1996 [2], from the previously mentioned three TCs, to focus on the development of an integrated standard system called Geometrical Product Specifications and Verification (GPS).

Surface texture knowledge is a key part of the GPS system and is important across a very wide spectrum of technical activities, from the design function to specification on a drawing, from the manufacturing process to verification. It has been recognized as being significant in many fields. In particular, Surface texture is an important factor in production processes. It is used to monitor the production process, prevent any failure of the products and ensure surface quality. It also can be used to infer the functional performance of the surface. Therefore, it is important for engineers to know and understand surface texture knowledge [4].

Surface texture is the local deviations of a surface from its ideal shape. The measure of the surface texture is generally determined in terms of its roughness, waviness and form. Surface texture and its measurement are becoming the most critical factors and important functionality indicators in the performance of high precision and nanoscale devices and components. Surface metrology as a discipline is currently undergoing a huge paradigm shift: from profile to areal characterization, from stochastic to structured surfaces, and from simple geometries to complex free-form geometries, all spanning the millimetre to sub-nanometre scales [5-6]. The knowledge-based system being developed in this thesis is basically capturing the profile surface texture knowledge in order to establish and prove the basic concepts.

4.2 The GPS (Geometrical Product Specification) system

The GPS system impacts all products in terms of the properties such as size, dimension, geometrical tolerancing and surface texture, from macro- to nano-scale components, from top-down manufacturing to bottom-up processes. Figure 4.1 is a framework for the GPS system, which links Function, Specification of micro- and nano-geometry through Manufacture, and Verification.

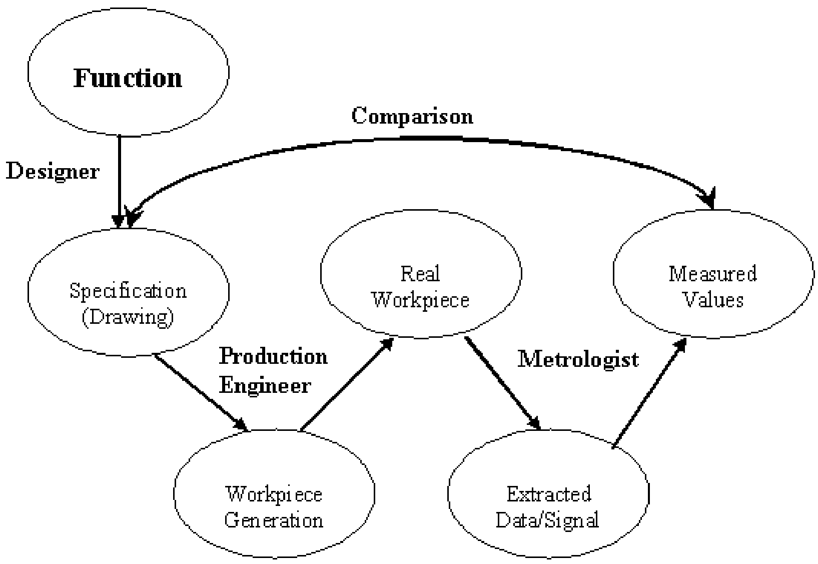


Figure 4.1 GPS Framework

GPS is the means of communication in which designers, production engineers and metrologists exchange unambiguous information about GPS product requirements.

Because of this, GPS documentation may be regarded as the basis of a binding legal contract. In such a market, GPS is the only stable means of communication. Consequently, incorrect and ambiguous definitions of GPS-requirements constitute high economical risks to industry and are liable to be subject to disputes between companies. Thus the understanding and implementation of the GPS system is very important to industry [4]. A complete industrial procedure relating to a workpiece feature includes the following steps:

- Firstly, the requirement for the product performance will be stated, for example, contact surfaces with sliding motions between them.
- The designers will then choose a suitable function for the feature which satisfies the requirements, and defines the product feature that fits the functions, for example fluid friction, dry friction and so on.
- The designers will define the specifications on the technical drawing, which indicates the detailed design intent, including the size, the dimension, the geometrical tolerancing and the surface texture of the product.
- The manufacturers can then choose the corresponding machine process and produce the product in accordance with the specification.
- The final procedure is verification. The metrologists will use a suitable measuring procedure to check whether the real surface of the product conforms to the specifications that have been specified.

Overall, the GPS standards are related to the complete process of 'three worlds': designing (setting up unambiguous specifications), manufacturing (interpreting specifications) and Metrology (verification), see figure 4.2.

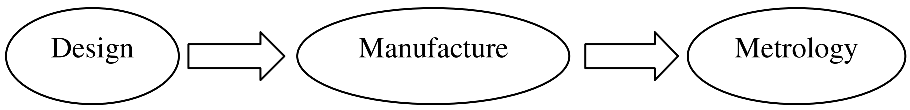


Figure 4.2 Three worlds

The world of Design: it is the responsibility of designers to translate the design intent into a requirement or requirements for specific GPS characteristics and select appropriate specifications to match the functional performance of certain features. The design process comprises the following steps [7]:

a) feature/feature function — the desired design intent of the GPS specification;

b) GPS specification — consisting of a number of GPS specification elements;

c) GPS specification elements — each of which controls one or more specification operations;

d) specification operations — organized in ordered sets to form a specification operator;

e) specification operator — correlates to a certain extent to the intended feature/feature function and defines the measurand of the specification.

The specification of geometrical products is a set of requirements concerning the geometry of a product. It covers three parts of the requirements: Dimensional tolerances, Geometrical tolerance and Tolerances on surface texture, see figure 4.3. GPS give an assurance for obtaining some essential properties of the product: functionality, safety, dependability and interchangeability [4].

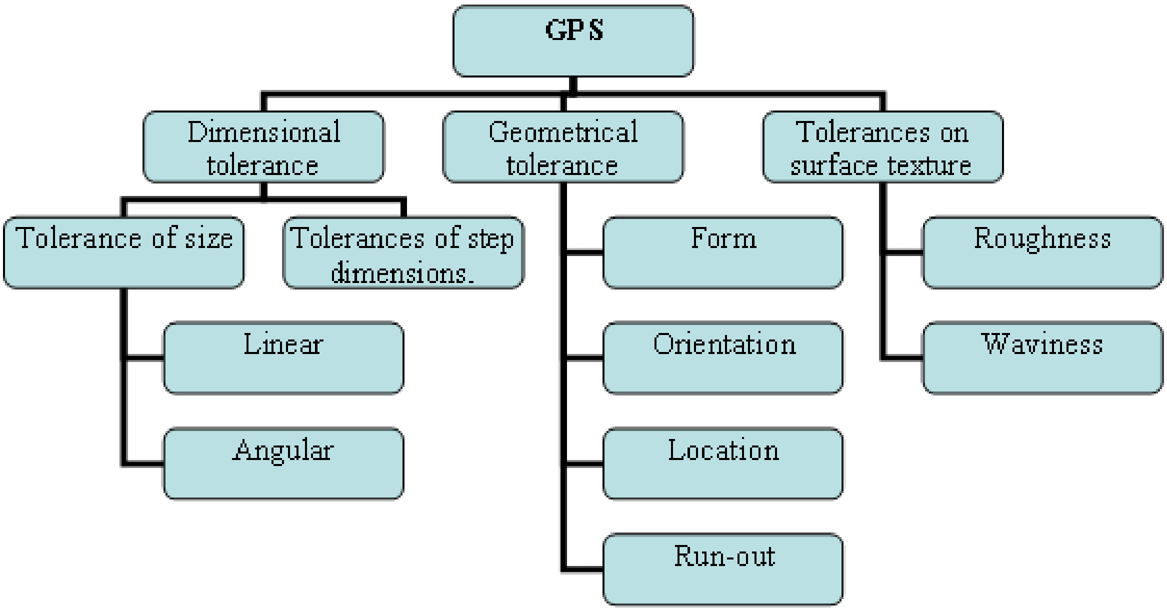


Figure 4.3 General concepts of Geometrical Product Specifications [4]

<u>Functionality</u>: the parameters for a product need to meet certain tolerances in order to satisfy the functional performances. Take a cylinder liner for example, if its surfaces meet the tolerance limits, the piston inside works well.

<u>Safety</u>: the geometrical specifications, such as roughness, shall secure the safety performance of a product. There may be legal requirements that a product must satisfy that have to be taken into consideration. For example, the proper specifications concerning roughness for a machine tool would avoid fatigue cracking, fretting wear, etc. destroying the machine.

Dependability: the proper tolerances would guarantee the long working life of the machine. Take the total hip replacement system for example, the better the surface characteristics of the components, the longer the life expectancy it guarantees.

Interchangeability: the interchangeability would make the assembly and repair of a machine much easier. The indicated tolerances would secure the consistency of the produced products.

The world of Manufacture: it is the responsibility of manufacturers to select the appropriate manufacturing processes to match the specifications of certain features and produce the real product within the tolerances. For example, a particular manufacturing process is capable of producing a limited range of surface roughness values [8]. For instance, the polishing procedure, which is frequently used to produce roughness values of 0.1~0.4μm.

The world of Metrology: it is the responsibility of metrologists to select the appropriate measurement process and determine whether the real surface of a product conforms to the GPS specification. The purpose is the verification of the feature/feature characteristic to the specification operator defined by the GPS specification.

4.2.1 Matrix model with chains

The Joint Harmonisation Group ISO/TC 3-10-57/JHG (working in years 1993 -1996) prepared a Technical Report ISO/TR 14638:1995 [9] about a classification system of GPS standards, known as the Masterplan. In this document all GPS standards have been divided into 4 groups [4], see figure 4.4:

- Fundamental GPS standards: this group consists of such standards that establish very fundamental rules for other standards and the general strucure. Currently there are two documents in this group: ISO 14659 [10] — Fundamental rules, principles and concepts and ISO 14638 [9] — Geometrical product specification (GPS) – Masterplan, which contains the outline of the Masterplan.
- Global GPS standards: this group consists of standards that are related to general GPS standards and complementary GPS standards. Global standards provide the mathematical foundation and formulas which influence general GPS chains of standards directly or as default documents.

- General GPS standards: this group is the main component of the Masterplan and is ordered in a matrix. See figure 4.5.
- Complementary GPS standards: this group consists of standards for specific categories of features or elements, including technical rules for drawing indications, definitions and verification principles.

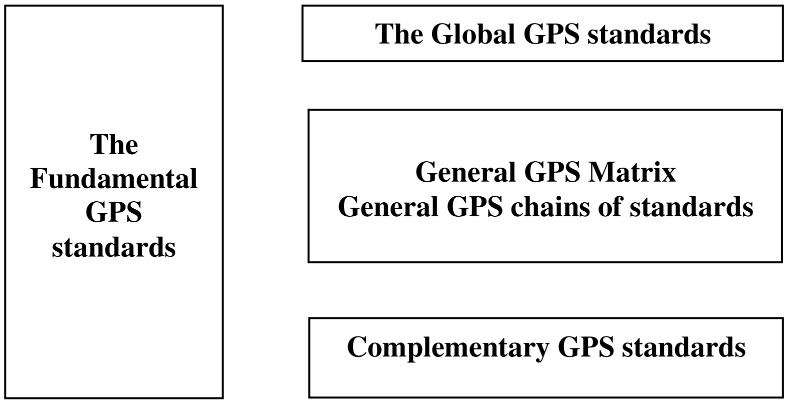


Figure 4.4 Overview of GPS Masterplan structure

Figure 4.5 illustrates the whole general GPS standards system, which is ordered in a matrix. Rows constitute chains of standards and columns concern various characteristics of geometrical features [4].

A chain of standards consists of a set of standards related to given characteristics. The GPS standards cover 6 different geometrical characteristics of features as follows [4]:

1. Product documentation indication – codification: the standards in this link define callout symbols on technical drawings, establish rules of their application and explain how to understand specifications.
2. Definition of tolerances: the main task of these standards is to define the tolerances and their numerical values as well (as translated from callout symbols).
3. Definitions of characteristics of actual (real) features: the standards contain unambiguous definition of the geometry of a non-ideal, real workpiece. The definitions are based on a set of data points of considered features.
4. Assessment of the workpiece deviations – comparison with specified limits: the standards placed in this link state how to prove conformance or non-conformance of a real workpiece with specifications, taking into account the uncertainty of inspection procedures.
5. Measurement equipment requirements: these standards describe characteristics of measurement instruments or specific types of equipment.

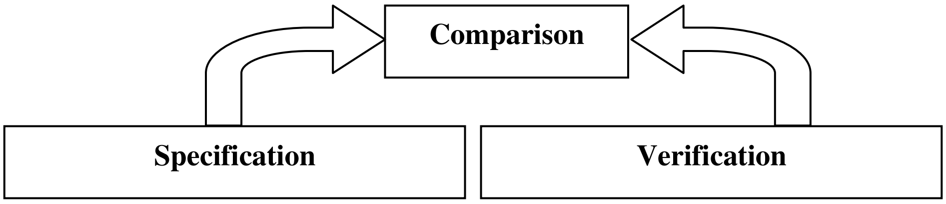
6. Calibration requirements – measurement standards: these standards establish the characteristics of calibration standards used in calibration procedures of the equipment described in link no.5.

Chain link number		1	2	3	4	5	6
Geometrical characteristic of feature		Product documentation indication – codification	Definition of tolerances	Definitions of characteristics of actual (real) feature	Assessment of the workpiece deviations	Measurement equipment requirements	Calibration requirements
1	Size						
2	Distance						
3	Radius						
4	Angle						
5	Form of line independent of datum						
6	Form of line dependent on datum						
7	Form of surface independent of datum						
8	Form of surface dependent on datum						
9	Orientation						
10	Location						
11	Circular run-out						
12	Total run-out						
13	Datums						
14	Roughness profile						
15	Waviness profile						
16	Primary profile						
17	Surface imperfections						
18	Edges						

Figure 4.5 general GPS matrix – layout

For example, standard ISO 1302 – Geometrical Product Specifications (GPS) — Indication of surface texture in technical product documentation [11] influences Product documentation indication (chain link 1) of the chains 14, 15 and 16 of standards on roughness profile, waviness profile and primary profile in the general GPS matrix, as highlighted in grey in figure 4.5.

All the chains of standards are involved in the complete process of designing (setting up unambiguous specifications), manufacturing (interpreting specifications) and verification (measuring). Figure 4.6 gives an example of a roughness profile to illustrate the relations between the six chains of standards. Chains 1-3 are used in the Specification processes, which make it possible to translate the functional requirements into the corresponding specification callouts, and interpret each callout symbols into the manufacturing requirements. Chains 4-6 are used in the Verification processes, which translate the specification into the verification requirements. And finally the value given from the specification processes and the value obtained from the verification processes are compared under the standards in chain 4.



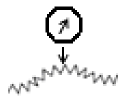
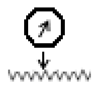
Chain link number	1	2	3	4	5	6
Geometrical characteristic of feature	Product documentation indication – codification	Definition of tolerances	Definitions of characteristics of actual (real) feature	Assessment of the workpiece deviations	Measurement equipment requirements	Calibration requirements
Roughness	 Ra 0.5	 Ra 0.5				

Figure 4.6 The relationships between the chain of standards

4.2.2 Tolerance = Feature + Characteristics + Condition

The tolerance of a workpiece consists of three components: feature, characteristics and condition. In the new generation of GPS technical standard, based on metrology models, several surface models are generated in order to express the features and characteristics more clearly and link the processes of design, manufacture and verification together.

Nominal model

The designer first defines a workpiece of perfect form with shape and dimensions that fit the functions requirements. This workpiece of perfect form is called the nominal model (see figure 4.7(a)). This first step establishes a representation of the workpiece with only nominal values that is impossible to produce or inspect (each manufacturing or measuring process has its own variability or uncertainty).

Skin model

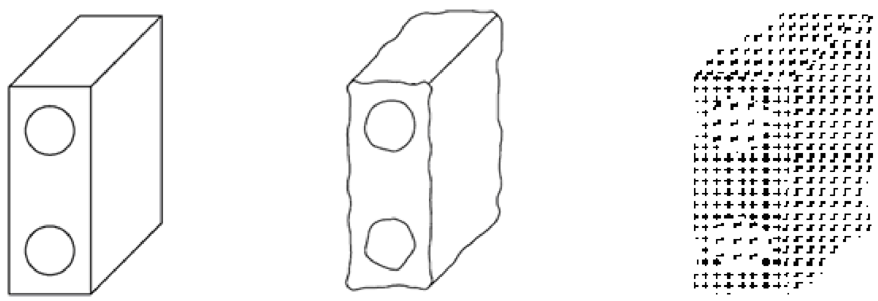
From the nominal geometry, the designer imagines a model with a real surface, which represents the variations that could be expected on the real surface of the workpiece. This model representing the imperfect geometry of the workpiece is called the non-ideal surface model (skin model) (see figure 4.7(b)).

The non-ideal surface model (skin model) is used to simulate variations of the surface at a conceptual level. On this model the designer will be able to optimize the maximum permissible limit values and still match the functional requirements. These maximum permissible limit values define the tolerances of each characteristic of the workpiece [7].

Verification model (Real surface)

The real surface of the workpiece, which is the physical interface of the workpiece with its environment, has imperfect geometry; this real surface of the model is compared with the skin model by the metrologists.

Since it is impossible to completely capture the dimensional variation of the real surface of the workpiece, the verification model as generated by instruments, consists of a finite point set which covers all the real surface and represents the features of the real surface (see figure 4.7(c)).



(a) Nominal Model (b) Non-ideal surface (skin model) (c) Real surface (verification model)

Figure 4.7 Surface models

4.2.2.1Features

The nature of feature is point, line or surface. It includes the integral feature and the derived feature. The integral feature is the surface or line on a surface. The derived feature comprises the centrepoint, median line, median surface or offset feature from one or more integral features.

Geometrical features exist in three worlds as stated in section 4.1: the world of design (nominal model & skin model), the world of manufacture (real workpiece) and the world of metrology (extracted and associated features). They are generated from the different surface models as figure 4.8 shows.

Figure 4.8 gives examples of geometrical features from Design, Manufacture, and Verification stages: Nominal feature is the ideal feature as defined by technical drawing; Non-ideal feature is the simulated imperfect geometry of the non-ideal surface model (skin model); Real feature is the non-ideal feature of a real surface that separate the entire workpiece from the surrounding medium; Extracted feature is a finite number of points extracted from the real feature and represents the real feature; and Associated feature is the feature of perfect form obtained by association of extracted feature in accordance with specified algorithms.

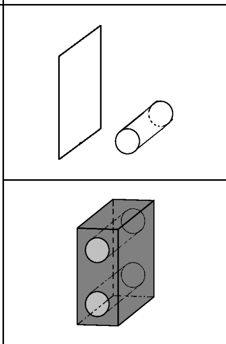
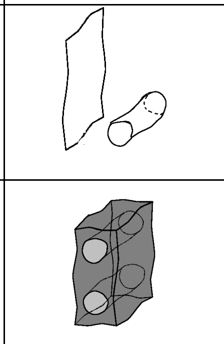
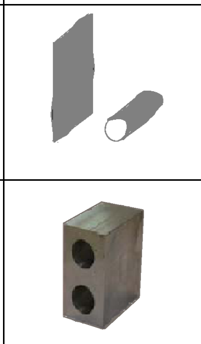
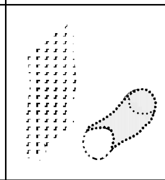
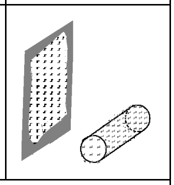
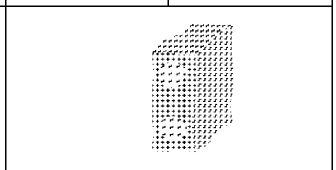
Features	**Nominal feature**	**Non-ideal feature**	**Real feature**	**Extracted feature**	**Associated feature**
Example					
					
	Nominal model	Skin model	Real workpiece	Verification model	
Process	Design		Manufacture	Verification	

Figure 4.8 Examples of geometrical features

4.2.2.2 Characteristics

A characteristic is the single property of one or more features expressed in linear or angular units. The features are described by characteristics, including different mathematical parameters and their numerical values as well, based on a set of data points from the features under consideration. These definitions are all defined in the GPS standards.

There are two kinds of characteristics: intrinsic characteristics and situation characteristics [7]. Intrinsic characteristics are characteristics defined on ideal features. For example, the diameter is the intrinsic characteristic for a circle. While situation characteristics are characteristics defined between ideal features or between ideal and non-ideal features, separated into location characteristics and orientation characteristics.

4.2.2.3 Condition

To ensure the functional performance of a surface, conditions are given that define acceptable limits for the measured value of a characteristic.

Take a cylinder liner in a 10 litre truck engine for example, it requires a good bearing surface and also retain a reservoir of oil for lubrication. According to the factorial designed experiment (FDE), the surface texture parameter Rz, which represents the maximum height of the profile, is an important factor that affects the oil consumption and wear of the liner [12]. To be satisfied with the functional requirement, the texture parameter Rz for the cylinder liner is suggested to be less than 4 µm.

4.2.2.4 Example of a tolerance

The three components of a tolerance namely: the feature, characteristics and conditions, are used to determine the functional properties of a surface.

Figure 4.9 gives an example of indication of tolerance on a technical drawing: the surface P, R-profile, has arithmetic mean deviation 0.5 µm, which indicates the surface texture information of surface P.

The feature in this tolerance is the integral feature extracted from the real surface P; The characteristic is the arithmetic mean deviation of the roughness profile for surface P; and the condition is that 16% of the measured values for the deviation should be less than 0.5 µm. (16% rule is the default rule for surface texture.)

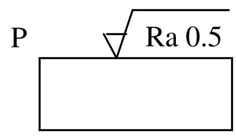


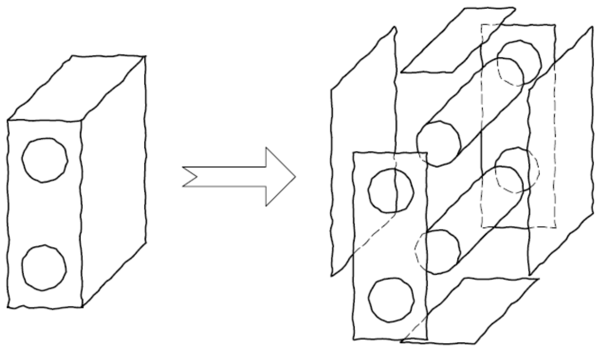
Figure 4.9 Example of tolerance indication on technical drawing

4.2.3 Duality principle

The new generation GPS system, is trying to provide a uniform foundation for designers, manufacturers and metrologists. One important principle that reflects this is the “Duality principle”.

The Duality principle [7] states that the specification of the skin model determines the verification of real model; hence the verification is a mirror of the specification. Each of them consists of the following operations, in order to obtain the features:

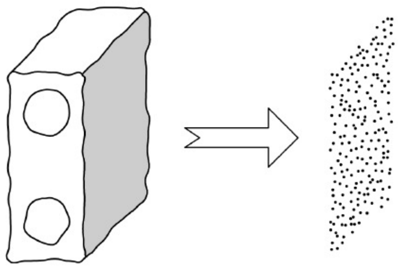
Partition: A feature operation called partition is used to identify the bounded surface which is to be characterised, see figure 4.10. The whole skin model is divided into 8 separate surfaces: 6 side faces and two cylinder holes. It is possible to do the following operations on a particular surface.



a) Skin model b) Non-ideal features obtained by partition of skin model

Figure 4.10 Partition of a skin model [7]

Extraction: A feature operation called extraction is used to identify a finite number of points from the surface (i.e. sampling the surface), see figure 4.11. Figure b) describes a series of points extracted from the side face of the skin model, which represent the features of the surface.



a) Skin model b) Extracted points from a feature of a skin model

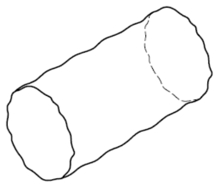
Figure 4.11 Extracted points from a feature of a skin model [7]

Filtration: A feature operation called filtration is used to separate out features of interest at different scales, see figure 4.12, it separates the surface profile into roughness profile and waviness profile.

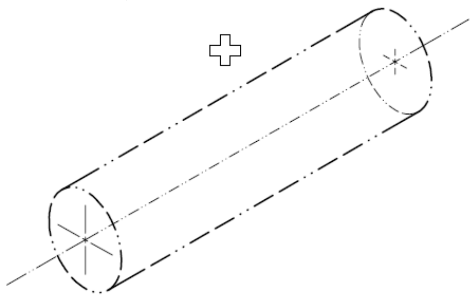
a) Unfiltered profile b) Waviness profile c) Roughness profile

Figure 4.12 Separation of a profile [7]

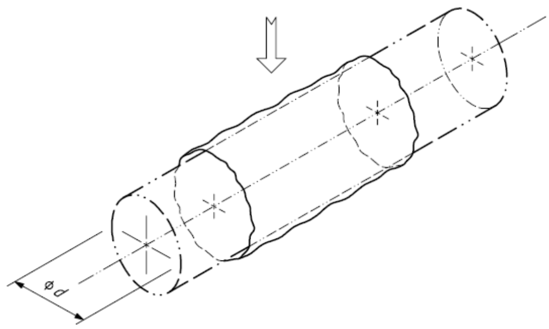
Association: A feature operation called association is used to fit the nominal surface to the real surface and remove the nominal form (e.g. using least square), see figure 4.13. Figure c) associates the ideal cylinder with the non-ideal cylinder, and then uses least square to remove the nominal features.



a) Non-ideal feature



b) Ideal cylinder



c) Association of ideal cylinder with the non-ideal feature

Figure 4.13 Association example [7]

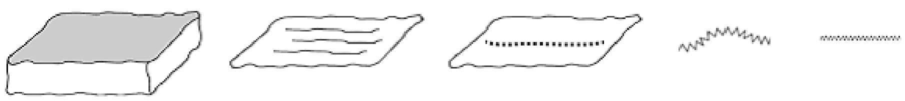
Evaluation: A feature operation called evaluation is used to identify either the value of a characteristic or its nominal value and its limits. The evaluation is always used after the feature operations defining one specification.

4.2.3.1 Specification operations

The specification operations are used to translate the functional requirements, i.e., design intent into the detailed geometrical specifications, which includes the following procedures [13]: design the nominal model which satisfy the functional performances; imagine the skin model from the nominal geometry, which represents the variations that could be expected on the real surface of the workpiece; then partition of the key features from the skin model; extraction of the integral features; and filtration of the features extracted; finally association of the filtered features and evaluation of the associated features.

For example, in accordance with ISO standards, the specification of *Ra* 1,5 indicates the following procedures, see figure 4.14:

- Figure (a), partition of a non-ideal surface from the skin model;
- Figure (b), partition of non-ideal lines from this non-ideal surface in multiple places, (several lines are partitioned from the surface on different places and would be used to do the following operations, the direction of the lines is normally perpendicular to the direction of surface texture lay);
- Figure (c), extraction using the default evaluation length defined in ISO 4288 [14] (4mm for the parameter *Ra* 1,5), to identify a finite number of points from the partitioned lines;
- Figure (d), filtration is carried out using a Gaussian filter with a cut-off wavelength defined in ISO 4288, (the default cut-off wavelength of *Ra* 1,5 is 0.8mm, which means allowing wavelengths below 0.8mm to be assessed with wavelengths above 0.8mm being reduced in amplitude);
- Association is omitted as the nominal form is assumed to have been removed;
- Figure (e), evaluation of the surface on the value of Ra as defined in ISO 4287 [15] and ISO 4288 [14], (the default comparison rule of *Ra* 1,5 is 16% rule, which is used to compare the measured value with the value indicated on the specification. The surface is considered acceptable if not more than 16% of all the measured values, based on the evaluation length, exceed the value specified on the specification, i.e. 1.5 μm).



(a) skin model (b) partition (c) extraction (d) filtration (e) evaluation

Figure 4.14 Example of specification operations

4.2.3.2 Verification operations

The verification operations are used to generate the tolerances from the real workpiece and compare with the specifications, which includes the following procedures [13]: partition of the key features from the real workpiece; extraction of the integral features using the instruments; filtration of the features extracted to obtain the desired geometrical tolerances; association of the filtered features; and finally valuation of the associated features.

In accordance with ISO standards, the verification of the specification *Ra* 1,5 includes the following procedures, see figure 4.15:

- Figure (a), partition of the required surface from the actual workpiece;
- Figure (b), partition of non-ideal lines by the physical positioning of the measuring instrument in multiple places, (the lines are partitioned from different places on the real surface by moving the position of the measuring instrument, such as moving the stylus tip, the direction of the lines should be perpendicular to the direction of surface texture lay);
- Figure (c), extraction of data from the surface with an instrument in accordance with the requirements of ISO 3274 [16], and using the evaluation length defined in ISO 4288, the feature information would be identified by a set of points within the evaluation length;
- Figure (d), filtration of data using a Gaussian filter with a cut-off wavelength defined in ISO 4288;
- Association is omitted as the nominal form is assumed to have been removed;
- Figure (e), evaluation of Ra value as defined in ISO 4287 and ISO 4288 (16 % rule). The measured value is compared with the value indicated on the specification by 16% rule.

(a) Real workpiece (b) partition (c) extraction (d) filtration (e) evaluation

Figure 4.15 Example of verification operations

4.2.3.3 Mirror relation between Specification and Verification

The operations outlined above determine the features to be characterised. The specification also consists of the measurand (specified value for the indicated parameter) from the surface and the verification consists of the measured value from the real surfaces, both of them are obtained after the evaluation process. They are compared with each other using a comparison rule to determine if the real surface is within tolerance. Figure 4.16 illustrates the “mirror” relation between Specification and Verification.

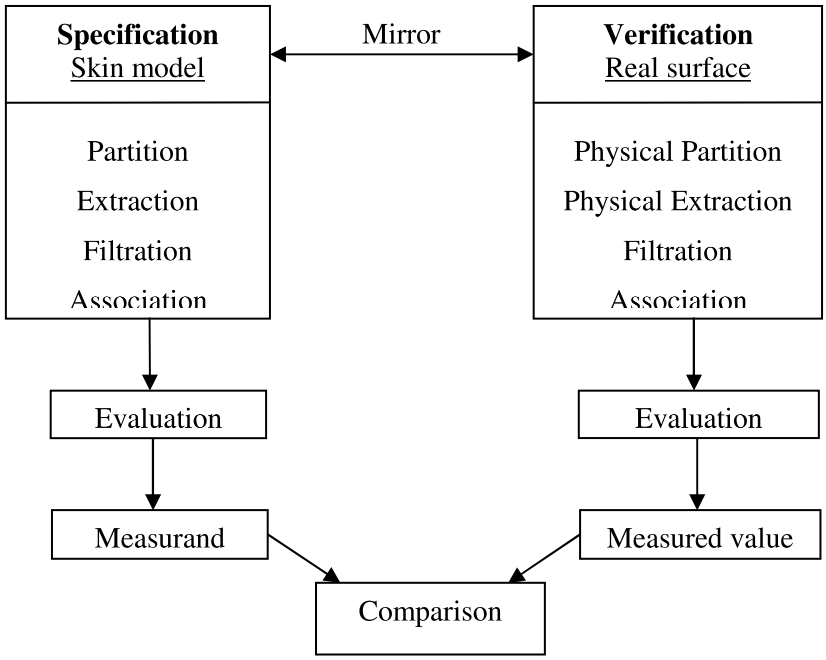


Figure 4.16 "mirror" relation between Specification and Verification

The Duality principle of the specification and the verification links three worlds together: design, manufacture and metrology, and provides a uniform platform for GPS to harmonize the surface models, tolerances and operations. At the design stage, the specification skin model is used to simulate the real surface, through the operations: partition, extraction, filtration, association and evaluation. Designers will be able to optimize the maximum permissible limit values and still match the functional requirements. At the metrology stage, the real surface is considered parallel with the skin model. After the same operations: partition, extraction, filtration, association and evaluation, the limit value derived from the real surface will be compared with the specification value.

4.2.4 General GPS model

According to the refinement processes discussed in Chapter 3, the GPS model is generated and refined. Figure 4.17 shows the refinement processes of general GPS model, which links Function, Specification of micro- and nano-geometry through Manufacture, and Verification.

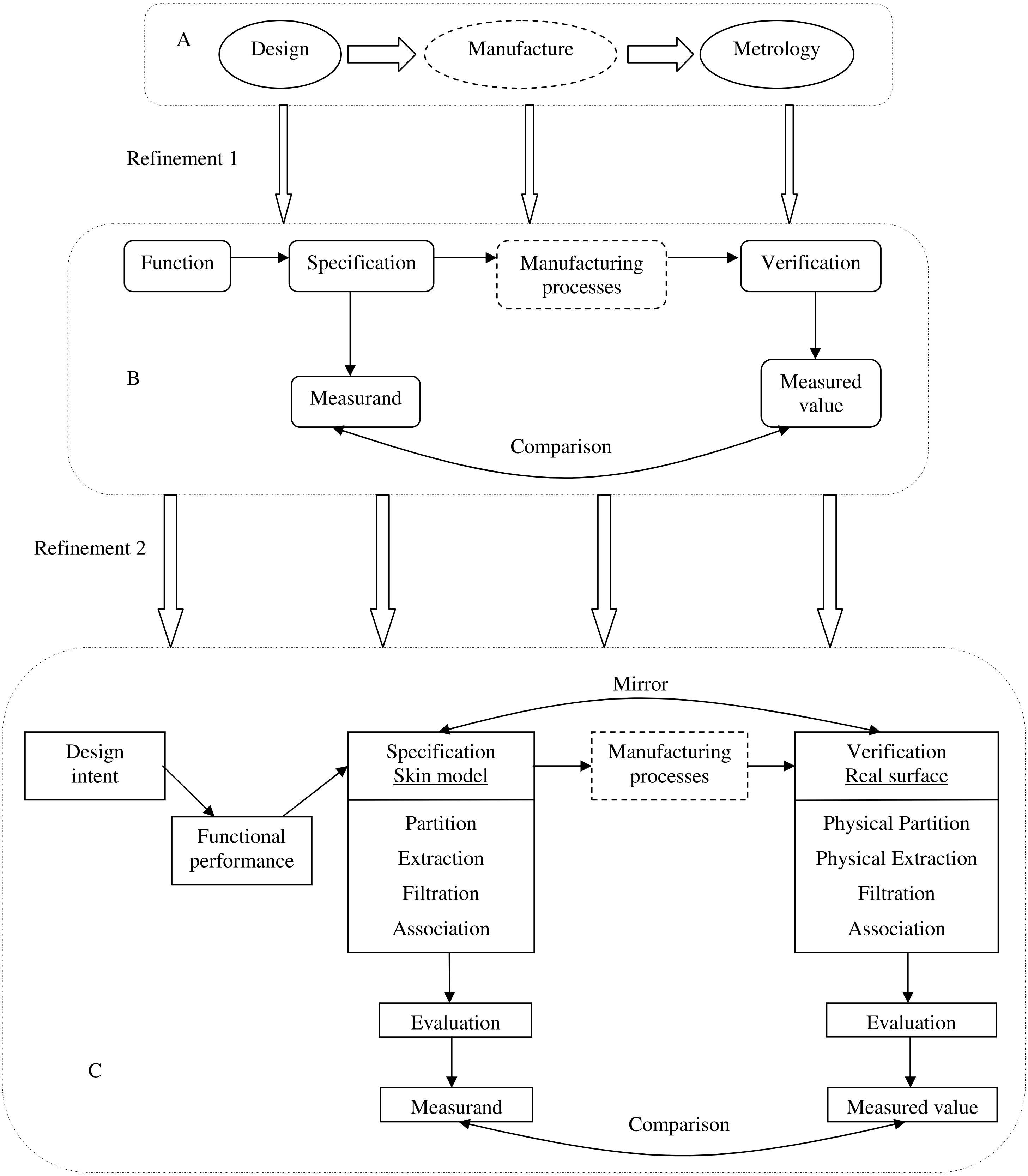


Figure 4.17 Refinement processes of general GPS model

Abstract model A:

Figure A is an abstraction of a concrete GPS system, which includes three objects: Design, Manufacture and Metrology. This model is originally from the three world concepts mentioned in section 4.1. The GPS system is involved in the complete production processes and links the design, the manufacture and the metrology of a product together.

Refinement process 1:

The GPS concepts are involved in three worlds "Design, Manufacture and Metrology". The designer's responsibilities are selecting appropriate specifications to match the functional performance of certain features; Then by means of associate manufacturing processes, the real workpieces are produced; And waiting for the verification, which is a process to compare the measured value obtained from the real workpiece with the measurand defined by the designer, see figure 4.17 refinement 1.

Implementable model B:

Figure B presents the detailed model obtained after refinement process 1. The object "Design" in abstract model A is the generation of "Function", "Specification" and "Measurand" in implementable model B; while the object "Metrology" is the generation of "Verification" and "Measured value", see figure 4.18.

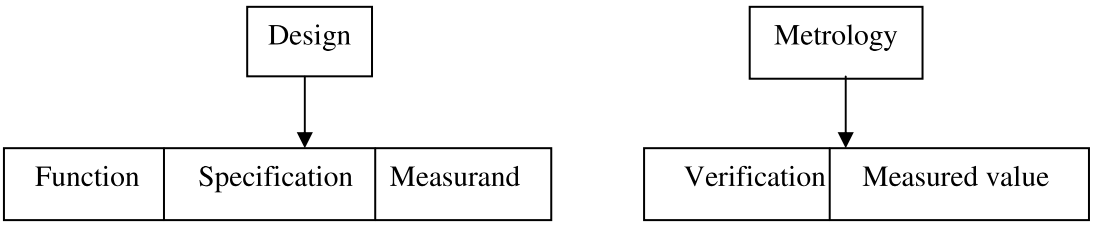


Figure 4.18 Refinement process 1

This model is a reflection of chains of GPS matrix. Section 4.2 described that chains 1-3 are used in the Specification processes, which make it possible to translate the functional requirements into the corresponding specification callouts, and interpret each callout symbol into the manufacturing requirements. Chains 4-6 are used in the Verification processes, which translate the specification into the verification requirements. And finally the value given from the specification processes and the value obtained from the verification processes are compared to verify the conformance.

Refinement process 2:

The function process is the translation of design intent into a requirement or requirements for specific GPS characteristics. As stated in Duality Principle, the verification is a mirror to specification, and both of them comprise the following operations: partition, extraction, filtration and association. The metrologist begins by reading the specification, taking into account the non-ideal surface model (skin model), in order to know the specified characteristics. From the real surface of the workpiece, the metrologist defines the individual steps of the verification plan, depending on the measuring equipment.

Implementable model C:

Figure C present the more detailed model obtained after refinement process 2. The object "Function" in model B is above the "Design intent" and "Functional performance" in model C; the object "Specification" is the generalization of "Partition", "Extraction", "Filtration", "Association" and "Evaluation"; and the object "Verification" is the generalization of "Physical Partion", "Physical Extraction", "Filtration", "Association" and "Evaluation". See figure 4.19.

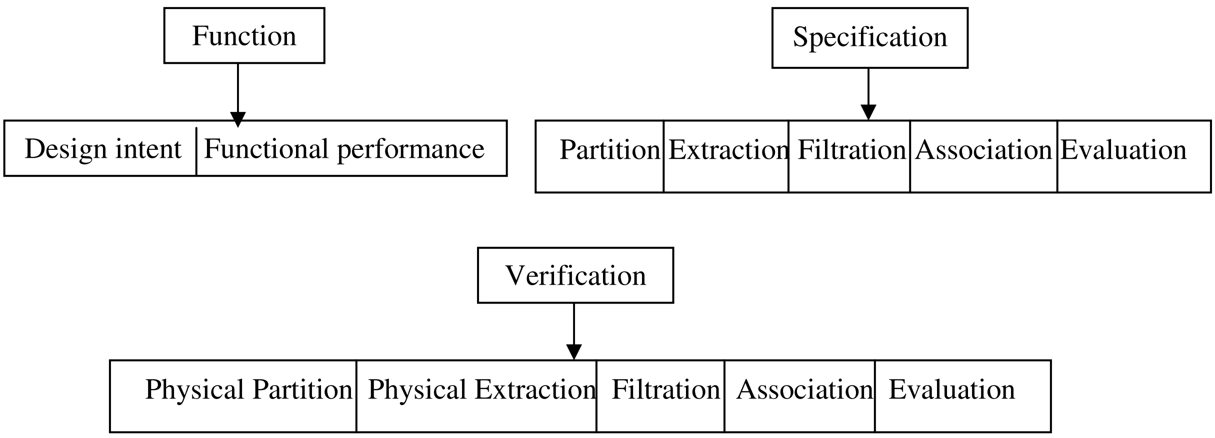


Figure 4.19 Refinement process 2

This model is refined from the tolerances definition and Duality principle of GPS system. The tolerances including features, characteristics and conditions of a workpiece are obtained step-by-step following the operations both for specification skin model and verification real surface. And according to the Duality principle, the specification of skin model determines the verification of real model; hence the verification is a mirror and parallel of the specification. Finally, the measurand derived from the skin model and the measured valued obtained after from the real surface are compared with each other using a comparison rule to determine if the real surface is within tolerance.

The operations in both specification and verification of GPS system are proved to be stable by the stability corollary as stated in Chapter 3. It is easy to show that these two refinements are both adjoint as required in Chapter 3 for consistency.

4.3 Surface texture knowledge-based system

The standards of surface texture knowledge mainly cover all the chain links 1-6 of the chain of standards on roughness profile, waviness profile and primary profile in the general GPS matrix [4], see figure 4.20.

Chain link number / **Geometrical characteristic of feature**		**1**	**2**	**3**	**4**	**5**	**6**
14	Roughness profile	ISO 1302	ISO 4287, 12085, 13565-1, 13565-2, 13565-3	ISO 4288, 12085, 11562, 13565-1	ISO 4288, 12085	ISO 3274	ISO 5436, 12179
15	Waviness profile	ISO 1302	ISO 4287, 11562, 12085	ISO 11562, 12085	ISO 12085	ISO 3274	ISO 5436, 12179
16	Primary profile	ISO 1302	ISO 4287, 11562, 13565-3	ISO 4288	ISO 4288	ISO 3274	ISO 5436, 12179

Figure 4.20 Position of surface texture standards in the GPS matrix model

Following the same refinement process as GPS system, the knowledge of surface texture can be divided into four parts. See figure 4.21, which shows the four parts: function of surface, specification, manufacture and verification. Each is a part with a different structure and undergoing different refinement processes. The following sections will discuss the knowledge representation methods for each part, and the details of each knowledge-base will be described in Chapters 5-8.

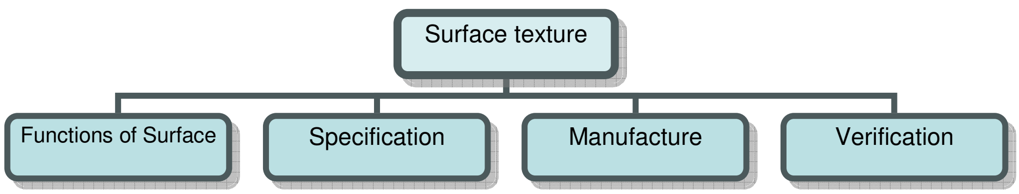


Figure 4.21 Surface texture framework

The main operations of the system are listed below and illustrated in figure 4.22:

1. Suggest functional requirements according to design intent; Step by step to help the users determine the requirements for the surface. For example, users would find the best suitable functional requirements according to the functional performances, wear types, related motions, etc.

2. Help to determine the specification callout to satisfy the functional performance of the surface; Suggest the important parameters to be indicated together with the limit values; Help to determine other symbols in the callout.
3. Retrieve a complete callout after entering a callout from a drawing and give the default values if available, and vice versa. Normally, the callouts indicated on the drawing are not complete, such as "Ra 3.3", where the missing values are given by the default values as defined in ISO standards [11]. The knowledge-based system can help engineers obtain the callout with complete information quickly, without looking up in references. The complete callout should be "U 0.008-2.5 / Ra516% 3.3", which means that the callout is used for an upper specification limit, the bandwidth is 0.008 millimetre to 2.5 millimetre, the parameter name is Ra, the evaluation length is 5 times of the sampling length, the comparison rule is 16%-rule and the limit value is 3.3 micrometer.
4. Suggest appropriate manufacturing processes to produce the surface. The suitable manufacturing procedures would be suggested to satisfy the specified tolerances and other requirements, such as materials, quantities, etc.
5. Suggest suitable measurement procedures to verify the manufactured surface, including selecting the proper measuring instruments, suggesting the correct measurement procedures and parameters, and carrying out the comparison of measured values with the specification.

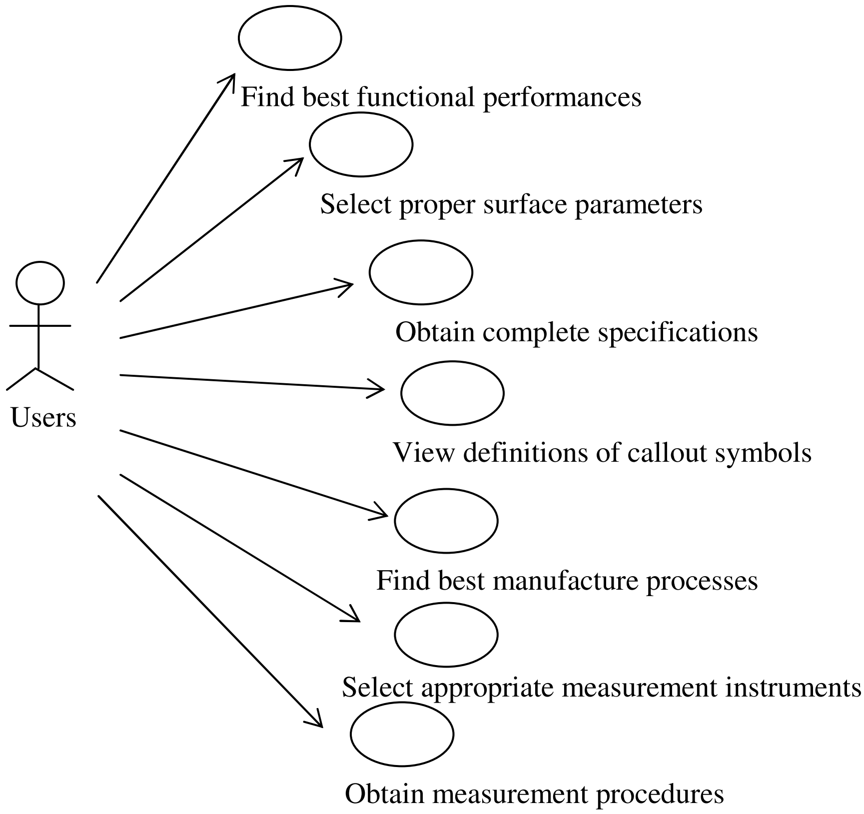


Figure 4.22 Examples of system operations

4.3.1 The architecture of surface texture knowledge-based system

Surface texture knowledge-based system consists of surface texture knowledge-base and an user-friendly interface. Figure 4.23 shows the architecture of the surface texture knowledge-based system. Surface texture knowledge-base comprises four knowledge-bases: Function knowledge-base, Specification knowledge-base, Manufacture knowledge-base and Verification knowledge-base, which are connected by relationships and rules. The knowledge stored in this system is mainly captured from the ISO standards and relevant books written by experts. The following sections will introduce the knowledge stored in the knowledge-base in detail, including the knowledge acquisition of surface texture knowledge-base and different structures of each knowledge-base. The inference engine is used to infer the information according to the rules and add this information into the database. The user-friendly interface is the bridge between the users and the knowledge base, which can transfer the information between the users and the program.

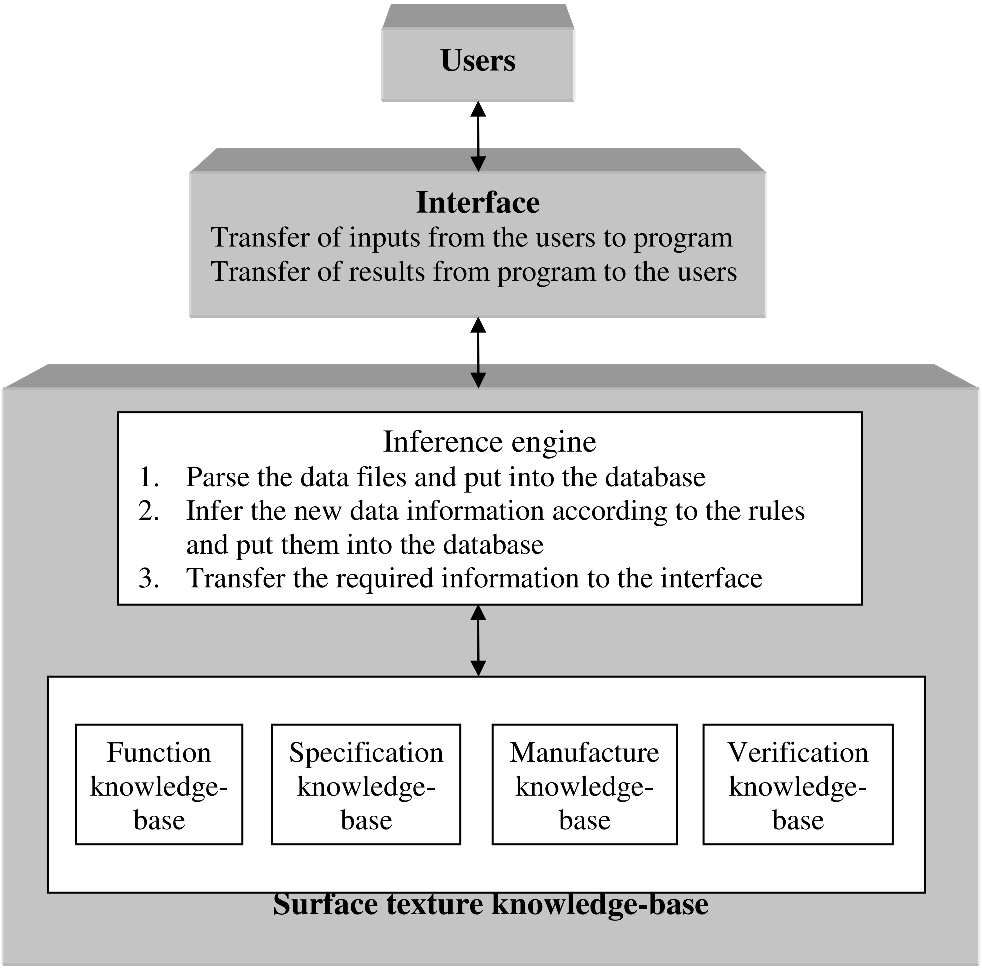


Figure 4.23 Architecture of Surface texture knowledge-based system

4.3.2 Knowledge acquisition

Knowledge acquisition is the first and very important step of the knowledge-based system design. It directly affects the efficiency and availability of the system.

The requirements for knowledge acquisition are:

1. Accuracy: the knowledge being captured should be able to exactly represent the methods from standards or the thoughts of experts';
2. Reliability: the knowledge should be validated by time;
3. Integrity: secure the integrity of the knowledge;
4. Refinement: get rid of the redundancy of the knowledge.

In order to satisfy the requirements above, the procedures to carry out the knowledge acquisition are to first distinguish the data structures of the knowledge, and look for the suitable knowledge representation models, this is the first and most difficult step; and then refine the knowledge-base into a complex implementable model which includes necessary information for software engineers to realise the integrate system.

The framework of surface texture is divided into four parts, see section 5.2: Function, Specification, Manufacture and Verification. Similar to the refinement processes of GPS system, the surface texture knowledge-base is refined to the structure below, see figure 4.24:

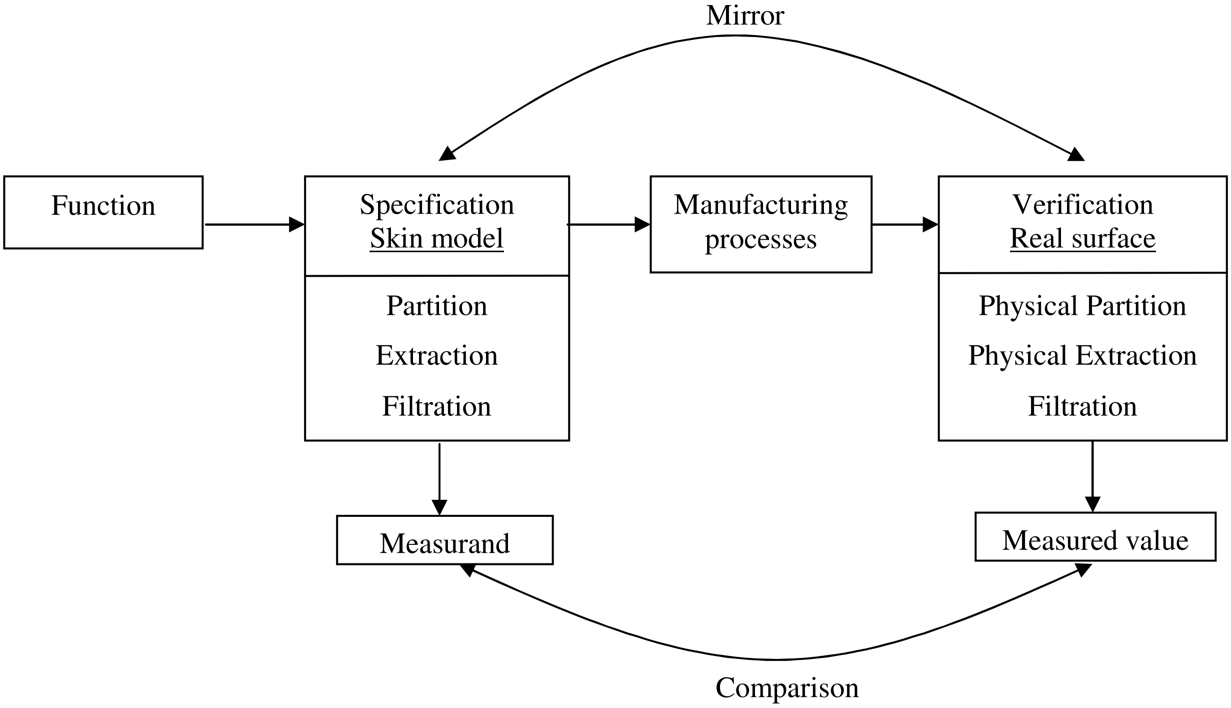


Figure 4.24 Knowledge acquisition for surface texture knowledge-base

Function: the function knowledge-base includes the information relating to the functional performance to specific design intent, and the suggestions of surface parameters selection as well;

For example, the design intent is to manufacture a cylinder liner in a 10 litre truck engine, then the functional performance for the liner is that it requires a good bearing surface and also retains a reservoir of oil for lubrication. And the knowledge-base would suggest that the surface texture parameter Rz, which represents the maximum height of the profile, is an important factor that affects the oil consumption and wear of the liner [12].

Specification: the specification knowledge-base includes the feature information obtained from the operations: Partition, Extraction, and Filtration. The feature should be satisfied with the functional requirements;

For example, to be satisfied with the functional requirement, the limit value of the parameter Rz for the above cylinder liner is suggested to be less than 4 μm. The corresponding information about the complete operation procedures, such as evaluation length for the extraction, and bandwidth for the filtration would be suggested by the knowledge-based system.

Manufacture: the manufacture knowledge-base includes the information linking the suitable manufacturing processes with surface texture; Not only the surface texture requirements, but also the material suitability, the design requirements, the quality issues, and so on will be considered for the correct process selection.

Verification: the verification knowledge-base includes the information of how to obtain the features from the real surface; the procedures in verification mirror the procedures in specification. And the measured values would be compared with the measurand of specification to check whether the real surface conform to the specification.

Take the above cylinder liner surface for example, the knowledge-based system would suggest three instruments that may be capable of doing the measurement: Stylus, Focus and SEM. And also provide the measurement parameters including sampling spacing, traversing length and so on. Finally, the measured values would compare with the specification value 4 μm by the comparison rule.

The detailed knowledge acquisition of each knowledge-base will be described in the following Chapters.

4.3.3 Knowledge representation - different structures of surface texture knowledge

After the knowledge is captured, it is necessary to distinguish the characteristics and data structures of the knowledge and find a suitable data model to represent the knowledge. The knowledge has different characteristics, such as general, specific, uncertain, etc. For example, table 4.1 shows some examples of knowledge with different characteristics.

Characteristics	Knowledge examples
General, explanative, certain	Ra is the parameter that means arithmetical mean deviation of the assessed profile.
Specific, explanative, uncertain	Stylus has bigger measurement ranges.
General, stated, uncertain	Roughness parameters are important for two rolling contact parts.

Table 4.1 Examples of knowledge characteristics

Different data models would be used to represent different knowledge. Chapter 2 discussed a series of data models in common use. This knowledge-based system would adopt suitable models to represent surface texture knowledge. The details are discussed below:

4.3.3.1 Function

Relating surface texture to performance is called function. The surface texture has a direct influence on the quality of the workpiece surface. Therefore, the determination of the surface roughness parameters of a certain functional surface is very important. It would help a manufacturer to produce a suitable workpiece matching the functional requirements.

The knowledge in this knowledge-base includes two parts: the knowledge that already exists in books, standards; and the knowledge need to be added by experts and users.

Since there are few references available for relations between functional performances and surface parameters, there is no easy way to capture all the necessary information for this knowledge-base. The system is trying to incorporate available resources about parameters selection rules and also being developed to be an open platform for experts or designers to add their expertise and specific examples later.

Figure 4.25 gives part of knowledge in function knowledge-base, which is a redrawing from ISO 12085 [17]. This is an example of a classification of surface function together with a relationship table for motif parameters. The left part of the table lists the classification of surface function, which takes a tree structure, see figure 4.26. The right part of the table is a relationship table for motif parameters, including the relations between parameters and different functions, and relations between parameters.

Surface		Functions applied to the surface			Parameters								
					Roughness profile			Waviness profile				Primary profile	
		Designations		Symbol[a]	*R*	*Rx*	*AR*	*W*	*Wx*	*Wte*	*AW*	*Pt*	*Pδc*
two parts in contact	with relative displacement	Slipping (lubricated)		FG	●			≤0,8R			○		●
		Dry friction		FS	●		○		●		○		
		Rolling		FR	●			≤0,8R	●		○		○
		Resistance to hammering		RM	○		○	○			○		●
		Fluid friction		FF	●		○				○		
		Dynamic sealing	with gasket	ED	●	○	○	≤0,6R	●		○		
			without gasket		○	●		≤0,6R					●
	without displacement	Static sealing	with gasket	ES	○	●		≤R		○	○		
			without gasket		○	●		≤R		●			
		Adjustment without displacement with stress		AC	○								●
		Adherence (bonding)		AD	●							○	
Independent surface	with stress	Tools (cutting surface)		OC	○		○	●			●		
		Fatigue strengths		EA	○	●	○						○
	without stress	Corrosion resistance		RC	●	●							
		Paint coating		RE			○				○		
		Electrolytic coating		DE	●	≤2R	●						
		Measures		ME	●			≤R					
		Appearance (aspect)		AS	●		○	○			○		

● Most important parameters: specify at least one of them.
○ Secondary parameters: to be specified if necessary according to the part functions.
The indication ≤ 0,8R, for example, means that, if the symbol FG is indicated on the drawing, and W not otherwise specified, the upper tolerance on W is equal to the upper tolerance on R multiplied by 0,8.

[a] The symbols (FG, etc.) are acronyms of French designations.

Figure 4.25 Example of a classification of surface function together with a relationship table for motif parameters taken from ISO 12085 [17]

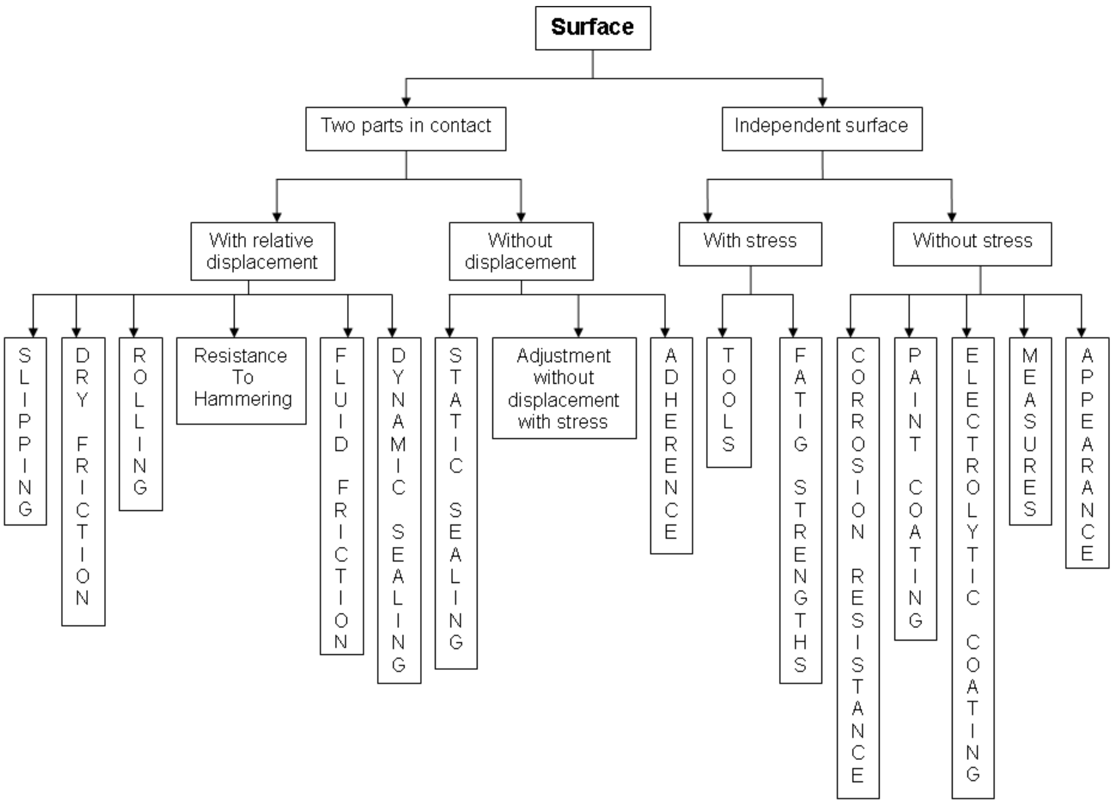
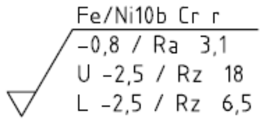


Figure 4.26 Tree structure of Function

A pattern language [18] & [19] is adopted to represent the knowledge in this knowledge-base. The pattern language not only provides an open platform structure, but also takes a tree structure matching the structure of function knowledge, which satisfies the requirements of functional knowledge-base. The details of the function knowledge-base are discussed in Chapter 7.

4.3.3.2 Specification

The main source of information for the specification knowledge base is the callout symbol as contained in ISO 1302 [11]. Surface texture callouts are the symbols used on a drawing to define the surface texture design intent, such as 

. The callout symbol comes from several feature operations: partition, extraction and filtration. The knowledge in specification takes the form of a hierarchical structure, see figure 4.27. Although it looks like a tree structure, it's more complicated. There are even more relations between son nodes, for example, Filtration relates to Extraction.

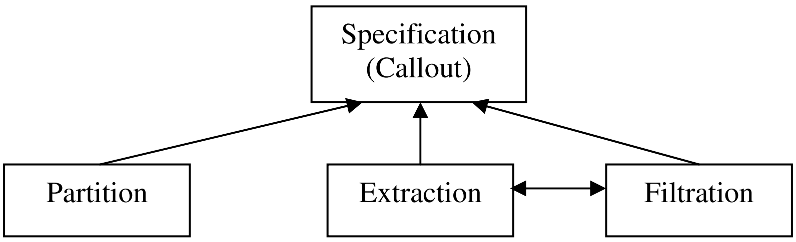


Figure 4.27 Hierarchical structure of Callout symbol

The knowledge in specification knowledge-base mainly comes from the ISO standards [11], this knowledge is general, stated and certain. The category model [20] is adopted to represent the knowledge in this knowledge-base, which has the ability to provide a formulism for hierarchical structure and has object-oriented capabilities. The details of specification knowledge-base are discussed in Chapter 5.

4.3.3.3 Manufacture

This knowledge-base is aimed to help choosing appropriate manufacturing processes according to the specifications of the surface. A set of so-called PRIMAs (Manufacturing Process Information Maps) have been developed to enable correct process selection [21]. PRIMAs give detailed data on the characteristics and capabilities of each process in a standard format under general headings including: material suitability, design considerations, quality issues, general economics and process fundamentals and variations.

Figure 4.28 shows a simple flowchart of manufacturing processes selection.

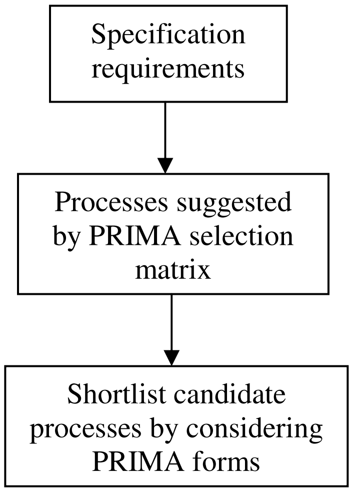


Figure 4.28 Flowchart of simple processes selection

First of all, a PRIMA selection matrix which has two basic variables [21]: material type and production quantity is matched with the requirements; and then the suggested processes would be provided for the manufactures in PRIMA format [21]. The format is very deliberate. First an outline of the process itself - how it works and under what conditions it functions best. Second a summary of what it can do - limitations and opportunities it presents, and finally an overview of quality considerations including process capability charts for relating tolerances to characteristic dimensions; and finally, the candidate processes would be determined by the manufacturers after considering all the criteria listed in PRIMA form [21].

The knowledge in manufacture knowledge-base mainly comes from K.G.Swift & J.D.Booker (2003) [21], this knowledge is general, explanative and uncertain. The knowledge-base will shortlist the candidate processes according to the design requirements, and suggest criteria for manufacturers to make the decision. The category model [20] is adopted to represent the knowledge in this knowledge-base, which not only has the ability to represent this flowchart and has object-oriented capabilities to represent the detailed PRIMA forms. The details of verification knowledge-base are discussed in Chapter 8.

4.3.3.4 Verification

Verification includes: classification of instruments and measurement procedures. According to the duality principle in ISO 17450-1 [7], the verification procedure mirrors the specification procedure, see figure 4.16. It also includes the following feature operations: partition, extraction and filtration, and takes the form of a hierarchical structure, see figure 4.29.

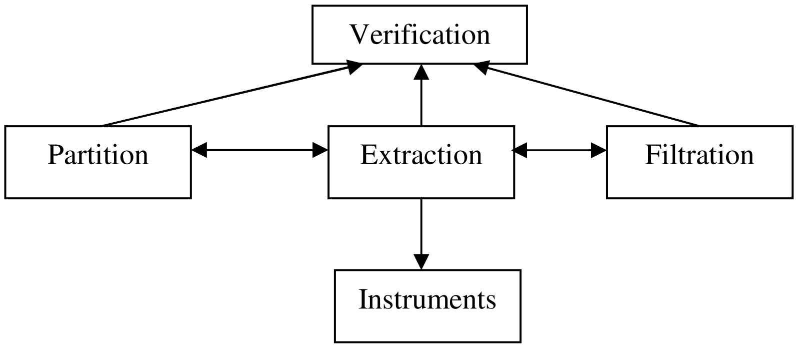


Figure 4.29 Hierarchical structure of Verification knowledge-base

The knowledge in verification knowledge-base mainly comes from ISO standards and references [22-23], this knowledge is specific, explanative and uncertain. Similar to specification knowledge-base, the category model is adopted to represent the knowledge in this knowledge-base, which has the ability to provide a formulism for hierarchical structure and has object-oriented capabilities, i.e. to represent the A-W plot. The details of verification knowledge-base are discussed in Chapter 6.

4.4 Summary

The GPS system and its surface texture concepts have been presented. GPS system impacts all products in three worlds: Design, Manufacture and Metrology. In order to represent the GPS system and link these three worlds together, the matrix model with ISO Masterplan are introduced which includes six chains of standards and covers specification and verification processes. The tolerance is generated from each process of specification and verification, and indicates the characteristics of product features. The duality principle states that verification mirrors to specification, which is the uniform foundation for designers and metrologists. According to these information and rules, the abstract GPS model is able to generate and refine to a concrete model, which could be a basis for the development of surface texture general model.

Based on refined GPS model, surface texture model has been generated, including four knowledge-bases: function, specification, manufacture and verification. The knowledge-bases will be large, distributed and take the different structures. Different structures for each knowledge-base are discussed in detail, and new data models — category model and pattern language have been put forward to satisfy the requirements.

The next four chapters will describe the design procedures for each knowledge-base in detail.

References

[1] Herbert B.Voelcker (1993), A *current perspective on tolerancing and metrology*, CRTD-Vol.27, International Forum on Dimensional Tolerancing and Metrology, ASME 1993

[2] X. Jiang (2002), *Review of the Next Generation GPS*, 2nd International Symposium on Instrumentation, Science and Technology, Jinan, China

[3] P. Bennich (1994), *Chains of Standards - A New Concept in GPS Standards*, Manufacturing review, The American Society of Mechanical Engineers, Vol. 7, No. 1, pp.29-38

[4] Z.Humienny (2001), *Geometrical Product Specifications,* Warsaw University of Technology Printing House, ISBN 83-912190-8-9

[5] X. Jiang, P. Scott, D. Whitehouse, L. Blunt (2007), *Paradigm Shifts in Surface Metrology, Part I: Historical Philosophy*, The Proceedings of Royal Society 463 pp. 2049-2070

[6] X. Jiang, P. Scott, D. Whitehouse, L. Blunt (2007), *Paradigm Shifts in Surface Metrology, Part II: The Current shift*, The Proceedings of Royal Society 463 pp. 2071-2099

[7] ISO/TS 17450-1 (2005), *Geometrical product specifications (GPS) — General concepts — Part 1: Model for geometrical specification and verification*, International Organisation for Standardisation, Geneva

[8] Leigh Mummery (1990), *Surface Texture Analysis – The handbook*, Hommelwerke Gmbh

[9] ISO/TR 14638 (1995), *Geometrical Product Specifications (GPS) ---Masterplan*, International Organization for Standardisation, Geneva

[10] ISO/DIS 14659 (2007), *Geometrical Product Specifications (GPS) -- Fundamentals -- Concepts, principles and rules*, International Organization for Standardisation

[11] ISO 1302 (2002), *Geometrical Product Specifications (GPS) — Indication of surface texture in technical product documentation,* International Organisation for Standardisation, Geneva

[12] Liam Blunt & Xiangqian Jiang (2003), *Advanced Techniques for Assessment Surface Topography*, Kogan Page Science, ISBN 1 9039 9611 2

[13] ISO/TS 17450-2 (2002), *Geometrical Product Specifications (GPS) --General Concepts--Part 2: Basic tenets, specifications, operators and Uncertainties*, International Organization for Standardisation

[14] ISO 4288 (1996), *Geometrical Product Specifications (GPS) -Surface texture: Profile method -Rules and procedures for the assessment of surface texture*, International Organisation for Standardisation, Geneva

[15] ISO 4287 (1997), *Geometrical Product Specifications (GPS) -Surface texture: Profile method –Terms, definitions and surface texture parameters*, International Organisation for Standardisation, Geneva

[16] ISO 3274 (1996), *Geometrical Product Specifications (GPS) - Surface texture: Profile method - Nominal characteristics of contact (stylus) instruments*, International Organisation for Standardisation, Geneva

[17] ISO 12085 (1996), *Geometrical Product Specifications (GPS) — Surface texture: Profile method — Motif parameters*, International Organisation for Standardisation, Geneva

[18] Christopher Alexander (1977), *A Pattern Language: Towns, Buildings, Construction*, New York: Oxford University Press

[19] Wikipedia, the free encyclopedia, *Pattern language*, http://en.wikipedia.org/wiki/Pattern_language

[20] N.N Rossiter; & D.A Nelson (1994), *The Categorical Product Data Model as Formalism for Object-Relational Databases*, Technical Report 505, Department of Computing Science, University of Newcastle, 1995

[21] K.G. Swift & J.D. Booker (2003), *Process selection: from design to manufacture*, Elsevier Butterworth-Heinemann, ISBN 0750654376

[22] Richard Leach (2001), *Measurement Good Practise Guide No.37-The Measurement of Surface Texture using Stylus Instruments*, National Physical Laboratory, ISSN 1368-6550

[23] D J Whitehouse (1997), *Surface metrology*, Measurement Science and Technology **8** 955-972

Chapter 5

Specification Refinement

5.1 Introduction

This chapter aims to provide a detailed explanation of refinement processes for the specification knowledge-base, including knowledge acquisition and knowledge representation.

5.1.1 Specification knowledge-base system

The specification knowledge-base is used to determine the features to be characterised, the characteristic of the features and the comparison rule. It aims to translate the functional requirements, i.e. design intent into the detailed geometrical specifications, which includes the feature information obtained from the operations: Partition, Extraction, and Filtration. The feature should satisfy the functional requirements. For example, to be satisfied with the functional requirements, the limit value of the parameter Rz (amplitude parameter) for a cylinder liner is suggested to be less than 4 μm. The corresponding information about the geometrical specifications, such as evaluation length for the extraction, and bandwidth for the filtration would be suggested by the knowledge-based system.

The main source of information for the specification knowledge-base is the callout symbol as contained in ISO 1302 [1]. Surface texture callouts are the symbols used on a drawing to define the surface texture design intent (see figure 5.1). The callout symbol information comes from several feature operations: partition, extraction and filtration (see detail explanation in section 5.2).

Key:

a. Indication of specification limit.
b. Filter type "X".
c. The transmission band, including the lower limit and the upper limit.
d. Profile (*R* – roughness profile, *W* – waviness profile or *P* – primary profile).
e. Characteristic/parameter.
f. Evaluation length as the number of sampling lengths.
g. Comparison rule ("16 %-rule" or "max-rule").
h. Limit value in micrometres.
i. Machining allowance.
j. Type of manufacturing process.
k. Surface texture lay.
l. Manufacturing methods.

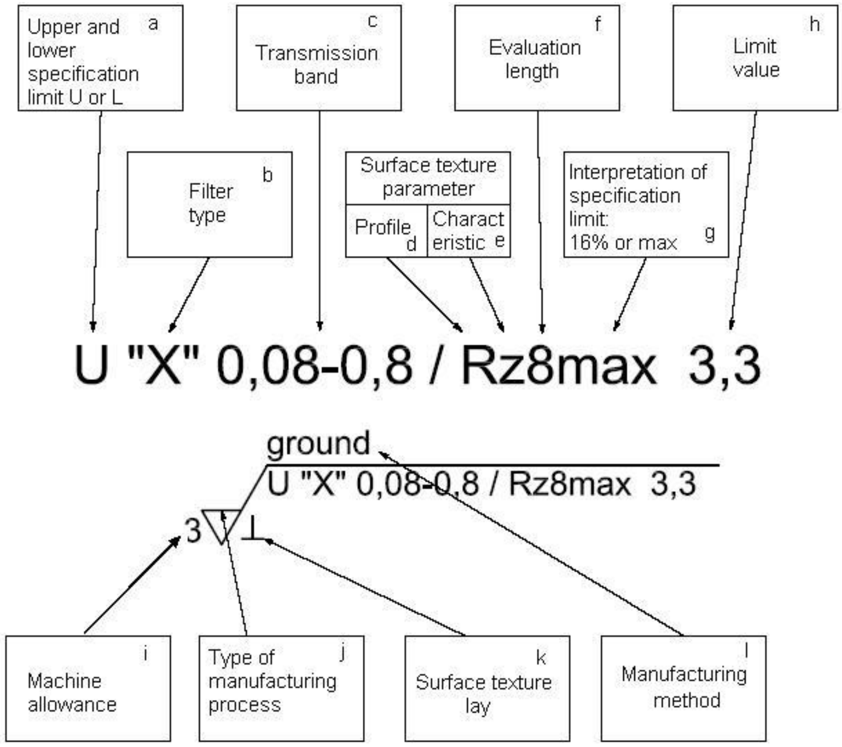


Figure 5.1 Surface texture callout symbol [1]

5.1.2 Introduction of 2D surface texture parameters

5.1.2.1 Parameters from three profiles

The measure of the surface texture is generally determined in terms of its roughness, waviness and form. Roughness is the irregularities which are inherent in the production process; waviness is the part of the texture on which roughness is superimposed; and form is the general shape of the surface, ignoring variations due to roughness and waviness, see figure 5.2 a), b), c) and d) [2]. The capital letters R, W and P are used to represent parameters calculated from these three profiles: R for roughness parameters calculated from roughness profiles, W for waviness parameters calculated from waviness profiles and P for primary profile parameters calculated from raw profiles.

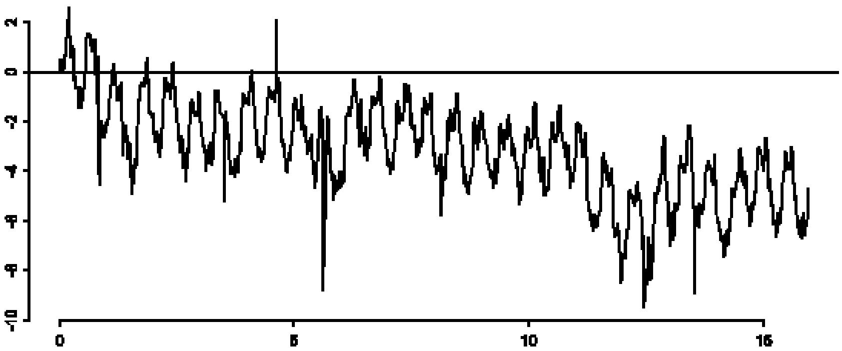


Figure a) Traced real profile

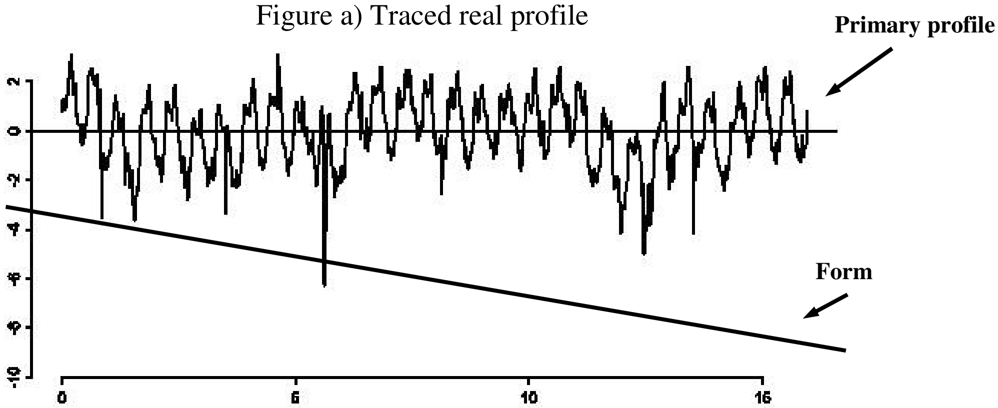


Figure b) Primary profile and form

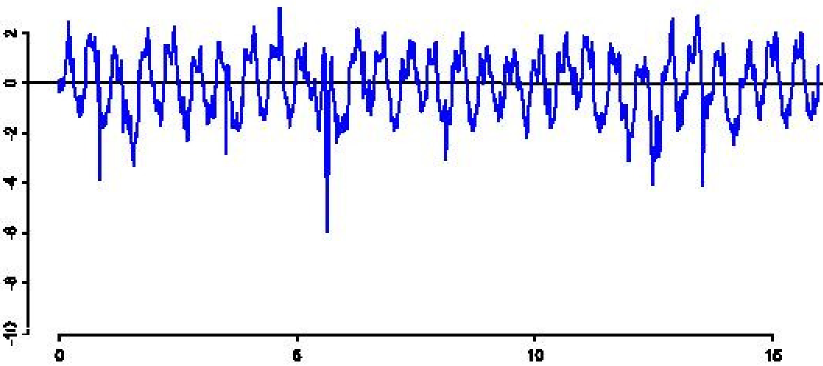


Figure c) Roughness profile

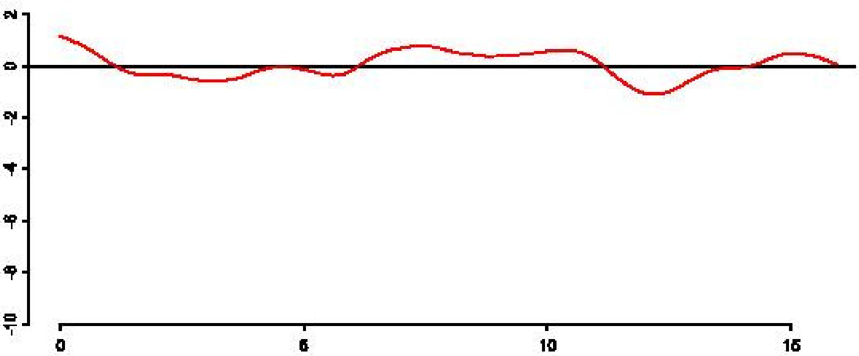


Figure d) Waviness profile

Figure 5.2 Roughness, waviness and primary profiles

5.1.2.2 Three groups of surface texture parameters

There are three principal groups of surface texture parameters: profile parameters defined in ISO 4287 [3], motif parameters defined in ISO 12085 [4] and parameters based on material ratio curve defined in ISO 13565-2 [5] and ISO 13565-3 [6].

Profile parameters

Profile parameters are used to quantify certain aspects of surface finish, including characteristics of the profile such as the height of the profile peak, the depth of the adjacent profile valley, and the spacing between them, see figure 5.3. Each profile element is characterised by a height Zj and a width Wi. Profile parameters include [2]:

Amplitude parameters determined solely by peak or valley heights, or both, irrespective of horizontal spacing, e.g. Ra (the arithmetic average value of the absolute departure of the profile from the reference line throughout the sampling length).

Spacing parameters determined solely by spacing of irregularities along the surface, e.g. Rsm (the average spacing of profile elements measured along the measurement direction).

Hybrid parameters determined by amplitude and spacing in combination, e.g. Rda (the arithmetical mean of the absolute value of slope calculated at each of the points in the profile within the sampling length).

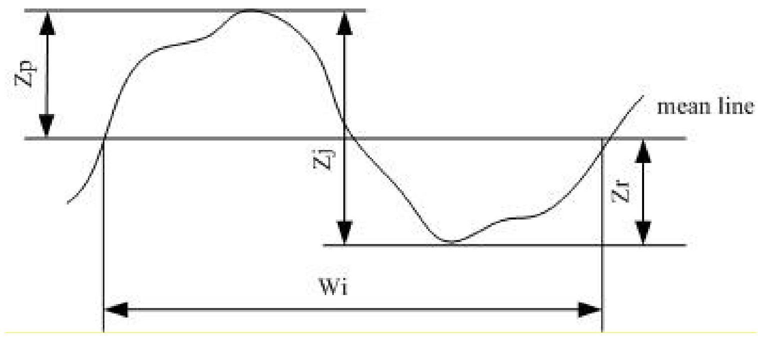


Figure 5.3 Profile element

Motif parameters

Motif is a potion of the primary profile between the highest points of two local peaks of the profile, which are not necessarily adjacent, see figure 5.4 [4]. Each motif is characterised by its two heights, H_j and H_{j+1}, and a width, AR_i. Only motifs that play a functional role are retained, i.e. any peaks or valleys determined by the analysis to be insignificant are removed from the result [2], once the motifs have been determined, the following parameters are calculated: R (mean motif height), AR (mean motif width) and Rx (maximum motif height).

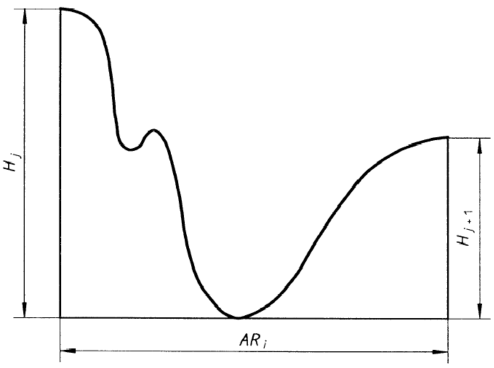


Figure 5.4 Motif parameters

Parameters based on material ratio curve

The relative material ratio curve describes the percentage of material traversed by a cut at level c located with respect to the highest point on the profile, see figure 5.5, Mr1 is the percentage of material traversed by a cut at level c1. This curve is equally called the Abbott-Firestone curve. It is the cumulative depth distribution function of the profile [7]. A material ratio curve can be plotted to provide a means of distinguishing different shapes of profile [2] and indicate bearing properties of the surface. Parameters calculated from the material ratio curve include Rk, Rpk, Rvk and etc, see figure 5.5.

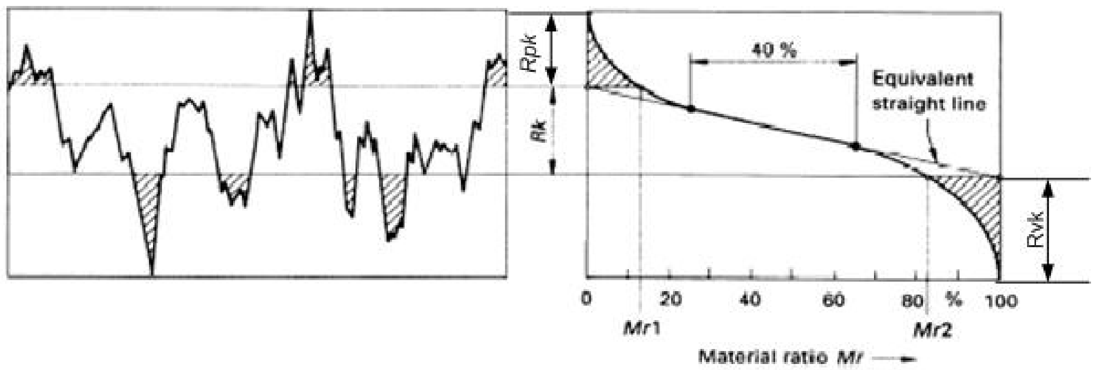


Figure 5.5 Parameters based on material ratio curve [5]

5.2 Knowledge acquisition

Knowledge acquisition is the first design stage. It involves acquiring knowledge from standards, books, references and experts, and then transferring this expertise from the sources to a knowledge-base.

An example of callout symbol is shown in figure 5.1. Many of the symbols in the callout have default values. These are values to be used when the symbol does not define all the required information. For example, “Ra 3.3”, has the complete representation “0.008-2.5 / Ra516% 3.3”, where the missing values are given by the default values [1]. Users, who are not familiar with the definitions, have to waste time looking up references in order to obtain the complete data information. The specification knowledge-base is for users to easily retrieve the necessary data information.

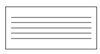
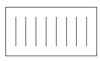
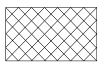
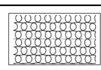
The knowledge of specification includes five parts: partition of the measured surface, extraction of a finite number of data points, filtration of the profile, the measurand to be indicated and choosing a suitable comparison rule. The sources come from ISO 1302 [1], ISO 3274 [8], ISO 4288 [9] and ISO 12085 [4]. The refinement processes for each part are discussed below:

5.2.1 Partition

The feature operation partition is used to identify the bounded surface which is to be characterized. The surface texture is influenced by the detailed form of the profile curve, while the profile curve is usually determined by manufacturing processes [1], therefore, the feature information of partition includes the direction of surface texture lay, the manufacture type and manufacture methods of the surface, which catch the initial properties of the surface being evaluated.

*5.2.1.1 Direction: surface texture lay **(k)** (see figure 5.1)*

The different types of surface texture lay are listed in table 5.1 [1]. The direction of the lay is usually determined by the manufacturing processes used. For example, the manufacturing process milling always leaves the parallel surface lay.

Symbol	**Definition**	**Example**
═	Parallel to plane of projection of view in which symbol is used	
⊥	Perpendicular to plane of projection of view in which symbol is used	
X	Crossed in two oblique directions relative to plane of projection of view in which symbol is used	
M	Multi-directional	

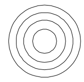
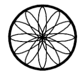
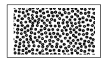
C	Approx. circular relative to centre of surface to which symbol applies	
R	Approx. radial relative to centre of surface to which symbol applies	
P	Lay is particulate, non-directional, or protuberant	

Table 5.1 Indication of surface lay

*5.2.1.2 Manufacture type **(j)** (see figure 5.1)*

There are three different manufacturing types, see figure 5.6 [1]: a) Any manufacturing process permitted; b) Material shall be removed. In this situation, material is allowed to be removed during manufacturing processes. For example, the manufacturing process milling shall remove material; c) Material shall not be removed. In this situation, material is prohibited to be removed during manufacturing processes. For example, the manufacturing process coating shall not remove material.

a) any manufacturing process permitted b) material shall be removed c) material shall not be removed

Figure 5.6 Indication of type of manufacturing process

*5.2.1.3 Manufacture methods **(l)** (see figure 5.1)*

The manufacturing process is a transformation process of raw materials into finished products, which greatly influences the produce surface lay, surface texture parameter values, etc (details see Chapter 9). The typical manufacturing processes include several units: casting processes, moulding processes, forming processes, machining processes, joining processes and rapid manufacturing.

This symbol in the callout indicates the manufacturing method, treatment, coatings or other requirements for the manufacturing process etc. to produce the surface, for example, turned, ground, plated [1]. Also this symbol links the specification knowledge-base and the manufacture knowledge-base. If manufacture methods on the specification are not indicated, the manufacture knowledge-base would be used to determine suitable manufacture processes matching the specification of the designed product.

After capturing the knowledge above, the refinement process of Partition is shown in figure 5.7. The abstract entity Partition in the simplified model of Specification is refined to the concrete data structures, which includes objects “direction symbol”, “direction definition”, “manufacture type symbol”, “manufacture type meaning” and “manufacture method”, and relationships among them.

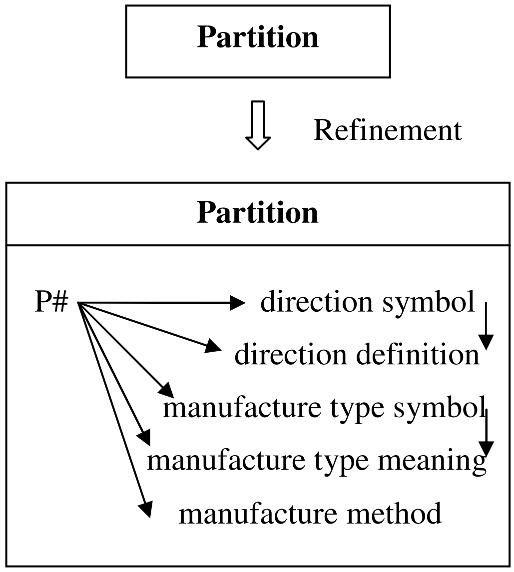


Figure 5.7 Refinement process of Partition

5.2.2 Extraction

The feature operation extraction is used to identify a finite number of points from the surface. The necessary information for carrying out the extraction includes the sampling length and the evaluation length of the evaluated surface.

5.2.2.1 Num_cutoff (f) (see figure 5.1)

Indicates the number of sampling lengths within the evaluation length. A cut-off is the wavelength which is used as a means of separating or filtering the wavelengths of a surface.

1) Profile parameters

Table 5.2 lists the indication of the number of sampling lengths for three profile parameters [1].

Profile	**Num_cutoff indication**
R-profile (roughness parameters)	If not otherwise indicated, the **default** number of cutoff wavelengths is **5** derived from ISO 4288 [9]. If the number of sampling lengths within the evaluation length differs from the default number of five, it shall be indicated adjacent to the relevant parameter designation.

	For example Rp3 or Rv3 or Rz3..., RSm3 ...all indicate that an evaluation length of three sampling lengths is desired.
W-profile (waviness parameters)	The number of sampling lengths shall always be indicated adjacent to the parameter designation of waviness. For example Wa3 or Wz3 ...all indicate that an evaluation length of three sampling lengths is desired.
P-profile (primary profile parameters)	The indication of the number of sampling lengths in the parameter designation of primary profile parameters **is not relevant**, as the evaluation length equals the sampling length and also equals the length of the feature being measured.

Table 5.2 Num_cutoff for profile parameters

2) Motif parameters

The indication of the num_cutoff is not relevant for motif parameters, because the operator used to calculate motif parameters has its own limit values and sampling length concepts do not exist [4].

3) Parameters based on material ratio curve

Table 5.3 lists the indication of the number of sampling lengths for parameters based on material ratio curve [5&6].

Profile	Num_cutoff indication
R-profile (roughness parameters)	If not otherwise indicated, the **default** number of cutoff wavelengths is **5** derived from ISO 13565-1 [10]. If the number of sampling lengths within the evaluation length differs from the default number of five, it shall be indicated adjacent to the relevant parameter designation. For example, Rk3 or Rpk3 ...all indicate that an evaluation length of three sampling lengths is desired.
P-profile (primary profile parameters)	The indication of the number of sampling lengths in the parameter designation of primary profile parameters is not relevant, as the evaluation length equals the sampling length and also equals the length of the feature being measured.

Table 5.3 Num_cutoff for parameters based on material ratio curve

5.2.2.2 Sampling length

Sampling length is the length in the direction of the X-axis used for identifying the irregularities characterizing the profile under evaluation [3], see figure 5.8. The feature being analyzed is cut into equal sample lengths, the sample lengths have the same numeric value as the cut-off.

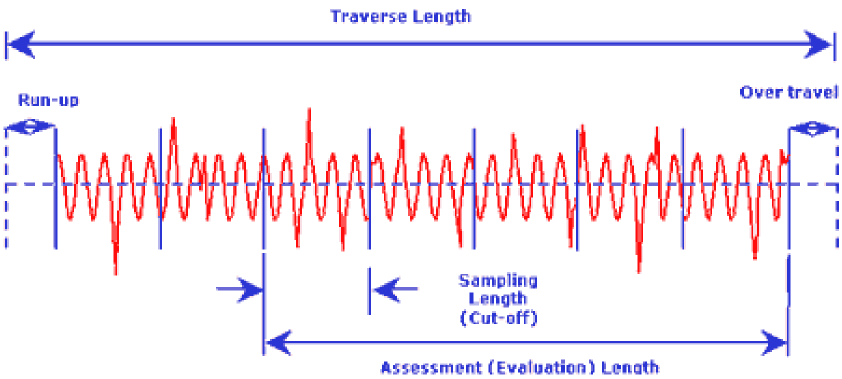


Figure 5.8 Example of traverse length, evaluation length and sampling length [11]

1) Profile parameters

Table 5.4 lists the value of the sampling length for profile parameters [1].

Profile	Sampling length
R-profile	The sampling length may be indicated as the upper limit λc in the callout symbol c, see figure 5.1. If there is no indication in the callout, tables 5.5 ~ 5.7 can be used to choose the roughness sampling length from the measured parameter values, according to ISO 4288 [9]. For example take the surface parameter Ra with a limit value of 3.3 micrometers, according to table 5.5, the parameter value belongs to the range of 2 < Ra ≤ 10, and the related sampling length shall be 2.5 millimetres.
W-profile	There are no defaults for waviness sampling length given in ISO standards, the sampling length is indicated as the upper limit in the callout symbol c, see figure 5.1. For example, 0,8-25 / Wz3 10, the sampling length 25 millimetres is indicated as the upper limit in the callout symbol.
P-profile	In the default case, *P*-parameters do not have any sampling lengths. It may be indicated if required for the function of the workpiece where it is indicated as the upper limit in the callout symbol c, see figure 5.1. For example -25 / Pz 225, the sampling length 25 millimetres is indicated.

Table 5.4 Sampling lengths for profile parameters

Ra （μm）	Roughness sampling length Lr （mm）	Roughness evaluation length Ln （mm）
(0,000) < Ra ≤ 0,02	0,08	0,4
0,02 < Ra ≤ 0,1	0,25	1,25
0,1 < Ra ≤ 2	0,8	4
2 < Ra ≤ 10	2,5	12,5
10 < Ra ≤ 80	8	40

Table 5.5 Roughness sampling lengths for the measurement of Ra, Rq, Rsk, Rku, RΔq and curves and related parameters for non-periodic profiles (for example ground profiles)

Rz, Rz1max（μm）	Roughness sampling length Lr（mm）	Roughness evaluation length Ln (mm)
0,025 < Rz,Rz1max ≤ 0,1	0,08	0,4
0,1 < Rz,Rz1max ≤ 0,5	0,25	1,25
0,5 < Rz,Rz1max ≤ 10	0,8	4
10 < Rz,Rz1max ≤ 50	2,5	12,5
50 < Rz,Rz1max ≤ 200	8	40
1) Rz is used when measuring Rz, Rv, Rp, Rc and Rt 2) Rz1max is used when measuring Rz1max, Rv1max, Rp1max and Rc1max		

Table 5.6 Roughness sampling lengths for the measurement of Rz, Rv, Rp, Rc and Rt of non-periodic profiles (for example ground profiles)

RSm (μm)	Roughness sampling length Lr (mm)	Roughness evaluation length Ln (mm)
0,013 < RSm ≤ 0,04	0,08	0,4
0,04 < RSm ≤ 0,13	0,25	1,25
0,13 < RSm ≤ 0,4	0,8	4
0,4 < RSm ≤ 1,3	2,5	12,5
1,3 < RSm ≤ 4	8	40

Table 5.7 Roughness sampling lengths for the measurement of R-parameters of periodic profiles, and RSm of periodic and non-periodic profiles

2) Motif parameters

Motif parameters do not use the concept of sampling length. The operator used to calculate motif parameters has its own limit values and sampling length concepts do not exist [4].

3) Parameters based on material ratio curve

Table 5.8 lists the value of the sampling length for parameters based on material ratio curve [5&6].

Profile	Sampling length
R-profile	If not otherwise indicated, the default sampling length for parameters based on material ratio curve is 0,8 millimetres derived from ISO 13565-1 [10].
P-profile	In the default case, *P*-parameters do not have any sampling lengths. The sampling length equals the evaluation length and also equals the length of the feature being measured.

Table 5.8 Sampling lengths for parameters based on material ratio curve

5.2.2.3 Evaluation length Ln

An evaluation length is the length of the profile left after filtering that is used for further analysis, see figure 5.8 [11].

1) Profile parameters

Table 5.9 lists the value of the evaluation length for profile parameters [1].

Profile	Evaluation length
R-profile	If not otherwise indicated, the default length of the feature for roughness analysis consists of five sample lengths; otherwise, the evaluation length equals the num_cutoff x sampling length. i.e. evaluation length = num_cutoff x sampling length For example take the surface parameter Ra with a limit value of 3.3 micrometers, i.e. Ra 3,3, according to table 5.5, the sampling length is 2.5 millimetres, and num_cutoff uses the default value 5, therefore, the evaluation length for this parameter is 5 x 2.5 = 12.5 millimetres.
W-profile	The evaluation length of the waviness profile equals the num_cutoff x sampling length of the waviness profile. i.e. evaluation length = num_cutoff x sampling length For example, 0,8-25 / Wz3 10, the num_cutoff is indicated as 3 adjacent to the parameter designation Wz, and the sampling length 25 millimetres is indicated as the upper limit in the callout symbol, therefore, the evaluation length is 3 x 25 = 75 millimetres.
P-profile	For primary profiles, the evaluation length equals the sampling length and also equals the length of the feature being measured. i.e. evaluation length = sampling length For example -25 / Pz 225, the evaluation length equals the sampling length of 25 millimetres as indicated in the callout.

Table 5.9 Evaluation lengths for profile parameters

2) Motif parameters

If not otherwise indicated, the default evaluation length is 16 millimetres. If a value other than the default is required, it shall be indicated between two oblique strokes between the bandwidth of the filter and the parameter designation [4].

e.g. 0,008-0,5/**12**/R 10, the evaluation length is indicated as 12 millimetres.

3) Parameters based on material ratio curve

Table 5.10 lists the value of the evaluation length for parameters based on material ratio curve [5&6].

Profile	Evaluation length
R-profile	The evaluation length of the roughness profile equals the num_cutoff x sampling length of the roughness profile. The default num_cutoff of the roughness profile equals five and the default sampling length of the roughness profile is 0,8 millimetres. i.e. evaluation length = num_cutoff x sampling length
P-profile	For primary profiles, the evaluation length equals the sampling length which is also equal to the length of the feature being measured. i.e. evaluation length = sampling length

Table 5.10 Evaluation lengths for parameters based on material ratio curve

After capturing the knowledge above, the refinement process of Extraction is shown in figure 5.9. The abstract entity Extraction in the simplified model of Specification is refined to the concrete data structures, which includes objects "num_cutoff", "sampling length" and "evaluation length", and relationships among them.

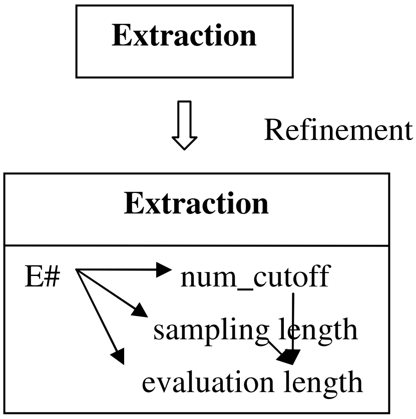


Figure 5.9 Refinement process of Extraction

5.2.3 Filtration

The feature operation filtration is used to separate the surface profile into roughness profile and waviness profile.

*5.2.3.1 Filter type **(b)** (see figure 5.1)*

The standardized filter is the Gaussian filter. The former standardized filter was the 2RC-filter [1]. In the future, other filter types may be standardized. In the transition period it may be convenient for some companies to indicate the filter type on drawings. Filter type may be indicated as "Gaussian" or "2RC". This is not standardized, but an indication of filter name as proposed here is unambiguous.

*5.2.3.2 The transmission band **(c)** (see figure 5.1)*

The transmission band consists of all the required wavelengths and is defined at the short wavelength by a short wavelength filter, while at the long wavelength by a long wavelength filter [1]. See figure 5.10, the transmission band for roughness profiles consists of a short wavelength filter λs and a long wavelength filter λc; while the transmission band for waviness profiles consists of a short wavelength filter λc and a long wavelength filter λf.

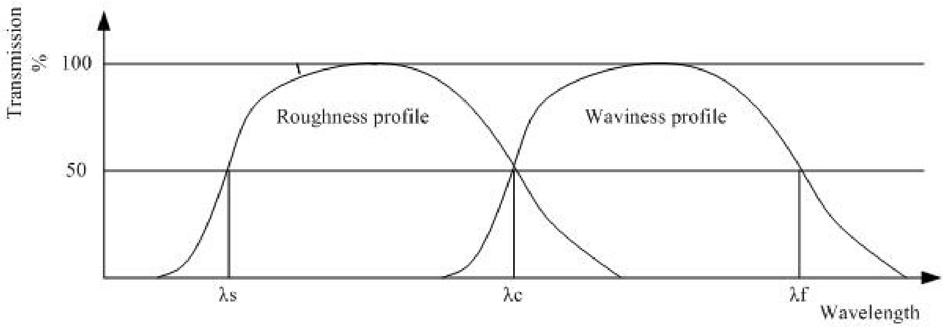


Figure 5.10 Transmission band [3]

The transmission band shall be indicated by the inclusion of the cut-off values of the filters (in millimetres), separated by a hyphen ("-"), the short-wave filter indicated first, and the long-wave filter second.

For example "0,0025-0,8" indicates a short-wave cut-off value of 0.0025 millimetres and a long-wave cut-off value of 0.8 millimetres, which will allow wavelengths between 0.0025mm and 0.8mm to be assessed with wavelengths below 0.0025mm and above 0.8mm being reduced in amplitude.

In some cases, it may be relevant to indicate only one of the two filters in the transmission band. The second filter then takes the default value, if it exists. If only one filter is indicated, the hyphen is maintained to indicate whether the indication is of the short-wave or the long-wave filter.

For example "0,008-" means only the short-wave cut-off value is indicated, and it is 0.008 millimetres. "-0,25" means only the long-wave cut-off value is indicated, and it is 0.25 millimetres.

Where no transmission band is indicated in connection with the parameter designation, the default transmission band applies to the surface texture requirement. The following steps illustrate how to choose the default values of upper and lower limit for the transmission band.

Upper limit of the transmission band

1) Profile parameters

For both roughness profiles and waviness profiles, the cut-off value of the upper limit equals the sampling length [3], see section 5.2.2.2.

i.e. Upper limit = sampling length

2) Motif parameters

Two bounds A and B are used in the motif algorithms according to ISO 12085 [4], to define respectively the maximum widths of the roughness and waviness motifs.

R-profile

The width for the roughness motif ARj should be less than or equal to the value A, and also greater than the value λs, see figure 5.11.

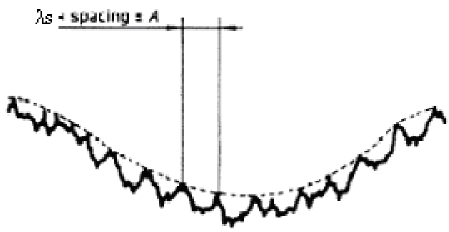


Figure 5.11 Roughness motifs

Upper limit A can be obtained from table 5.11 according to the evaluation length either indicated in the callout or referenced to the default value.

Evaluation length (mm)	A (mm)	B (mm)	λs (μm)
0,64	0,02	0,1	2,5
3,2	0,1	0,5	2,5
16	0,5	2,5	8
80	2,5	12,5	25

Table 5.11 Transmission band for motif parameters

W-profile

The width for the waviness motif AWj should be less than or equal to the value B, and also greater than the value A, see figure 5.12. Table 5.11 is reapplied to obtain the upper limit B for the waviness profile.

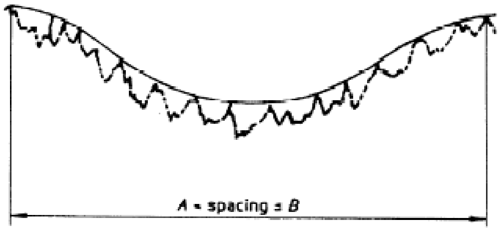


Figure 5.12 Waviness motifs

3) Parameters based on material ratio curve

For roughness profiles, the upper limit λc is defined equal to the sampling length according to ISO 1302 [1] and ISO 13565-1 [10]. The default sampling length for parameters based on material ratio curve is 0,8 millimetres, see section 5.2.2.2.

i.e. Upper limit = sampling length

Lower limit of the transmission band

1) Profile parameters

Table 5.12 lists the value of the lower limit for profile parameters [1].

Profile	Lower limit
R-profile	Lower limit λs may be indicated as the lower limit in the callout symbol c, see figure 5.1. If there is no indication in the callout, lower limit λs can be obtained from ISO 3274 [8] according to the value of upper limit λc, see table 5.13.
W-profile	The lower limit of the W-profile transmission band is λc (short-wave filter), and will be indicated as the lower limit in the callout symbol c, see figure 5.1.
P-profile	The lower limit of the P-profile of the transmission band is λs (short-wave filter), and will be indicated as the lower limit in the callout symbol c, see figure 5.1.

Table 5.12 Lower limit for profile parameters

λc (mm)	λs (µm)	λc/λs	r_{tip} max (µm)	Maximum sampling spacing ((µm)
0,08	2,5	30	2	0,5
0,25	2,5	100	2	0,5
0,8	2,5	300	2	0,5
2,5	8	300	5	1,5
8	25	300	10	6

Table 5.13 Relationship between the roughness cut-off wavelength λ_C, tip radius and roughness cut-off ratio λ_C/λ_s, ISO 3274 [8]

2) Motif parameters

Table 5.14 lists the value of the lower limit for motif parameters.

Profile	Lower limit
R profile	As mentioned in figure 5.11, the width for the roughness motif ARj should be greater than the value λs according to ISO 12085 [4]. The lower limit λs can be obtained from table 5.12 according to the evaluation length.
W profile	As mentioned in figure 5.12, the width for the waviness motif AWj should be greater than the value A according to ISO 12085 [4]. The lower limit A can be obtained from table 5.12 according to the evaluation length.

Table 5.14 Lower limit for motif parameters

3) Parameters based on material ratio curve

Table 5.15 lists the value of the lower limit for parameters based on material ratio curve.

Profile	Lower limit
R profile	If not otherwise indicated, the default lower limit λs for roughness profiles is 0,0025 millimetres according to ISO 1302 [1] and ISO 13565-1 [10].
P profile	The lower limit for primary profiles of the transmission band is λs (short-wave filter), which has no default value to be defined according to ISO 1302.

Table 5.15 Lower limit for parameters based on material ratio curve

After capturing the knowledge above, the refinement process of Filtration is shown in figure 5.13. The abstract entity Filtration in the simplified model of Specification is refined to the concrete data structures, which includes objects “filter type”, “up limit (upper limit)” and “low limit (lower limit)”, and relationships among them. The object “up limit” in the entity Filtration also has the relationship ① with the object “sampling length” in the entity Extraction, (see section 5.2.6 for details).

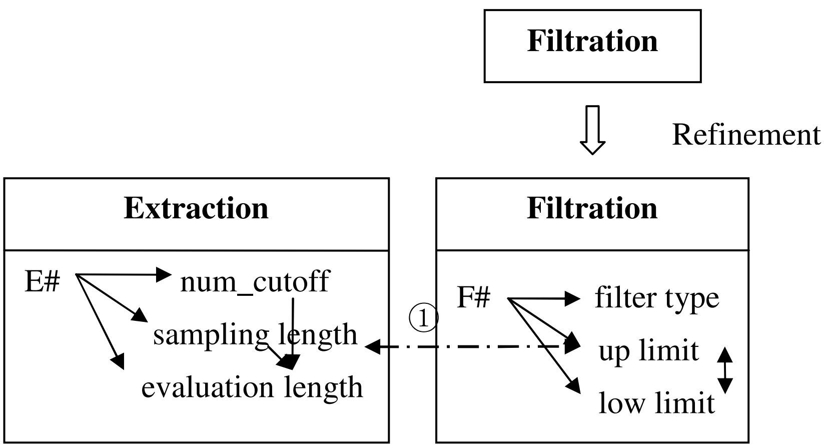


Figure 5.13 Refinement process of Filtration

5.2.4 Measurand

Measurand consists of the specified value for the indicated parameter, which is used to characterise specific aspects of the surface.

5.2.4.1 Tolerance type ***(a)*** *(see figure 5.1)*

If not otherwise indicated, the default tolerance type is upper limit "U". There are two types of tolerance limit for a surface, the upper tolerance limit and the lower tolerance limit. The indication can be of an upper type with indication U or of a lower type with indication L according to ISO 1302 [1].

*5.2.4.2 Parameter (**d** – profile indication, **e** – characteristic indication) (see figure 5.1)*

There are three possible profile indications R, W and P designated in the callout symbol d, see figure 5.1: R is the indication of a roughness profile, W is the indication of a waviness profile and P is the indication of a primary profile, see figure 5.2 in section 5.1.2.1. There are also three principal groups of surface texture parameters: profile parameters defined in ISO 4287 [3], motif parameters defined in ISO 12085 [4] and parameters based on material ratio curve defined in ISO 13565-2 [5] and ISO 13565-3 [6], see section 5.1.2.2.

1) Profile parameters

Table 5.16 gives the detail designation of profile parameters according to ISO 4287 [3], Taylor-Hobson [11].

Amplitude parameters: measures of the vertical characteristics of the surface deviations. For example, Ra is the arithmetic average value of the absolute departure of the profile from the reference line throughout the sampling length, and Rq is the root mean square of the distance of the filtered or unfiltered profile from its mean line.

Spacing parameters: measures of the horizontal characteristics of the surface deviations. For example Rsm is the average spacing of profile elements measured along the measurement direction.

Hybrid parameters: a combination of both the vertical and horizontal characteristics of the surface deviations. For example Rda is the arithmetical mean of the absolute value of slope calculated at each of the points in the profile within the sampling length.

Curves and related parameters: measures of the percentage of the length of the bearing surface at any specified depth in the profile to the profile length. For example

Rmr is the length of material surface (expressed as a percentage of the evaluation length L) at a specific depth below a reference level.

Parameter	**Amplitude**									**Spacing**	**Hybrid**	**Curves and related**		
	Peak-valley					Mean value								
R-profile parameters	*Rp*	*Rv*	*Rz*	*Rc*	*Rt*	*Ra*	*Rq*	*Rsk*	*Rku*	*RSm*	*RΔq*	*Rmr (c)*	*Rδc*	*Rmr*
W-profile parameters	*Wp*	*Wv*	*Wz*	*Wc*	*Wt*	*W a*	*Wq*	*Wsk*	*Wku*	*WSm*	*WΔq*	*Wm r(c)*	*Wδc*	*Wmr*
p-profile paramters	*Pp*	*Pv*	*Pz*	*Pc*	*Pt*	*Pa*	*Pq*	*Psk*	*Pku*	*PSm*	*PΔq*	*Pmr (c)*	*Pδc*	*Pmr*

Table 5.16 Profile parameter names and types

2) Motif parameters

Table 5.17 lists the motif parameter types and names according to ISO 12085 [4], see section 5.1.2.2.

Parameters	**Amplitude**			**Spacing**
Roughness profile (roughness motif parameters)	*R*	*Rx*	—	*AR*
Waviness profile (waviness motif parameters)	*W*	*Wx*	*Wte*	*AW*

Table 5.17 Motif parameter names and types

3) Parameters based on material ratio curve

Table 5.18 lists the parameters based on material ratio curve according to ISO 13565-2 [5] and ISO 13565-3 [6]. These parameters are calculated from the material ratio curve as illustrated in figure 5.5, see section 5.1.2.2.

<table>
<tr><th></th><th colspan="5">Parameters</th></tr>
<tr><td rowspan="2">R-profile parameters based on linear material ratio curve</td><td>Rk</td><td>Rpk</td><td>Rvk</td><td>Mr1</td><td>Mr2</td></tr>
<tr><td>Rke</td><td>Rpke</td><td>Rvke</td><td>Mr1e</td><td>Mr2e</td></tr>
<tr><td>R-profile parameters based on the material probability curve</td><td colspan="2">Rpq</td><td colspan="2">Rvq</td><td>Rmq</td></tr>
<tr><td>P-profile parameters based on the material probability curve</td><td colspan="2">Ppq</td><td colspan="2">Pvq</td><td>Pmq</td></tr>
</table>

Table 5.18 Parameters based on material ratio curve

*5.2.4.3 Value **(h)** (see figure 5.1)*

Indicates the tolerance limit of the indicated parameter for a surface. The value of the limit is always given in micrometres, for example Ra 3,3 in figure 5.1 means the tolerance limit for the parameter Ra is 3.3 micrometers, which will be applied for the comparison rule explained in section 5.2.5.

5.2.4.4 Machining allowance ***(i)*** *(see figure 5.1)*

The machining allowance is a planned deviation between an actual dimension and a nominal dimension, and generally indicated only in those cases where more process stages are shown in the same drawing [1]. It allows an area of excess metal to be left to complete subsequent machining. For example, the outer diameter of a pin may be ground to 0.0005 inches oversize because it is known that subsequent heat-treatment of the pin is going to cause it to shrink by 0.0005 inches [12]. The machining allowance is indicated in millimetres.

After capturing the knowledge above, the refinement process of Measurand is shown in figure 5.14. The abstract entity Measurand in the simplified model of Specification is refined to the concrete data structures, which includes objects "tolerance type", "parameter type", "parameter name", "value" and "machine allowance", and relationships among them. Objects in the entity Measurand also have the relationship ② with objects in the entity Extraction and ③ with objects in the entity Filtration (see section 5.2.6 for details).

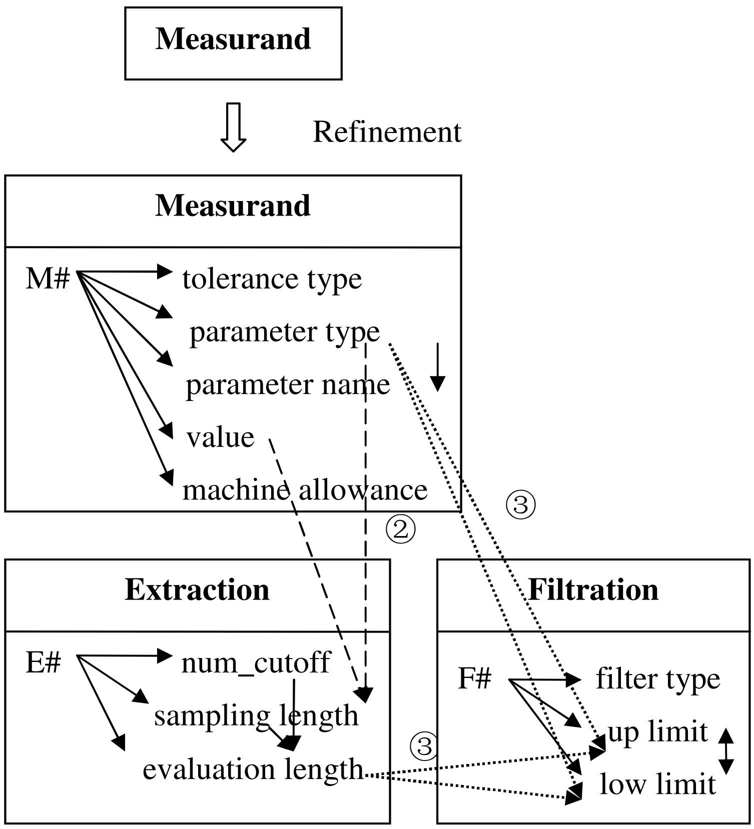


Figure 5.14 Refinement process Measurand

5.2.5 Comparison rule

Comparison rule is used to compare the measured value with the value indicated on the specification, and determine whether the real surface is within tolerance.

5.2.5.1 Rule type **(g)** *(see figure 5.1)*

There are two types of comparison rules, “16 %-rule” and “max-rule”. If not otherwise indicated, the “16 %-rule” is defined as the default rule for all indications of surface texture requirements. If the “max-rule” is required, the indication “max” will be indicated in the callout symbol g, see figure 5.1.

5.2.5.2 Rule interpretation

“16%-rule”: for requirements specifying an upper limit of a parameter, the surface is considered acceptable if not more than 16% of all the measured values, based on the evaluation length, exceed the value specified on the specification. Conversely, for requirements specifying a lower limit of a parameter, the surface is considered acceptable if not more than 16% of all the measured values, based on the evaluation length, are less than the value specified on the specification.

“max-rule”: where the requirements specify a maximum value of the parameter, none of the measured values of the parameter over the entire surface can exceed the value specified.

After capturing the knowledge above, the refinement process of Comparison rule is shown in figure 5.15. The abstract entity Comparison rule in the simplified model of Specification is refined to the concrete data structures, which includes objects “rule type”, and “rule interpretation”, and the relationship between them.

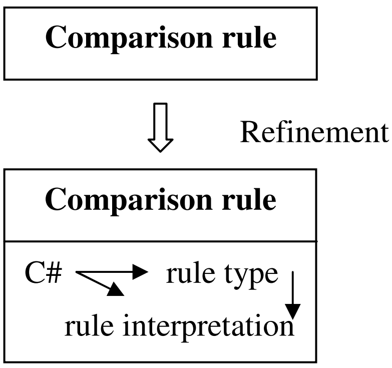


Figure 5.15 Refinement process of Comparison rule

5.2.6 Relations in the specification knowledge-base

The relationships refined in the concrete model of Specification are shown in figure 5.13 and figure 5.14 and will be illustrated below:

Relationship ① — equals: Relationship between entities Extraction and Filtration

The upper limit of the transmission band in Filtration equals the sampling length in Extraction for both profile parameters and parameters based on material ratio curve, see section 5.2.2.2 and 5.2.3.2. For example take the surface parameter Ra with a limit value of 3.3 micrometers, the sampling length shall be the default value of 2.5 millimetres according to ISO 4288 [9]; and as the upper limit of the transmission band equals the sampling length, it is equal to 2.5 millimetres.

i.e. upper limit = sampling length

Relationship ② — determines: Relationship between entities Extraction and Measurand

Parameter type and value in Measurand determine the sampling length in Extraction as shown in table 5.5 – 5.7 for profile parameters, see section 5.2.2.2. For example take the surface parameter Ra with a limit value of 3.3 micrometers, the parameter type is Ra, according to table 5.5, the parameter value belongs to the range of $2 < Ra \leq 10$, therefore the related sampling length shall be 2.5 millimetres.

i.e. parameter type + value → sampling length

Relationship ③ — determine: Relationship among entities Extraction, Filtration and Measurand

Evaluation length determines both the upper limit and lower limit for motif parameters as shown in table 5.11, see section 5.2.3.2. For example, 0,008-0,5/80/R 10, the evaluation length is indicated as 80 millimetres, according to table 5.11, the related upper limit A for the roughness profile is 2.5 millimetres, and the related lower limit λs for the roughness profile is 25 micrometres.

i.e. evaluation length → upper limit

evaluation length → lower limit

Relationship ④ — callout: Relationship among entities Partition, Extraction, Filtration, Measurand and comparison rule (see figure 5.20)

Callout is the product of direction symbol, manufacture type symbol, manufacture method, num_cutoff, filter type, upper limit, lower limit, tolerance type, parameter type, value, machine allowance and rule type, see figure 5.1.

5.3 Knowledge representation

5.3.1 Functional Model of Callout

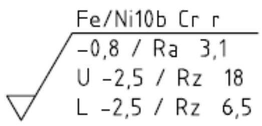
Chapter 2 mentioned that the functional data model provides what is probably the most natural query language DAPLEX [13], based on function composition, it is intended that a manipulation language can be developed based on DAPLEX, thereby providing a conceptually natural query language for the category model.

5.3.1.1 Entities and functions

In Chapter 4, the callout symbol is designed to take the form of a hierarchical structure. The detail information including in the structure are as follows:

Entities:

A callout includes several simple callout symbols, for example, a callout



, includes three simple callouts: “-0,8 / Ra 3,1”, “U -2,5 / Rz 18”, and “L – 2,5 / Rz 6,5”. A simple callout symbol contains the feature of the surface, the tolerance and the comparison rules.

i. Feature: There are two ways to describe a surface feature, the profile and the areal (the standards on areal are currently being developed). A profile consists of the direction of surface lay, the filter and the evaluation length. The filter comprises the filter type and the bandwidth: lower limit and upper limit.
ii. Tolerance: The tolerance includes the parameter, an indication of upper or lower specification limits and the limit value. The parameter has a parameter name and a type.
iii. Comparison rules: The comparison rule contains the evaluation length and the type of comparison rule.

Functions:

i. The parameter type and the limit value determine the bandwidth of the filter.

ii. The evaluation length can be obtained from the number of sampling length and the value of upper limit (sampling length).

5.3.1.2 Functional Model by P/FDM

P/FDM [14] is an implementation of Shipman's Functional Data Model and DAPLEX [13]. The callout knowledge-base is first structured using the functional model and tested through P/FDM. Figure 5.16 illustrates the functional model of a simple surface texture callout using P/FDM [14].

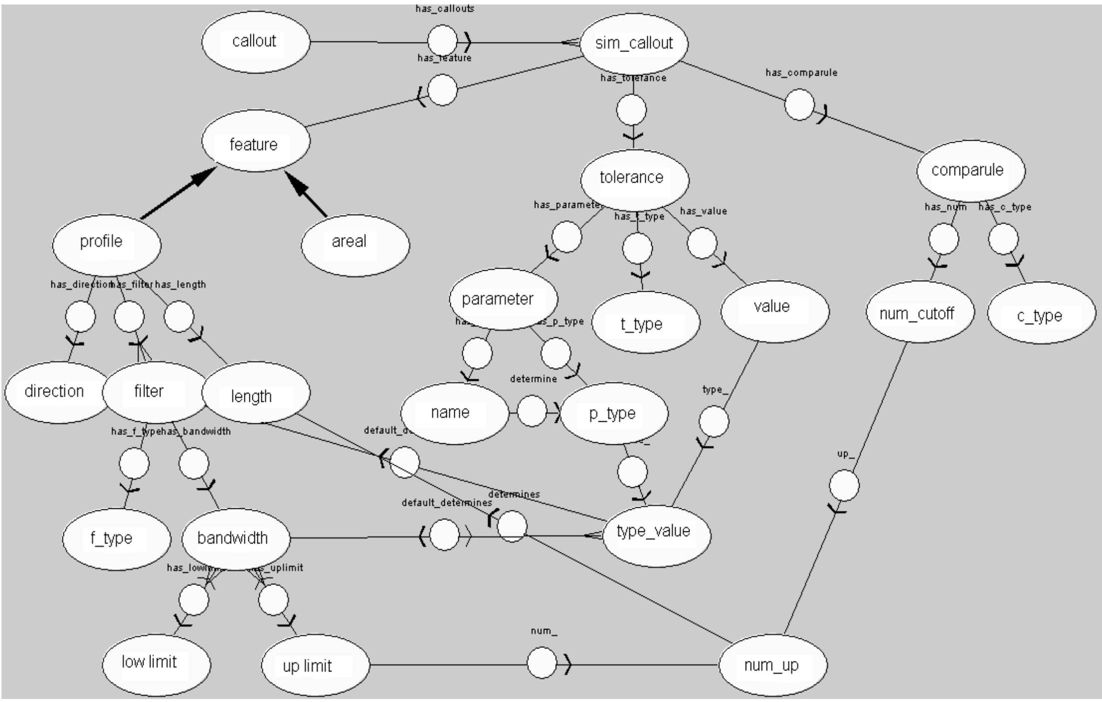


Figure 5.16 Functional Model of Callout

Figure 5.17 gives a typical schema example of how to define the knowledge-base and how to declare an entity, its attributes and relations with other entities. This particular example shows how to declare the entity of a parameter and value.

```
declare parameter ->> entity
declare has_name(parameter) -> name
declare has_p_type(parameter) -> p_type

declare value ->> entity
declare value(value) -> float
declare range(value) -> integer
declare type_(value,string) -> type_value
.
.
.
```

Figure 5.17 Schema to define the knowledge-base

Figure 5.18 shows an example of Daplex queries. It is used to retrieve the information from the knowledge-base. This particular example shows how to find the complete representation of the callout “Ra 3.3”.

```
for each n0 in name such that name(n0)="Ra"
 for each p1 in p_type such that p1=determine( n0, "p")
  for each v2 in value such that value(v2)=3.3
   for each t3 in type_value such that types(t3)=p_type(p1) and range(t3)=range(v2)
    for each b4 in bandwidth such that b4=default_determines(t3)
     for each u5 in uplimit such that u5=has_uplimit( b4)
      for each b6 in bandwidth such that b6=b4
       for each l7 in lowlimit such that l7=has_lowlimit( b6)
        for each d8 in direction such that d_name(d8)=""
         for each f9 in f_type such that f_name(f9)=""
          for each t10 in t_type such that t_name(t10)=""
           for each n11 in num_cutoff such that n_name(n11)=""
            for each c12 in c_type such that c_name(c12)=""
Print( direction(d8), f_type(f9), lowlimit(l7), uplimit(u5), name(n0), value(v2), num(n11), c_type(c12) );
```

Figure 5.18 Daplex Queries

Figure 5.19 shows the result of this operation, which lists the information required for a complete callout, including the bandwidth, evaluation length as the number of sampling lengths and the type of the comparison rule.

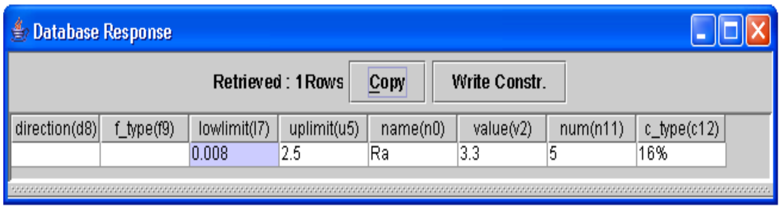


Figure 5.19 The Response

The functional model represents the structure of a simple callout and provides the schema and the query language to define the knowledge-base. Based on the functional model, category model and its manipulation language of specification can be developed.

5.3.2 Category Model of Specification

Following the refinement processes described above, the category model of specification knowledge-base has been developed. In this model, relations among structures are represented by the pullback structure.

5.3.2.1 Specification model

Specification knowledge-base comprises five entities: partition, extraction, filtration, measurand and comparison rule. The detailed knowledge information contained in each entity were specified in the previous sections. The model below in figure 5.20 represents the specification knowledge-base, based on category theory. Each entity is represented as an object with attributes in it. The arrows inside the objects represent the intra-object relations. ① , ②, ③ and ④ are the simplified pullback structures used to represent the relationships between objects.

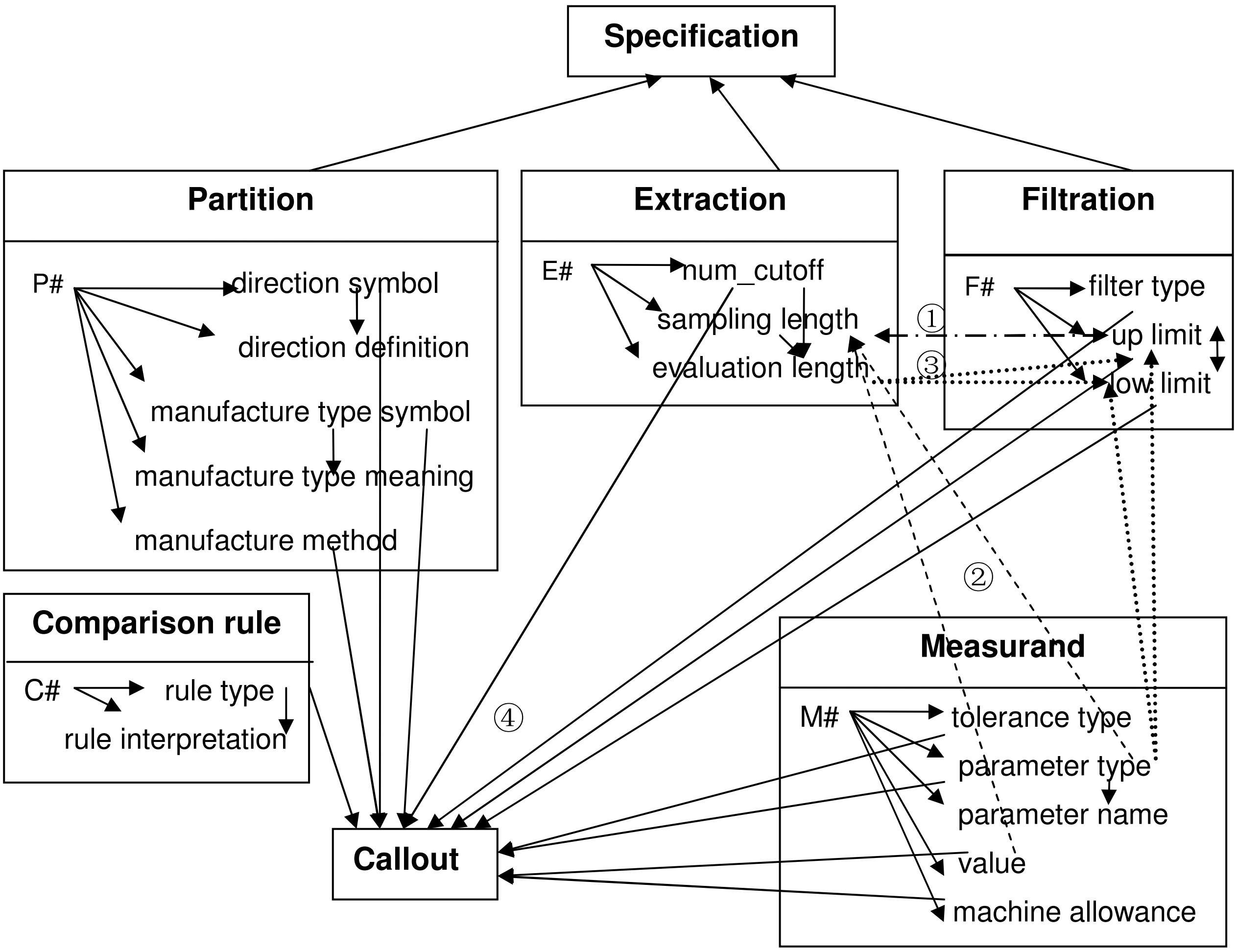


Figure 5.20 Specification model

5.3.2.2 Relationships in pullback structure

① equals – relationship between entities Extraction and Filtration, see figure 5.21:

The relationship p1 in figure 5.21 shows a pullback structure between Extraction and Filtration. Product p1 stores all the possible relations and extra information between Extraction and Filtration, where *equals:: sampling length * upper limit, equals* is the name of the product p1, and *sampling length * upper limit* is the type of the product p1, i.e., the equal relation is between the sampling length and the upper limit; $E\# *_{p1}$

F# is the restricted product, which stores the exact relation between the sampling length and the upper limit, i.e., the equal relation between sampling length and upper limit is only applied for both profile parameters and parameters based on material ratio curve.

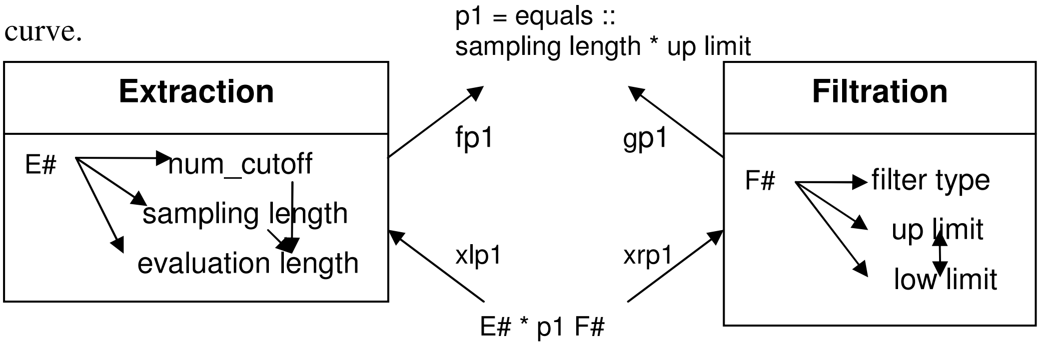


Figure 5.21 Relation “equals” in pullback structure

② determines - relationship between entities Extraction and Measurand, see figure 5.22:

Here the relationship is that the parameter type and the value in Measurand determine the sampling length in Extraction.

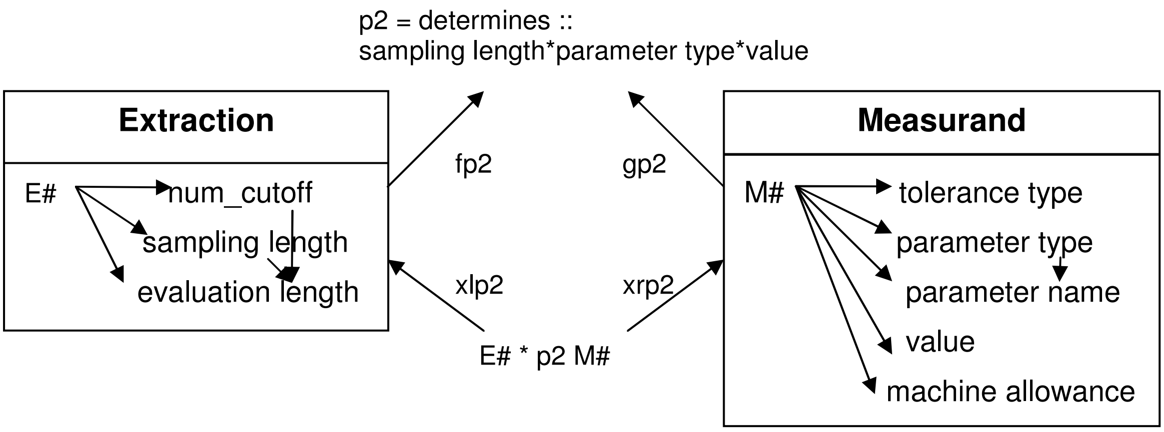


Figure 5.22 Relation “determines” in pullback structure

③ determine - relationship among entities Extraction, Filtration and Measurand, see figure 5.23:

Here the relationship is that the evaluation length determines both the upper limit and lower limit for motif parameters.

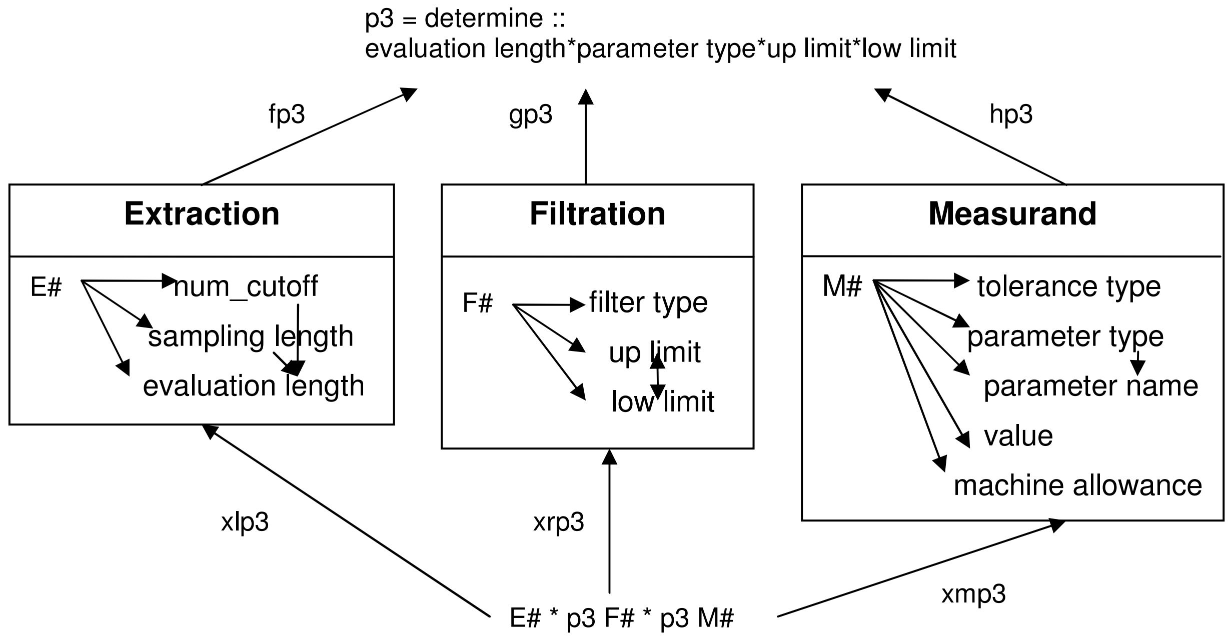


Figure 5.23 Relation "determine" in pullback structure

④ callout - relationship among entities Partition, Extraction, Filtration, Measurand and comparison rule, see figure 5.24:

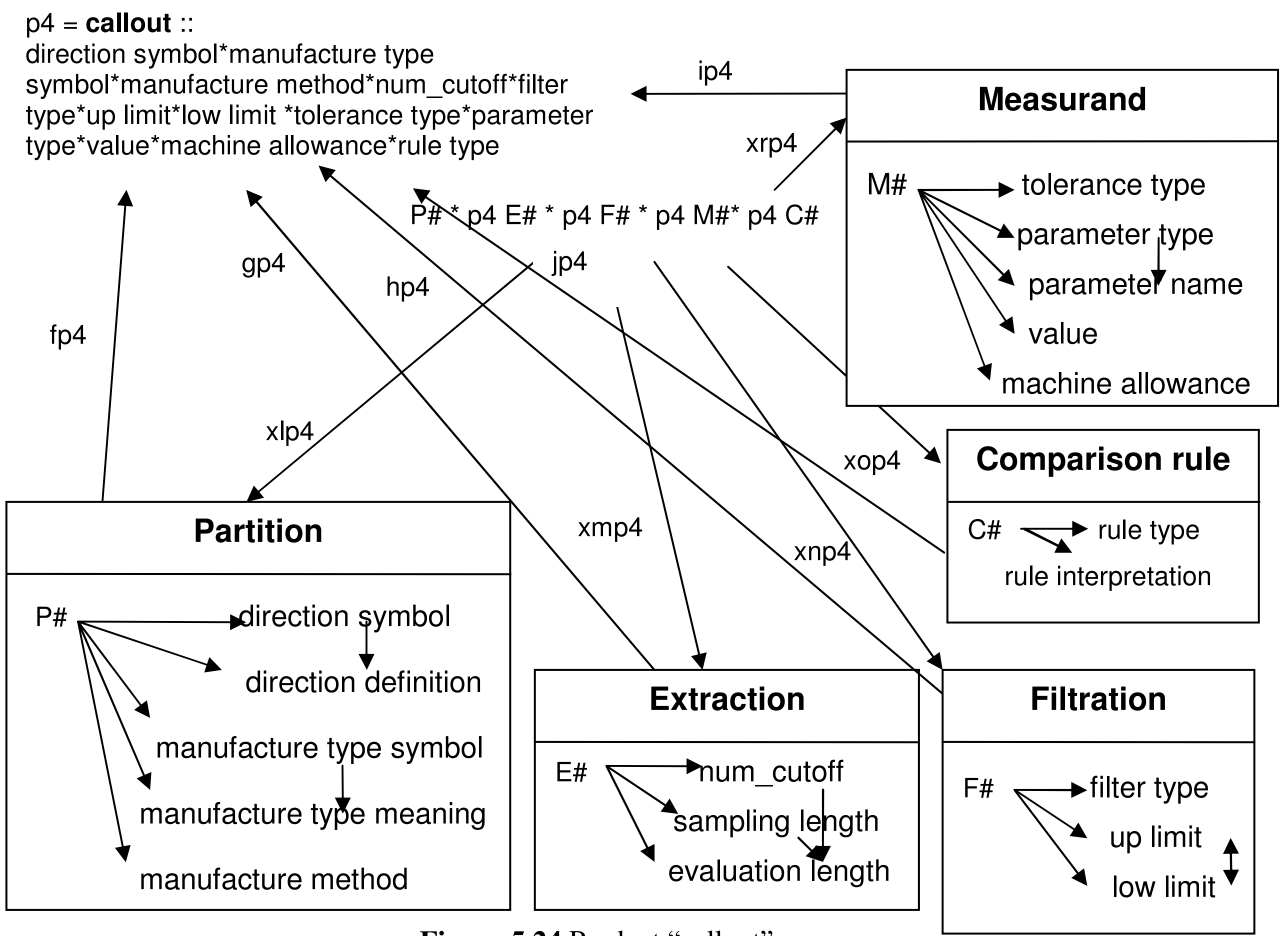


Figure 5.24 Product "callout"

Here the relationship is that the callout is the product of direction symbol, manufacture type symbol, manufacture method, num_cutoff, filter type, upper limit, lower limit, tolerance type, parameter type, value, machine allowance and rule type.

5.3.2.3 Operations of the specification knowledge-base

The main operations of the specification knowledge-based system are listed below:

1) Review the symbol definitions and contents of a callout. Users can obtain detail explanations and descriptions of each symbol contained in a callout, such as the definition of the direction and the names of different parameter types.
2) Retrieve a complete callout after entering a callout from a drawing and give the default values if available. For example, users can obtain the complete callout "U 0.008-2.5 / Ra516% 3.3" after entering the callout "Ra 3.3".
3) Help to understand the meaning of the symbols of a callout. For example, users can understand the differences between comparison rules "Max-rule" and "16%-rule".
4) Further refer to the function, the manufacture and the verification knowledge-base. The specification knowledge-based system is connected with the others. Users can easily traverse through them.

5.3.2.4 Controlling reasoning

The information and rules stored in specification knowledge-base are represented by the category model as shown in figure 5.20- figure 5.24. In order to realize the operations list in section 5.3.2.3, controlling reasoning is needed which involves deciding when to apply what piece of knowledge. This category model has the advantage of built-in rules and relations, which allows the system automatically to match the rules against the requirements, see the flowchart in figure 5.25:

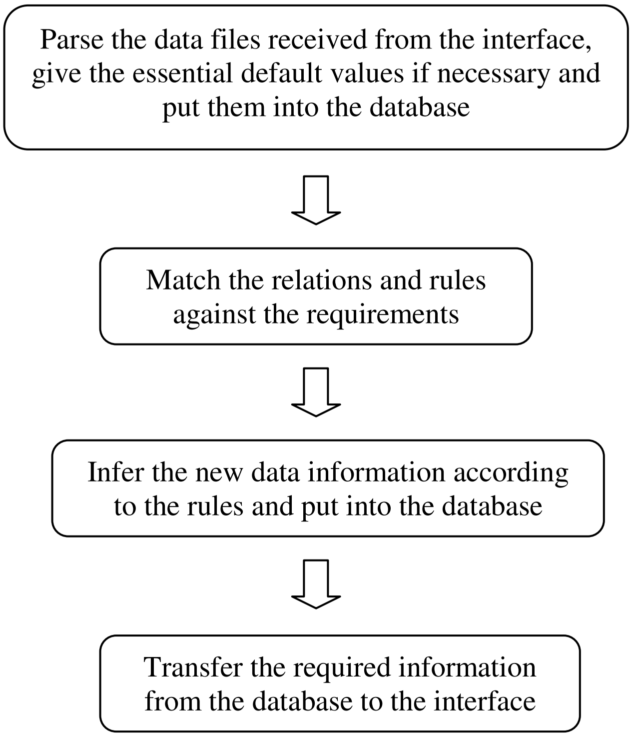


Figure 5.25 Flowchart of controlling reasoning

5.3.2.5 Queries of the specification knowledge-base

Two examples of queries are provided below, which are used to retrieve the required knowledge from the knowledge-base. [Hom-sets are defined as a set of arrows, each arrow being specified by its name or its source and target.]

1) Print the direction definition of the direction symbol '═' (see table 5.1):

   According to data model in the Partition entity, the object direction symbol determines the object direction definition, see section 5.2.1.1.

   In order to retrieve that information from the knowledge-base, one simple query is needed to access to the entity Partition. Firstly find the symbol '═' in the direction symbol database, and then search the matching direction definition and finally print it out.

   A → Partition

   Hom set = P# → direction symbol, P# → direction definition, direction symbol → direction definition

   Objects = P#, direction definition, direction symbol | direction symbol = '═'

   B → A

   Hom set = { }

Objects = direction symbol, direction definition

The query only needs to subset the category Partition, where the object direction symbol has the value of '═'. The output for this query is:

The definition for the symbol '═': Parallel to plane of projection of view in which symbol is used.

2) Print the simple callout symbol of 'Ra 3,3' (without manufacture methods, direction and machine allowance)

The missing values for the simple callout include the tolerance type, the bandwidth, and num_cutoff, which can be derived from the default values defined in ISO standards (ISO 1302 [1], ISO 4287 [3]), see section 5.2.2.1, 5.2.3.2, and 5.2.4.1.

When the users input 'Ra 3,3' on the interface, see figure 5.26, the inference engine will parse the data information, give the default values that num_cutoff is 5 and tolerance type is upper limit 'U', and then insert them into the database.

The following queries access to the entities Measurand, Filtration, Extraction, and the relations p1, p2 and p4. The detailed queries are given below:

A → Measurand

Hom set = M# → tolerance type, M# → parameter type, M# → value

Objects = M#, tolerance type, parameter type, value | tolerance type = 'U', parameter type = 'Ra', value = '3,3'

/* A is the subcategory of the category Measurand, where the tolerance type has the value of 'U', the parameter type has the value of 'Ra', and the value has the value of '3,3' */

B → P2

Hom set = xrp2

Objects = E# * p2 M#, E#, sampling length, M#, parameter type, value | parameter type ∈ A, value ∈ A

C → B

Hom set = { }

Objects = sampling length, E#

/* Subset the P2 pullback to include those objects with values only in A and obtain the related sampling length */

D → P1

Hom set = xlp1

Objects = E# * p1 F#, E#, sampling length, F#, upper limit | sampling length $\in$ C

E → D

Hom set = { }

Objects = upper limit, F#

/* Subset the P1 pullback to include those objects with values only in C and obtain the related upper limit*/

F → Filtration

Hom set = F# → upper limit, upper limit → lower limit

Objects = F#, lower limit | upper limit $\in$ E

/* Subset the category Filtration where the upper limit has the value in E */

G → Extraction

Hom set = E# → num_cutoff

Objects = E#, num_cutoff | E# $\in$ C, num_cutoff = '5'

/* Subset the category Extraction where the num_cutoff has the value of '5' */

H → P4

Hom set = xmp4, xnp4, xrp4

Objects = E# * p4 F4* p4 M4, num_cutoff, upper limit, lower limit, tolerance type, parameter type, value | num_cutoff $\in$ G, upper limit $\in$ E, lower limit $\in$ F, tolerance type $\in$ A, parameter type $\in$ A, value $\in$ A

I → H

Hom set = { }

Objects = num_cutoff, upper limit, lower limit, tolerance type, parameter type, value

/* Subset the P4 pullback to include those objects with values in G, E, F and A */

Inference engine is being used to infer the new data information during the queries. The output of above queries is shown in figure 5.27, which retrieves the simple callout as 'U 0,008-2,5 / Ra516% 3,3'.

5.4 Interface of the specification knowledge-base

The sample interface of specification knowledge-based system is shown in figure 5.26. The users can input the data already known into the blanks, and then click the 'Enter' button, the corresponding data information will output in the interface. Figure 5.27

shows an output interface. After the user input the callout ‘Ra 3,3’, and click the ‘Enter’ button, the simple complete callout ‘U 0,008-2,5 / Ra516% 3,3’ will be retrieved and displayed in the callout blank, and the other related detail specification information will be displayed as well.

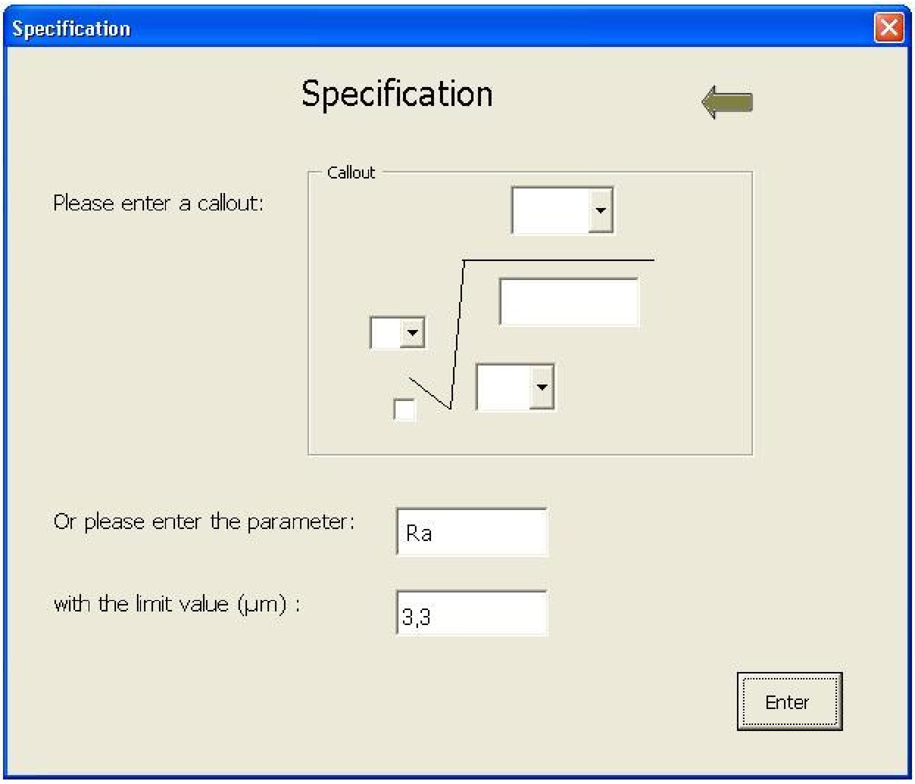


Figure 5.26 Interface of specification

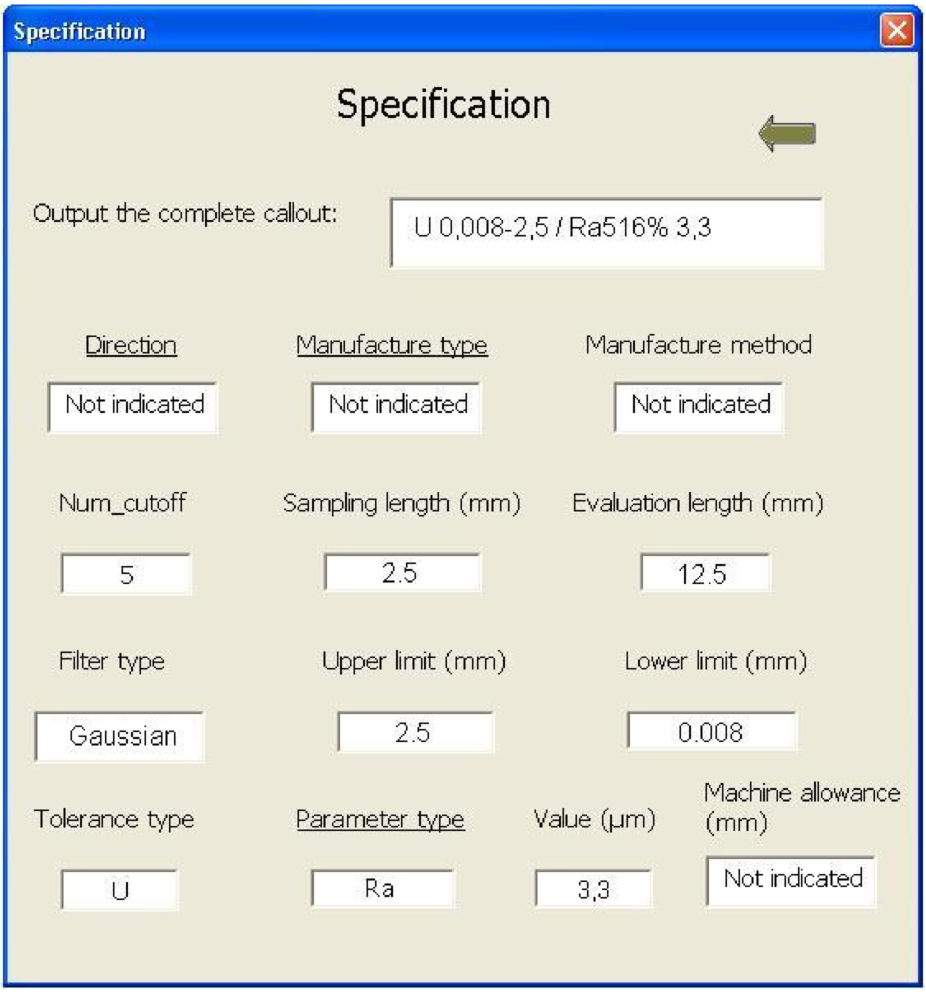


Figure 5.27 Output of the interface

5.5 Summary

This Chapter describes the detailed design procedures for specification knowledge-base, including:

Knowledge acquisition – the knowledge are mainly coming from ISO standards [1], [3-6], [8-9] and divided into five parts: partition, extraction, filtration, measurand and comparison rule. The refinement processes of each part are provided, which constitute the final specification category model.

Knowledge representation – the functional model of specification and its manipulation language DAPLEX are introduced, which provide a basis for the development of category model and its query languages. The complete category model represents the knowledge structure and rules for specification knowledge-base. Pseudo query language and interfaces complements the implementations of specification knowledge-base.

References

[1] ISO 1302 (2002), *Geometrical Product Specifications (GPS) — Indication of surface texture in technical product documentation,* International Organisation for Standardisation, Geneva

[2] M.A. Dagnall (1997), *Exploring surface texture – a fundamental guide to the measurement of surface finish*, Taylor Hobson Limited

[3] ISO 4287 (1997), *Geometrical Product Specifications (GPS) — Surface texture: Profile method — Terms, definitions and surface texture parameters*, International Organisation for Standardisation, Geneva

[4] ISO 12085 (1996), *Geometrical Product Specifications (GPS) — Surface texture: Profile method — Motif parameters*, International Organisation for Standardisation, Geneva

[5] 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 Organisation for Standardisation, Geneva

[6] ISO 13565-3 (2000), *Geometrical Product Specifications (GPS) — Surface texture: Profile method; Surfaces having stratified functional properties — Part 3: Height characterization using the material probability curve*, International Organisation for Standardisation, Geneva

[7] http://www.digital-surf.com/en/guideparam2d.htm

[8] ISO 3274 (1996), *Geometrical Product Specifications (GPS) - Surface texture: Profile method - Nominal characteristics of contact (stylus) instruments*, International Organisation for Standardisation, Geneva

[9] ISO 4288 (1996), *Geometrical Product Specifications (GPS) -Surface texture: Profile method -Rules and procedures for the assessment of surface texture*, International Organisation for Standardisation, Geneva

[10] 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 Organisation for Standardisation, Geneva

[11] http://www.taylor-hobson.com/faqsurfacedetail.php?faqid=27

[12] http://en.wikipedia.org/wiki/Allowance#Engineering_and_machining

[13] Shipman (1981), *The Functional Data Model and the Data language DAPLEX,* Computer Corporation of America, ACM Transactions on Database Systems, 6(1981) 140-173

[14] Object Database Group (1996), *P/FDM v.10.1*, University of Aberdeen, AUCS/TR9501

Chapter 6

Verification Refinement

6.1 Introduction

This chapter aims to provide a detailed explanation of refinement processes for verification knowledge-base, including knowledge acquisition and knowledge representation. The verification knowledge-base is used to determine the measurement procedures: to select an appropriate measuring instrument to determine how to obtain the features from the real surfaces, to calculate the measured parameter value and to compare the measured value with the tolerance value. A complete measurement procedure includes choosing an appropriate measuring instrument, partition of the measured surface, extraction of a finite number of data points, filtration of the profile, the parameters to be calculated and choosing a suitable comparison rule. The details are described below:

At first, a set of surface texture specifications is given with a manufactured product, which includes the parameters and their values. The instruments chosen need to match instrument attributes with measuring requirements. The Amplitude-Wavelength Diagram can be applied as part of the selection of instruments.

Before carrying out the measurement, the instrument needs calibration. Surface texture measuring instruments are calibrated using secondary calibration artefacts. Each of the calibration artefact types has a limited range of application according to its own characteristics and those of the instrument to be calibrated. Calibration of an instrument should always choose those task related instrument metrological characteristics which are relevant for the intended measurements for calibration and also match calibration artefacts with instrument metrological characteristics to be measured. For example, for the measurement of roughness parameter Ra, the vertical profile components need to be calibrated [1-3].

After calibration, the measurement procedure starts. The probe traverses the surface, and records the traced profile. Traverse length depends on the specification of the surface; it is usually greater than the evaluation length. Usually at least ten measurements should be made. The measurement area should be selected to be typical of the surface [1]. The nominal form of the surface to be measured is assumed to have already been removed. Filtering is the procedure that enables the user to separate certain spatial frequency components of the surface profile. A filter is an electronic, mechanical, optical or mathematical transformation of a profile to attenuate wavelength components of the surface outside the range of interest of the user [1]. There are three profile filters normally used in the measurement procedures, λ_s profile filter, which defines where the intersection occurs between the roughness and shorter wavelength components present in a surface; λ_c profile filter, which defines where the intersection occurs between the roughness and waviness components and λ_f profile filter, which defines where the intersection occurs between the waviness and longer wavelength components present in a surface. After filtering, the measured value can be calculated.

The final step is to make a comparison of measured values with requirements specified in the specifications. The default comparison rule is the 16% rule, which means for requirements specifying an upper limit of a parameter, the surface is considered acceptable if not more than 16% of all the measured values, based on the evaluation length, exceed the value specified on the specification. Another comparison rule is the max-rule, here the requirements specify a maximum value of the parameter, none of the measured values of the parameter over the entire surface can exceed the value specified.

6.2 Knowledge acquisition

The knowledge of verification includes five parts: partition of the measured surface, extraction of a finite number of data points, filtration of the profile, the parameters to be calculated and choosing an appropriate measuring instrument. The sources are mainly coming from ISO 1302 [4] and ISO 3274 [5].

6.2.1 Partition

The feature operation partition is used to identify the bounded surface which is to be characterized. Since the surface texture is influenced by the detailed form of the profile curve, the feature information needed for carrying out the partition operation includes the traverse length of the surface profile being evaluated and the traverse direction of the measurement instrument.

6.2.1.1 Traverse length

Traverse length is the length of surface traversed by the measurement instrument (e.g.

stylus) during a measurement [1]. It is usually greater than the evaluation length ln, see figure 6.1.

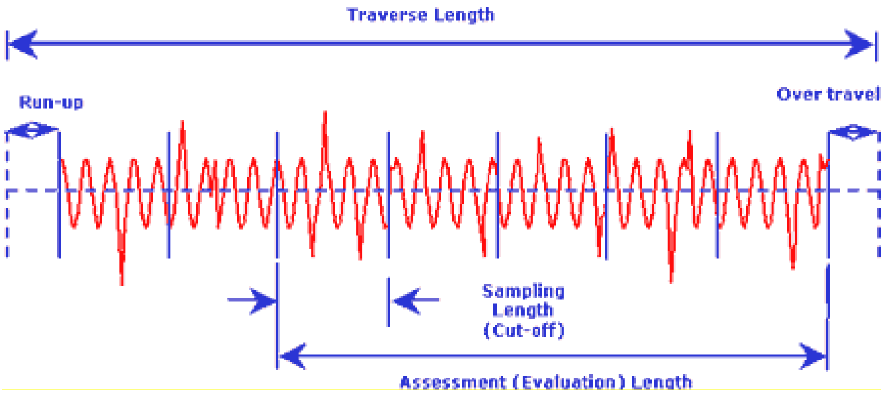


Figure 6.1 Example of traverse length and evaluation length [6]

6.2.1.2 Traverse direction

The traverse direction [1] is the direction traced by the measurement instrument (e.g. stylus) during a measurement, which should be perpendicular to the direction of the lay unless otherwise indicated. Take the surface texture lay '═' for example, which means the lay direction is parallel to plane of projection of view in which callout symbol is used, then the traverse direction should be perpendicular to plane of projection of view in which symbol is used, see figure 6.2. The different kinds of surface lay directions are introduced in section 5.2.1.1, which belongs to the partition operation of specification knowledge-base.

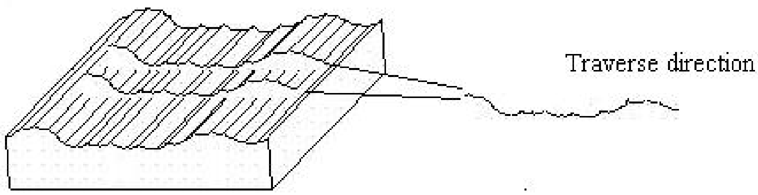


Figure 6.2 Traverse direction

After capturing the knowledge above, the refinement process of Partition is shown in figure 6.3. The abstract entity Partition in the simplified model of Verification is refined to the concrete data structures, which includes objects "traverse length", and "traverse direction".

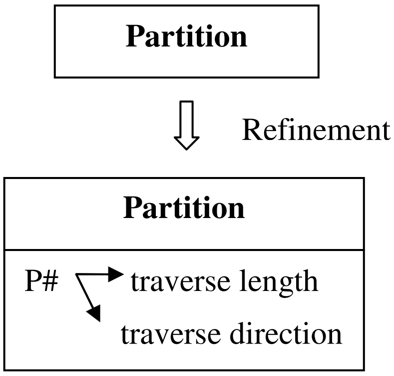


Figure 6.3 Refinement process of Partition

6.2.2 Extraction

The feature operation extraction is used to identify a finite number of points from the surface. The necessary information for carrying out the extraction includes the primary profile lower limit, the sampling spacing and the sampling length.

6.2.2.1 Primary profile lower limit λs

The short wavelength filter with cut-off λs [1] is used to extract the primary profile for the evaluation of primary profile parameters. Normally, the value of the lower limit λs is indicated as the short-wave cut-off value in the transmission band in the callout symbol for both roughness and primary profile parameters, see figure 5.1. For example "0,0025-" indicates a short-wave cut-off value of 0.0025 millimetres, which will allow wavelengths above 0.0025mm to be assessed with wavelengths below 0.0025mm being reduced in amplitude. Where no transmission band is indicated in connection with the parameter designation, the default transmission band applies to the surface texture requirement, see section 6.2.3.2.

6.2.2.2 Sampling spacing

The sampling spacing [5] is the width length between two adjacent measuring points on the surface, which can be obtained from ISO 3274 [5] according to the value of upper limit λc or lower limit λs. The relationship between λc, λs and the sampling spacing is shown in table 6.1.

λc (mm)	λs (μm)	Maximum sampling spacing (μm)
0,08	2,5	0,5
0,25	2,5	0,5
0,8	2,5	0,5
2,5	8	1,5
8	25	6

Table 6.1 Relationship between the roughness cut-off wavelength λc and maximum sampling spacing [5]

6.2.2.3 Sampling length lp, lr, lw

Sampling length [4] is the length in the direction of the X-axis used for identifying the irregularities characterizing the profile under evaluation [7], see figure 5.8. Sampling length concept is applied to profile parameters [4] and parameters based on material ratio curve [8&9]. Motif parameters [10] do not use the concept of sampling length. The operator used to calculate motif parameters has its own limit values and sampling length concepts do not exist.

- lp is the sampling length for the *P*-profile (primary profile), which is equal to the evaluation length ln.
- lr is the sampling length for the *R*-profile (roughness profile), which is numerically equal to the wavelength of the profile filter λc.
- lw is the sampling length for the *W*-profile (waviness profile), which is numerically equal to the wavelength of the profile filter λf.

The sampling length in the Extraction of the verification knowledge-base mirrors the sampling length in the Extraction of the specification knowledge-base. The details will be explained in section 6.3.3.

After capturing the knowledge above, the refinement process of Extraction is shown in figure 6.4. The abstract entity Extraction in the simplified model of Verification is refined to the concrete data structures, which includes objects the primary profile lower limit λs, the sampling spacing and the sampling length lp, lr and lw, and the relationship between the λs and the sampling spacing. Since the traverse length should be greater than the evaluation length ln, arrow ① represents this relationship, (see section 6.2.6 for details).

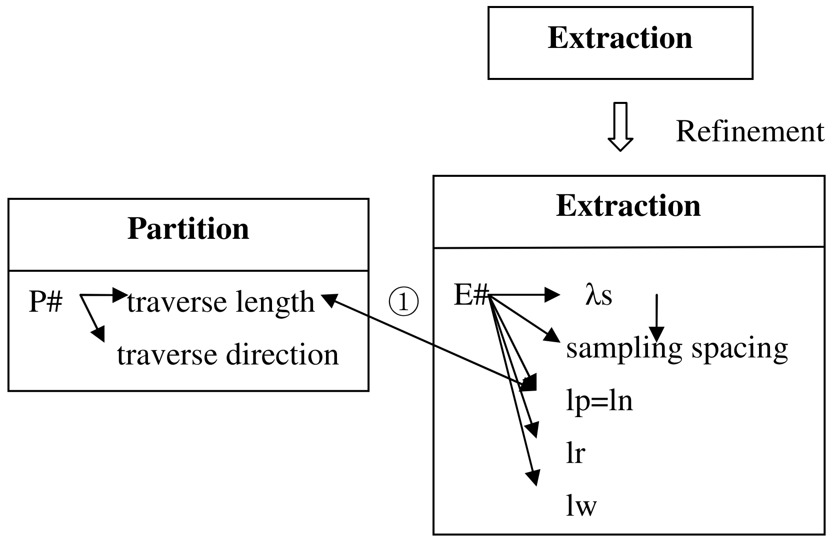


Figure 6.4 Refinement process of Extraction

6.2.3 Filtration

The feature operation filtration is used to separate the surface profile into roughness profile and waviness profile.

6.2.3.1 Filter type

The filter type may be indicated as “Gaussian” or “2RC”. This is not standardized, but an indication of filter name as proposed here is unambiguous. The filter type mirrors the filter type in the Filtration of specification knowledge-base. The details will be explained in section 6.3.3.

6.2.3.2 Cut-off wavelength

The cut-off wavelength is used as a means of separating or filtering the wavelengths of a surface. The value of cut-off wavelength mirrors the upper limit in the specification knowledge-base, the details will be explained in section 6.3.3.

1) Profile parameters

Table 6.2 lists the value of the cut-off wavelength for profile parameters [7].

Profile	**Cut-off wavelength**
R- profile (upper limit λc)	λc is the cut-off wavelength to separate the roughness profile from the waviness profile. λc is equal to the sampling length for the *R*-profile lr [4].
W- profile (upper limit λf)	λf is the cut-off wavelength to separate the waviness profile from the primary profile. λf is equal to the sampling length for the *W*-profile lw [4].

Table 6.2 Cut-off wavelength for profile parameters

2) Motif parameters

Two bounds A and B are used in the motif algorithms according to ISO 12085 [10], to define respectively the maximum widths of the roughness and waviness motifs. Table 6.3 lists the value of the cut-off wavelength for motif parameters.

Profile	Cut-off wavelength
R-profile (upper limit A)	The width for the roughness motif ARj should be less than or equal to the value A. Upper limit A mirrors the upper limit in the Filtration of the specification knowledge-base, which can be obtained from table 5.11 according to the evaluation length either indicated in the callout or referenced to the default value.
W-profile (upper limit B)	The width for the waviness motif AWj should be less than or equal to the value B. Upper limit B mirrors the upper limit in the Filtration of specification knowledge-base, which can also be obtained from table 5.11 according to the evaluation length either indicated in the callout or referenced to the default value.

Table 6.3 Cut-off wavelength for motif parameters

3) Parameters based on material ratio curve

For roughness profiles, the upper limit λc is defined equal to the sampling length lr according to ISO 1302 [4] and ISO 13565-1 [11]. The default sampling length for parameters based on material ratio curve is 0,8 millimetres, see section 6.2.2.2.

After capturing the knowledge above, the refinement process of Filtration is shown in figure 6.5. The abstract entity Filtration in the simplified model of Verification is refined to the concrete data structures, which includes objects "filter type", and "cut-off wavelength". The object cut-off wavelength in the entity Filtration also have relationships ② & ③ with objects sampling lengths lr and lw in the entity Extraction, (see section 6.2.6 for details).

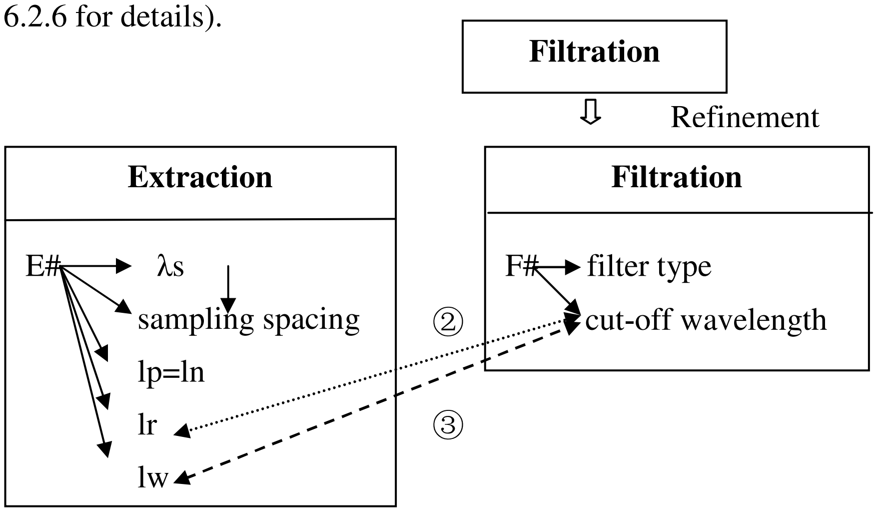


Figure 6.5 Refinement process of Filtration

6.2.4 Measured value

A measured value consists of the value obtained from the measurement of the indicated parameter, which will be compared with the measurand.

Parameter type

The detail designations of parameters are introduced in section 6.2.4.2. The parameter type mirrors the parameter type in Measurand of the specification knowledge-base.

Value

This is the value measured by an instrument and given in micrometers. The measured value will be compared with the measurand of specification to check whether the real surface conforms to the specification.

Number of measurements

Usually at least 10 measurements should be made as recommended in reference Richard Leach (2001) [1].

After capturing the knowledge above, the refinement process of Measured value is shown in figure 6.6. The abstract entity Measured Value in the simplified model of Verification is refined to the concrete data structures, which includes objects "parameter type", "value" and "number of measurements".

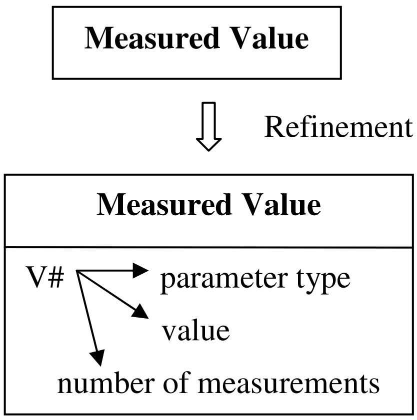


Figure 6.6 Refinement process of Measured value

6.2.5 Instrument

6.2.5.1 Instrument selection

Measurement of surface topography plays an important role in manufacturing, being used for both the control of manufacturing processes and for final product acceptance. These measurements can be performed with a variety of instruments which have different capabilities and limitations [12]. Nowadays there are three principal measuring methods: Stylus methods, Optical methods and other (non-optical) methods [13].

A typical instrument includes: the probe, the transducer and reference, the amplifier, filter or processor, recorder or other output such as a meter or computer, see figure 6.7.

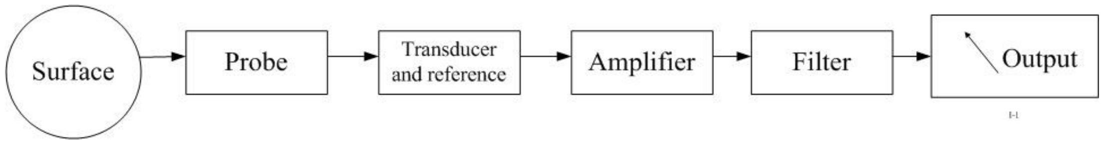


Figure 6.7 Illustration of a typical instrument

- The stylus methods use the stylus as the central component of the probe. It is a contact method. It examines heights relative to the reference line in the plane perpendicular to the surface [13].
- Some optical methods use an optical probe and involve projecting light on to a surface. They are all non-contact methods, which look for lateral structure, namely spacings and detail in the plane of the surface [13].
- Other instruments include the new generation of scanning microscopes such as the Scanning Electron Microscope (SEM), the Scanning Tunnelling Microscope (STM) and the Atomic Force Microscope (AFM) [13]. They use the scanning probe and involve electrons, tunnelling current and atomic force respectively rather than light.

The instruments chosen need to match instrument attributes with measuring requirements. When selecting the instrument, the measurement range and resolution of different instruments are the most important factors to consider. The following A-W diagram in figure 6.8 shows an amplitude-wavelength plot of some instruments. It can be applied as part of the selection of instruments. In the figure, the two axes represent the resolution and the range of the instruments both in vertical and horizontal directions.

According to the A-W diagram, the STM/AFM measuring systems have the highest resolution in both directions; however, the measurement range is small. The stylus instrument has a large vertical range, although it can have the highest vertical resolution to sub-nanometer, it is best suitable for measuring engineering surfaces at micron or sub-micron scale. The focus detection instrument has a slightly inferior resolution to the stylus instrument in both the vertical and horizontal directions. The interferometer has the highest resolution (to sub-nanometer level) in the vertical direction, but the horizontal resolution is not comparable with this [14].

It is possible to locate the target parameter value on the plot and find out what kind of instrument matches the requirements, see symbol ◆ in figure 6.8, which represents the target parameter value Ra 3,3. The x axis can be located by the sampling spacing, which is listed in table 6.1, and the y axis can be located by the parameter value. For this example, Ra 3,3, the sampling space is 1.5μm and the parameter value is 3.3μm, thus the target dot is located in the position (1.5μm, 3.3μm). From the A-W diagram in figure 6.8, there are three instruments that may be capable of carrying out the measurement: Stylus, Focus and SEM detection.

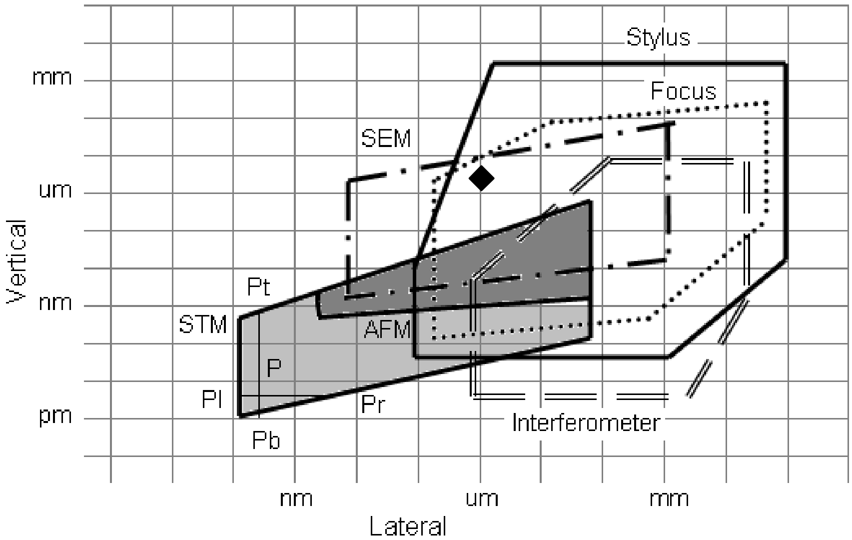


Figure 6.8 Instrumentation: Amplitude-Wavelength Diagram

6.2.5.2 Characteristics of the instruments

Table 6.4 lists major characteristics of typical instruments [13], including measurement tool, spatial resolution, spatial range, vertical resolution (z resolution), vertical range (z range), measurement frequency and comments.

Method	Measurement tool	Spatial resolution	Spatial range	Z resolution	Range z	Frequency	Comments
Stylus	Stylus tip	0.1μm	100mm	0.3nm	1000μm	20Hz	Contacts workpiece
Focus	Optical probe	0.5μm	50mm	0.5nm	100μm		Non-contacting
Interferometer	Optical probe	1μm	10mm	0.01nm	10μm	minutes	Non-contacting
SEM	Detection	0.01μm	1mm	2nm	10μm	minutes	Vacuum needed
STM	Conductive probe	0.0001μm	0.1mm	0.001nm	0.1μm	minutes	Only for the conducting surfaces
AFM	Atom force tip	0.005μm	0.08mm	1nm	0.1μm	minutes	Both for conducting and non conducting surfaces

Table 6.4 The characteristics for typical methods [13]

After selecting the instruments using A-W plot, users can check the detailed characteristics of candidate instruments and choose the most suitable instrument for the measurement.

6.2.5.3 Inserting new instruments

The system allows users to insert new instruments into the knowledge-base. Users could add the new instruments with essential attributes: z resolution, z range, spatial resolution and spatial range. The system will generate a new polygon on the A-W plot for the new instrument according to attributes, and insert a new row in the characteristics table for it as well.

After capturing the knowledge above, the refinement process of Instrument is shown in figure 6.9. The abstract entity Instrument in the simplified model of Verification is refined to the concrete data structures, which includes objects "A-W plot", "characteristics", and a sub-entity Add Instruments with objects "z resolution", "z range", "spatial resolution" and "spatial range".

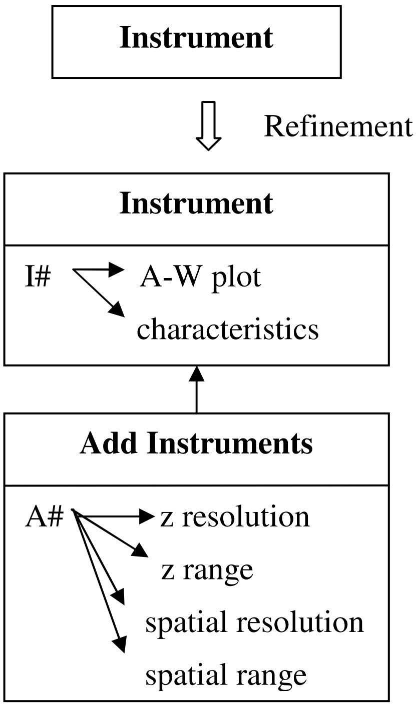


Figure 6.9 Refinement process of Instrument

6.2.6 Relations in the verification knowledge-base

The relationships refined in the concrete model of Verification are shown in figure 6.4, figure 6.5 and figure 6.10, and will be illustrated below:

Relationship ① — greater: Relationship between entities Partition and Extraction

The traverse length in Partition should be greater than the evaluation length ln (lp) in Extraction, see section 6.2.1.1.

i.e. traverse length > evaluation length ln (lp)

Relationship ② — equal: Relationship between entities Extraction and Filtration

The cut-off wavelength λc in Filtration equals the sampling length lr in Extraction, see section 6.2.3.2. For example take the surface parameter Ra with a limit value of 3.3 micrometers, the sampling length lr mirrors the sampling length in Specification, and shall be the default value of 2.5 millimetres, see section 6.2.2.2; and as the cut-off wavelength λc equals the sampling length lr, it is equal to 2.5 millimetres.

i.e. cut-off wavelength λc = Sampling length lr

Relationship ③ — equals: Relationship between entities Extraction and Filtration

The cut-off wavelength λf in Filtration equals the sampling length lw in Extraction, see section 6.2.3.2.

i.e. cut-off wavelength λf = Sampling length lw

Relationship ④ — determine: Relationship among entities Extraction, Instrument and Measurand (see figure 6.10)

The sampling spacing and the limit value of measurand help to select the suitable instrument on A-W plot, see section 6.2.5.1. The symbol on the A-W plot representing the target parameter is located by the sampling spacing at x axis, and the parameter value at y axis. The polygons surrounding the symbol represent instruments satisfying requirements.

Relationship ⑤ — compares: Relationship among entities Measured value, Measurand and Comparison rule (see figure 6.10)

The measured value is compared with the measurand value by comparison rules, see section 6.2.5. There are two types of comparison rules, “16 %-rule” and “max-rule”, which will be indicated on the callout.

6.3 Knowledge representation

Following the refinement processes described above, the category model of verification knowledge-base is developed. In this model, relations among structures are represented by the pullback structure.

6.3.1 Category Model of Verification

The verification model comprises five entities: partition, extraction, filtration, instrument and measured value. The detailed knowledge contained in each part are already specified in previous sections. The model below in figure 6.10 represents the verification knowledge-base, based on category theory.

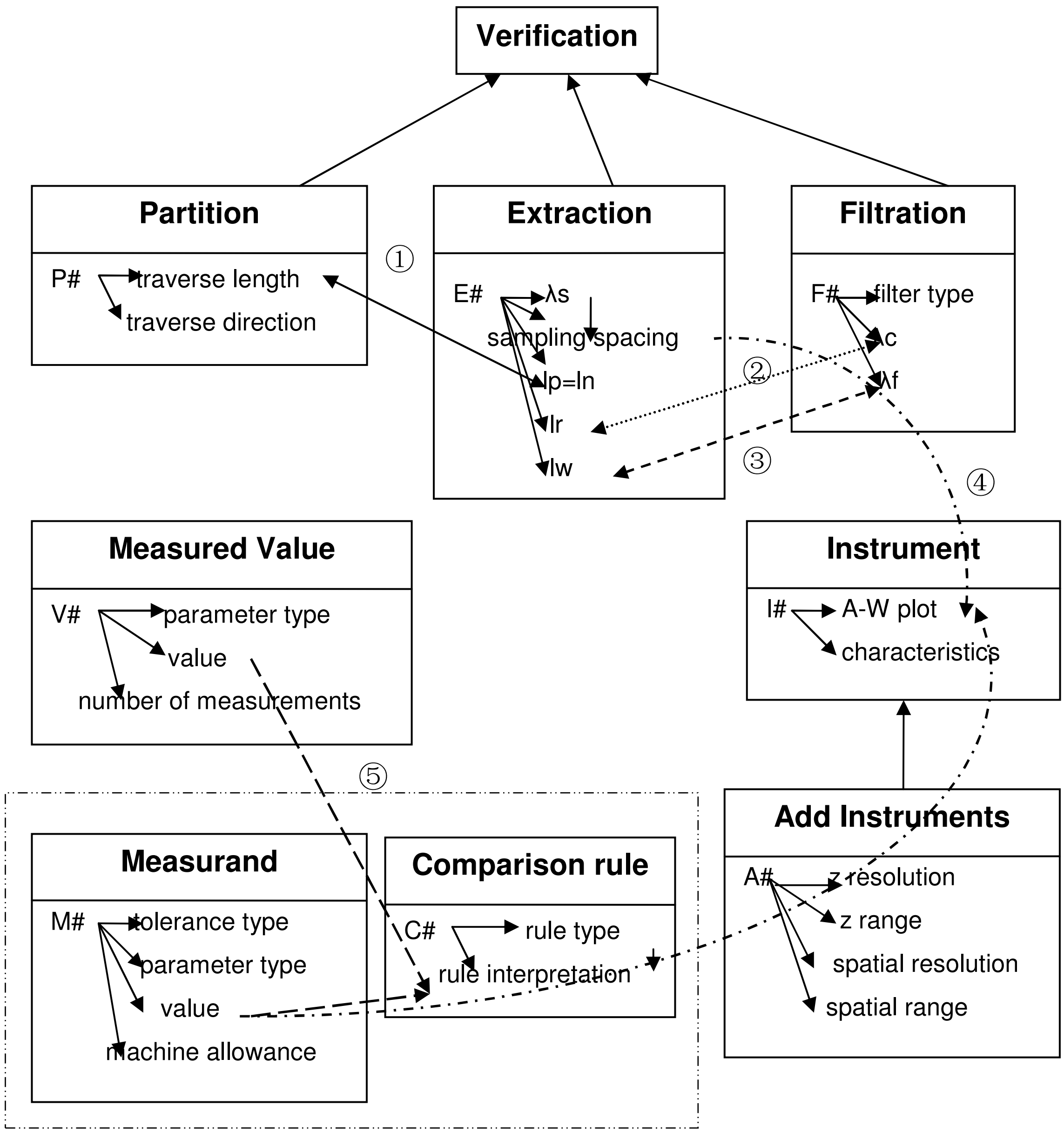


Figure 6.10 Verification model

Each part is represented as a category with attributes in it. The arrows inside categories represent the intra-object relations. ①, ②, ③, ④ and ⑤ are the simplified pullback structures used to represent the relationships between objects. Entities Measurand and Comparison rule belong to specification knowledge-base.

6.3.2 Relationships in pullback structure

① greater – relationship between entities Partition and Extraction, see figure 6.11:

Here the relationship is that the traverse length in Partition should be greater than the evaluation length ln (lp) in Extraction.

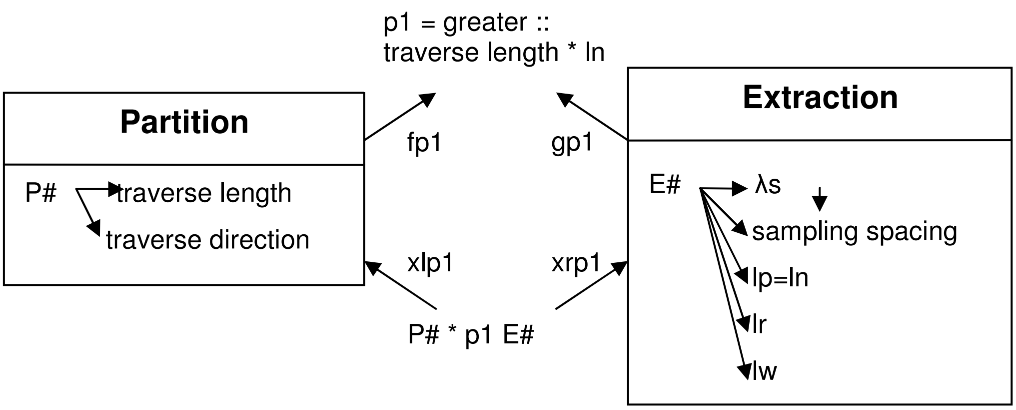


Figure 6.11 Relation "greater" in pullback structure

② equal – relationship between entities Extraction and Filtration, see figure 6.12:

Here the relationship is that the cut-off wavelength λc in Filtration equals the sampling length lr in Extraction.

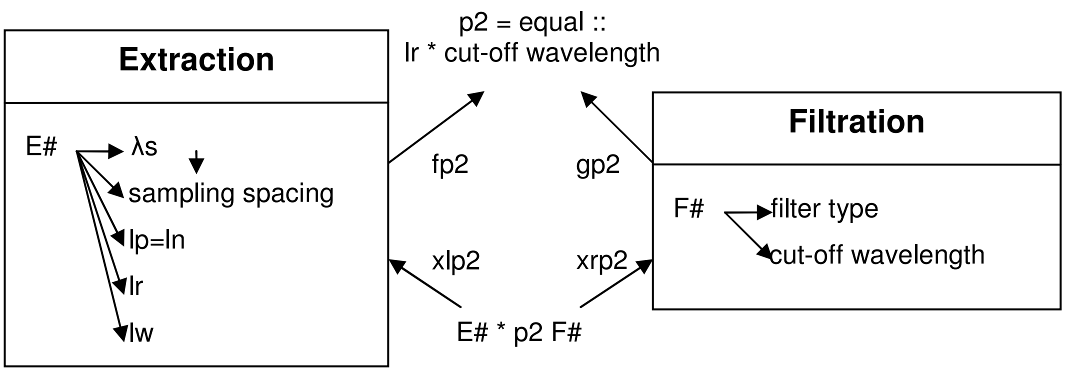


Figure 6.12 Relation "equal" in pullback structure

③ equals – relationship between entities Extraction and Filtration, see figure 6.13:

Here the relationship is that the cut-off wavelength λf in Filtration equals the sampling length lw in Extraction.

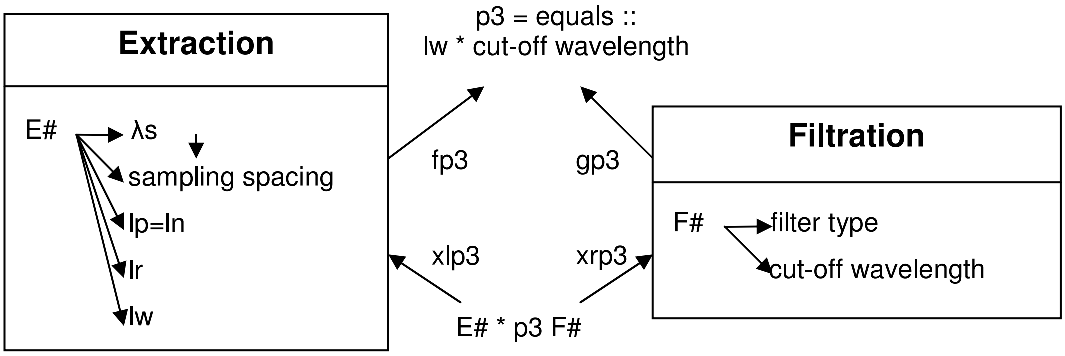


Figure 6.13 Relation "equals" in pullback structure

④ determine – relationship among entities Extraction, Instrument and Measurand, see figure 6.14:

Here the relationship is that the sampling spacing and the limit value of measurand help to select the suitable instrument on A-W plot.

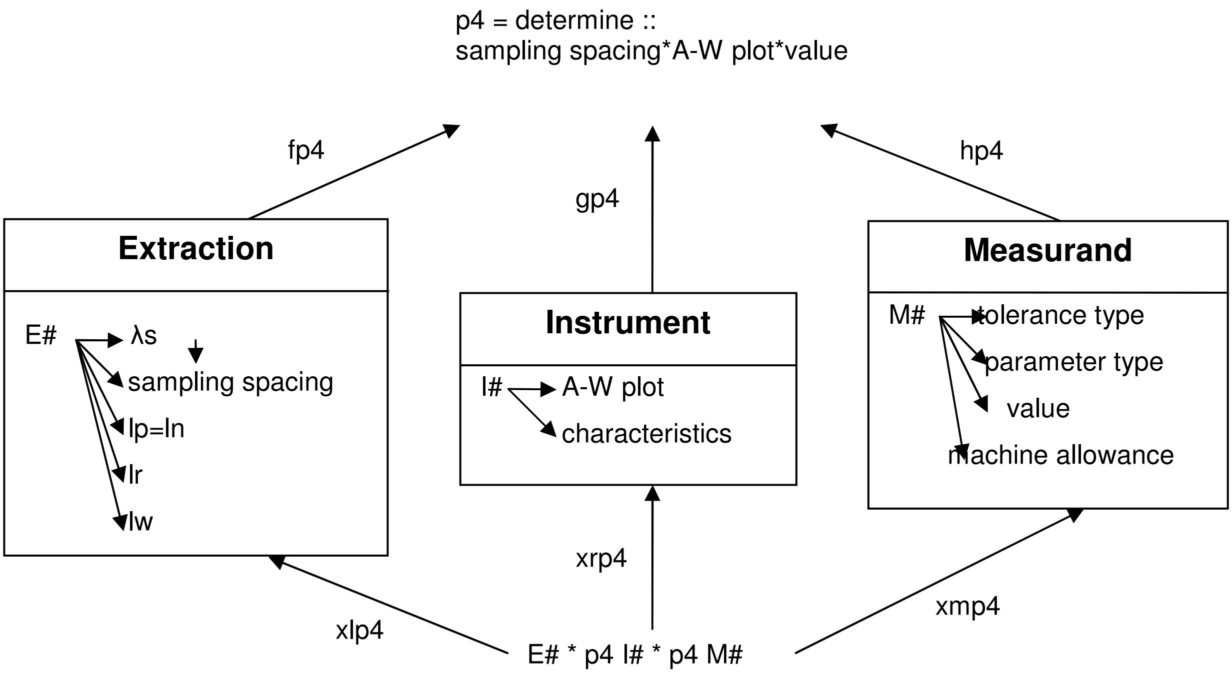


Figure 6.14 Relation "determine" in pullback structure

In the A-W plot, each polygon represents the measuring range and resolution of one instrument and is bounded by several line segments. Each single line segment of the polygon has a slope. The sampling spacing and value are a pair of co-ordinates which specifies the location of the target for the surface texture parameter in the plot. In order to find out the polygons which are surrounding the symbol, the comparison of coordinates between the symbol and polygons is needed.

The concept '**partial pre-order**' [15] in category theory can be used to carry out the comparison, see section 3.2.2, i.e. if $x \le p$, and $p \le y$, then implies $x \le y$. The detailed procedures are explained below:

- Select one line segment of a polygon as a reference
- Imagine a parallel line which goes through the coordinates of the target and has the same slope as the reference line
- Compare and record the position of the parallel line and the reference line
- Repeat the above procedures on the rest of line segments of the polygon
- Obtain the result whether the coordinate of the target locates inside or outside the polygon

For example, see figure 6.15: P (s,v) is the coordinate representing the target, s represents the sampling spacing and v represents the vertical limit value; G is the polygon representing the range and resolution of a specific instrument, bounded by four lines: a, b, c and d.

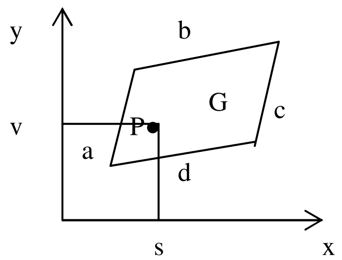


Figure 6.15 An example to illustrate the compare procedures

For line a: $y = m1x+n1$

when $y=v$, $x=s1$

compare s and s1, if $s >= s1$, then 'P >= a', else 'P < a'

For line b: $y = m2x+n2$

when $x=s$, $y=v1$

compare v and v1, if $v > v1$, then 'P > b', else 'P <= b'

For line c: $y = m3x+n3$

when $y=v$, $x=s2$

compare s and s2, if $s > s2$, then 'P > c', else 'P <= c'

For line d: $y = m4x+n4$

when $x=s$, $y=v2$

compare v and v2, if v >= v2, then ‘P >= d’, else ‘P < d’

note: [m1, m2, m3, m4, n1, n2, n3, n4, v, s, v1,s1,v2, s2] R

Result: If ‘P >=a’ and ‘P <= b’ and ‘P <= c’ and ‘P >=d’, then ‘P is in G’ (P is inside the polygon G)

The lines to describe the polygons in A-W plot are shown below together with the comparison procedures (The unit is μm) [16-18]:

STM:

See figure 6.16 for the polygon of a STM, line T1 represents the lower lateral limit (horizontal resolution) of a STM and given by a vertical line, then according to clockwise direction, T2 represents the minimum radius of curvature a STM probe can be sensed and given by a slope line, T3 represents the upper lateral limit (horizontal range) of a STM and given by another vertical line, and T4 represents the maximum slope a STM probe can be sensed and given by a slope line at the bottom.

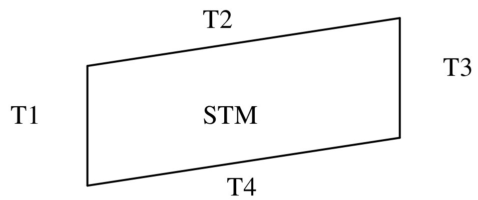


Figure 6.16 Polygon of STM

Line T1: $x=10^{-4}$

compare s and $s1=10^{-4}$, if s >= s1, then ‘P >= T1’, else ‘P < T1’

Line T2: $y=10^{-1}\times x+10^{-3}$

when x=s, y=v1

compare v and v1, if v > v1, then ‘P > T2’, else ‘P <= T2’

Line T3: $x=10^{2}$

compare s and $s2=10^{2}$, if s > s2, then ‘P > T3’, else ‘P <= T3’

Line T4: $y=10^{-5}\times x+10^{-6}$

when x=s, y=v2

compare v and v2, if v >= v2, then ‘P >= T4’, else ‘P < T4’

Result: If ‘P >=T1’ and ‘P <= T2’ and ‘P <= T3’ and ‘P >=T4’, then the target P is in the bounded polygon for a STM.

Therefore, the STM is able to measure the parameter P.

AFM:

See figure 6.17 for the polygon of an AFM, line A1 represents the lower lateral limit (horizontal resolution) of an AFM and given by a vertical line, then according to clockwise direction, A2 represents the minimum radius of curvature an AFM probe can be sensed and given by a slope line, A3 represents the upper lateral limit (horizontal range) of an AFM and given by another vertical line, and A4 represents the maximum slope an AFM probe can be sensed and given by a slope line at the bottom.

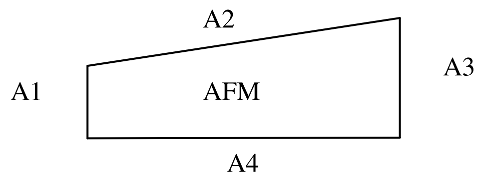


Figure 6.17 Polygon of AFM

Line A1: $x=5\times10^{-3}$

compare s and $s3=5\times10^{-3}$, if s >= s3, then 'P >= A1', else 'P < A1'

Line A2: $y=10^{-1}\times x+10^{-3}$

when x=s, y=v3

compare v and v3, if v > v3, then 'P > A2', else 'P <= A2'

Line A3: $x=10^{2}$

compare s and $s4=10^{2}$, if s > s4, then 'P > A3', else 'P <= A3'

Line A4: $y=10^{-3}$

compare v and $v4=10^{-3}$, if v >= v4, then 'P >= A4', else 'P < A4'

Result: If 'P >=A1' and 'P <= A2' and 'P <= A3' and 'P >=A4', then the target P is in the bounded polygon for an AFM.

Therefore, the AFM is able to measure the parameter P.

SEM:

See figure 6.18 for the polygon of a SEM, line E1 represents the lower lateral limit (horizontal resolution) of a SEM and given by a vertical line, then according to clockwise direction, E2 represents the minimum radius of curvature a SEM probe can be sensed and given by a slope line, E3 represents the upper lateral limit (horizontal range) of a SEM and given by another vertical line, and E4 represents the maximum slope a SEM probe can be sensed and given by a slope line at the bottom.

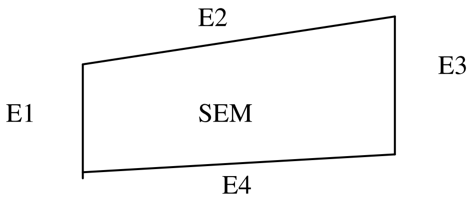


Figure 6.18 Polygon of SEM

Line E1: $x=10^{-2}$

compare s and $s5=10^{-2}$, if s >= s5, then 'P >= E1', else 'P < E1'

Line E2: $y=10^{-1}\times x+10$

when x=s, y=v5

compare v and v5, if v > v5, then 'P > E2', else 'P <= E2'

Line E3: $x=10^{3}$

compare s and $s6=10^{3}$, if s > s6, then 'P > E3', else 'P <= E3'

Line E4: $y=10^{-4}\times x+10^{-3}$

when x=s, y=v6

compare v and v6, if v >= v6, then 'P >= E4', else 'P < E4'

Result: If 'P >= E1' and 'P <= E2' and 'P <= E3' and 'P >= E4', then the target P is in the bounded polygon for a SEM.

Therefore, the SEM is able to measure the parameter P.

Stylus instrument:

See figure 6.19 for the polygon of a Stylus instrument, line L1 represents the lower lateral limit (horizontal resolution) of a Stylus instrument and given by a vertical line, then according to clockwise direction, L2 represents the minimum radius of curvature a Stylus tip can be reached and given by a slope line, L3 represents the upper vertical limit (vertical range) of a Stylus instrument and given by a horizontal line at the top, L4 represents the upper lateral limit (horizontal range) of a Stylus instrument and given by another vertical line, L5 represents the maximum slope a Stylus tip can be reached and given by a slope, and L6 represents the lower vertical limit (vertical resolution) of a Stylus instrument and given by a horizontal line at the bottom.

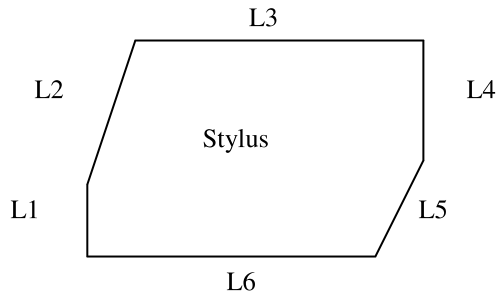


Figure 6.19 Polygon of Stylus instrument

Line L1: $x=10^{-1}$

compare s and $s7=10^{-1}$, if s >= s7, then 'P >= L1', else 'P < L1'

Line L2: $y=10^3\times x+10^{-2}-10^2$

when y=v, x=s8

compare s and s8, if s >= s8, then 'P >= L2', else 'P < L2'

Line L3: $y=5\times10^3$

compare v and $v7=5\times10^3$, if v >v7, then 'P > L3', else 'P <= L3'

Line L4: $x=10^5$

compare s and $s9=10^5$, if s >s9, then 'P > L4', else 'P <= L4'

Line L5: $y=10^{-7}\times x$

when y=v, x=s10

compare s and s10, if s >s10, then 'P > L5', else 'P <= L5'

Line L6: $y=2\times10^{-4}$

compare v and v8, if v >= v8, then 'P >= L6', else 'P < L6'

Result: If 'P >= L1' and 'P >= L2' and 'P <= L3' and 'P <= L4' and 'P <= L5' and 'P >=L6', then the target P is in the bounded polygon for a Stylus.

Therefore, the Stylus is able to measure the parameter P.

Focus detection instrument:

See figure 6.20 for the polygon of a Focus detection instrument, line F1 represents the lower lateral limit (horizontal resolution) of a Focus detection instrument and given by a vertical line, then according to clockwise direction, F2 represents the minimum radius of curvature a probe can be sensed and given by a slope line, F3 represents the upper vertical limit (vertical range) of a Focus detection instrument and given by a horizontal line at the top, F4 represents the upper lateral limit (horizontal range) and given by another vertical line, F5 represents the maximum slope a probe can be sensed and given by a slope, and F6 represents the lower vertical limit (vertical resolution) of a Focus detection instrument and given by a slope line at the bottom.

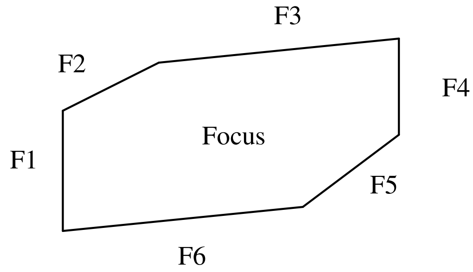


Figure 6.20 Polygon of Focus detection instrument

Line F1: $x=2\times10^{-1}$

compare s and $s11=2\times10^{-1}$, if s >= s11, then 'P >= F1', else 'P < F1'

Line F2: $y=10\times x$

when y=v, x=s12

compare s and s12, if s >= s12, then 'P >= F2', else 'P < F2'

Line F3: $y=10^2$

compare v and $v9=10^2$, if v >v9, then 'P > F3', else 'P <= F3'

Line F4: $x=6\times10^4$

compare s and $s13=6\times10^4$, if s >s13, then 'P > F4', else 'P <= F4'

Line F5: $y=2\times10^{-6}\times x-10^{-3}$

when y=v, x=s14

compare s and s14, if s >s14, then 'P > F5', else 'P <= F5'

Line F6: $y=10^{-6}\times x+10^{-4}$

when x=s, y=v10

compare v and v10, if v >= v10, then 'P >= F6', else 'P < F6'

Result: If 'P >= F1' and 'P >= F2' and 'P <= F3' and 'P <= F4' and 'P <= F5' and 'P >=F6', then the target P is in the bounded polygon for a Focus.

Therefore, the Focus is able to measure the parameter P.

The Interferometer:

See figure 6.21 for the polygon of an Interferometer, line I1 represents the lower lateral limit (horizontal resolution) of an Interferometer and given by a vertical line, then according to clockwise direction, line I2 the minimum radius of curvature a probe can be sensed and given by a slope line, I3 represents the upper vertical limit (vertical range) of an Interferometer and given by a horizontal line at the top, I4 the upper lateral limit (horizontal range) of an Interferometer and given by another vertical line, I5 represents the maximum slope a probe can be sensed and given by a slope, and I6 represents the lower vertical limit (vertical resolution) of an Interferometer and given by a horizontal line at the bottom.

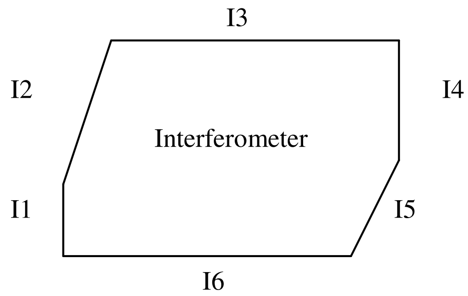


Figure 6.21 Polygon of Interferometer

Line I1: x=1

compare s and s15=1, if s >= s15, then 'P >= I1', else 'P < I1'

Line I2: $y=10^{-1}\times x+10^{-2}-10^{-1}$

when y=v, x=s16

compare s and s16, if s >= s16, then 'P >= I2', else 'P < I2'

Line I3: y=10

compare v and v11=10, if v >v11, then 'P > I3', else 'P <= I3'

Line I4: $x=2\times10^{4}$

compare s and $s17=2\times10^{4}$, if s >s17, then 'P > I4', else 'P <= I4'

Line I5: $y=10^{-7}\times x+10^{-4}-10^{-3}$

when y=v, x=s18

compare s and s18, if s >s18, then 'P > I5', else 'P <= I5'

Line I6: $y=8\times10^{-6}$

compare v and v12, if v >= v12, then 'P >= I6', else 'P < I6'

Result: If 'P >= I1' and 'P >= I2' and 'P <= I3' and 'P <= I4' and 'P <= I5' and 'P >=I6', then the target P is in the bounded polygon for an Interferometer.

Therefore, the Interferometer is able to measure the parameter P.

⑤ compares – relationship among entities Measured value, Measurand and Comparison rule, see figure 6.22:

Here the relationship is that the measured value is compared with the measurand value by comparison rules.

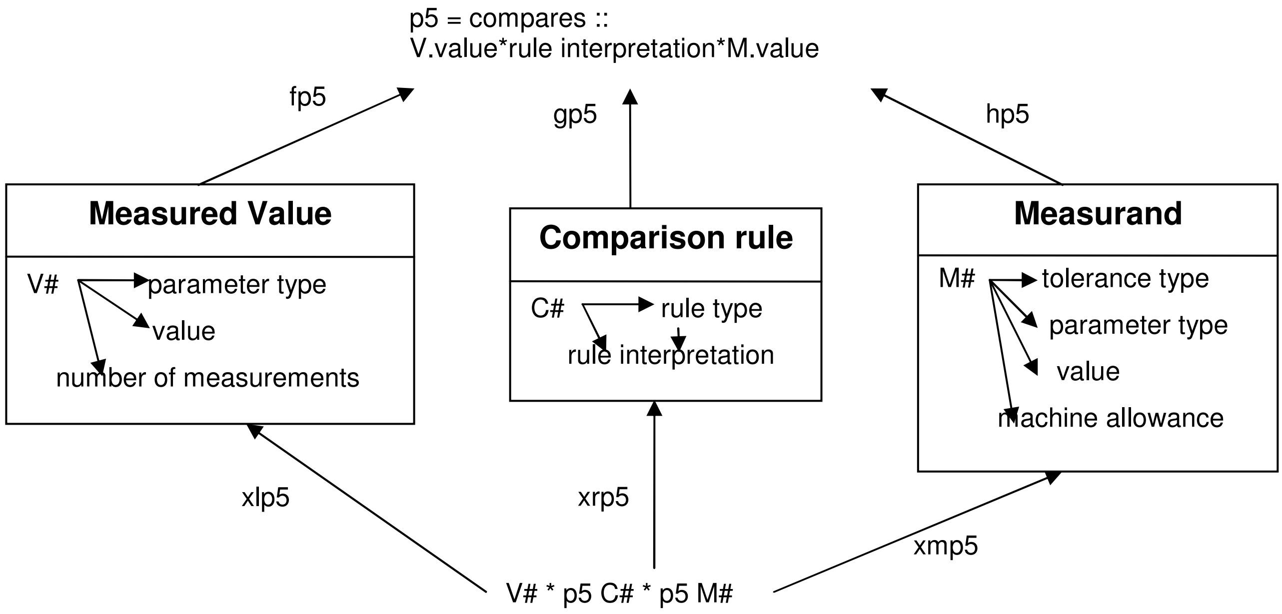


Figure 6.22 relation “compares” in pullback structure

6.3.3 Mirror relationships between the specification and verification knowledge-base

Chapter 5 mentioned that the verification knowledge-base mirrors the specification knowledge-base. The flowchart of the implementation of two knowledge-bases is shown in figure 6.24.

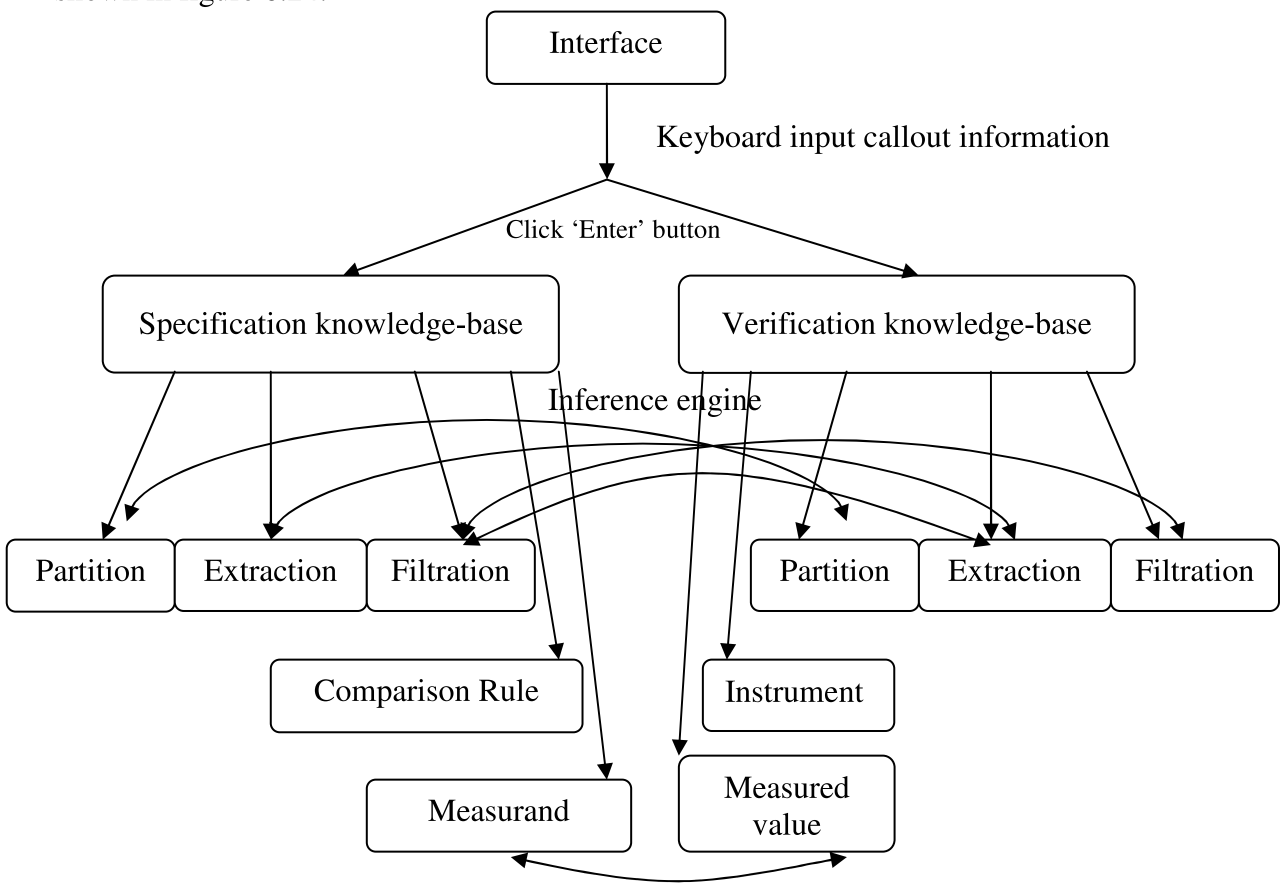
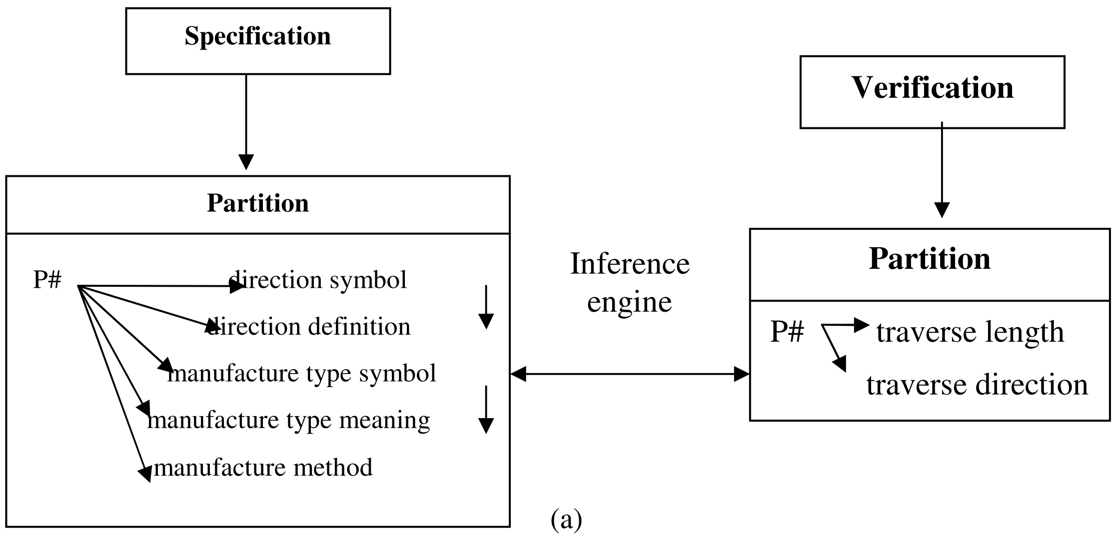


Figure 6.23 Flowchart of implementation

The detailed mirror relationships between the specification and verification knowledge-base are described in figure 6.24 (a) to (e) below:

Partition in Verification mirrors Partition in Specification:

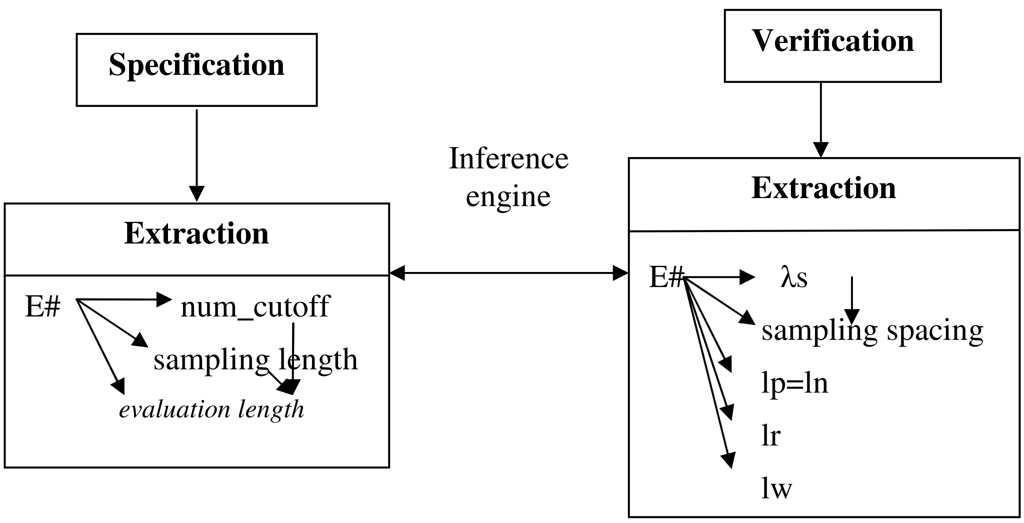
The traverse direction in Verification is perpendicular to the surface texture lay direction in the Specification, see figure (a) and section 6.2.1.2.



(a)

Extraction in Verification mirrors Extraction in Specification:

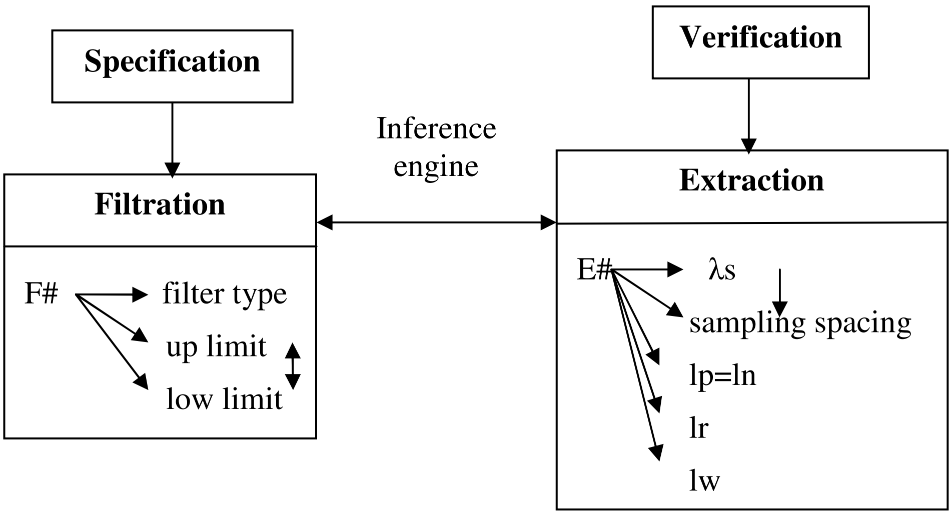
The evalution length ln in the extraction of verification equals the evaluation length in the extraction of specification. The sampling length lr, lw, and lp in the extraction of verification equals the sampling length in the extraction of specification, see figure (b) and section 6.2.2.3.



(b)

Extraction in Verification mirrors Filtration in Specification:

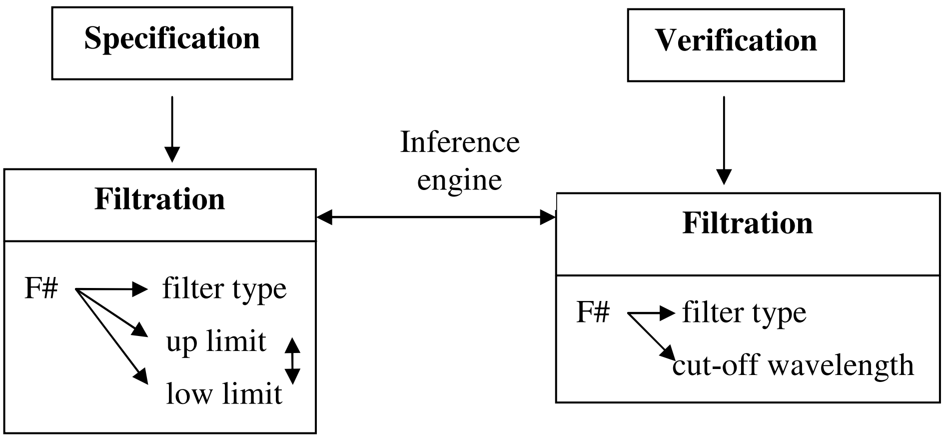
The lower limit λs in the extraction of verification equals the low limit λs in the filtration of specification, see figure (c) and section 6.2.2.1.



(c)

Filtration in Verification mirrors Filtration in Specification:

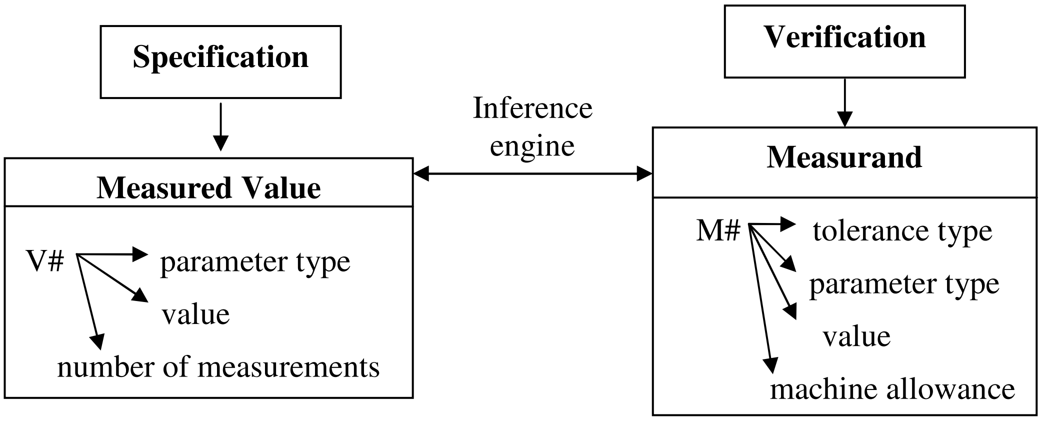
The filter type in the filtration of verification is the same as the filter type in the filtration of specification. The cut-off wavelength λc and λf in the filtration of verification equals the up limit in the filtration of specification, see figure (d) and section 6.2.3.2.



(d)

Measurand in Verification mirrors Measured value in Specification:

The parameter type of the measurand is the same as the parameter type of the measured value, see figure (e) and section 6.2.4.1.



(e)

Figure 6.24 The relationships between specification and verification

6.3.4 Functions of the verification knowledge-based system

The functions of the verification knowledge-based system are listed below:

1) Review the measurement processes definitions and contents. Users can obtain a detailed explanation of each procedure within the measurement processes, such as the definition of traverse length, the sampling spacing, etc.
2) Retrieve a suitable measurement process, including traverse length, traverse direction, sampling length, cut-off wavelength of filters, filter type, instrument and comparison rule. The system will suggest to the user the most suitable processes by matching records stored in the knowledge-base with measurement requirements.
3) Check detailed characteristics of candidate instruments. Users can carry out further comparison of suggested instruments and make a final decision.
4) Provide the comparison result after inputting both the measurand and the measured value. The system can calculate the result by using certain comparison rules and determine whether the surface is within the specified tolerance.
5) Further refer to the function, the specification and the manufacture knowledge-base. The verification knowledge-based system is connected with the others. Users can easily traverse through them.

6.4 Interface of the verification knowledge-base

The sample interface of verification knowledge-based system is shown in figure 6.25, which is an output for the callout ‘Ra 3,3’. The retrieved information includes the traverse length, the traverse direction, the filter type, the lower and upper limit, the sampling length, the sampling spacing, and the most important, suggested instruments. If users want to compare the suggested instruments in detail, they can simply click the hyperlink of 'Suggested Instruments'. The returning result would be a table that lists all the characteristics of each instrument. And users can also add new instruments just clicking the hyperlink of "Add instruments".

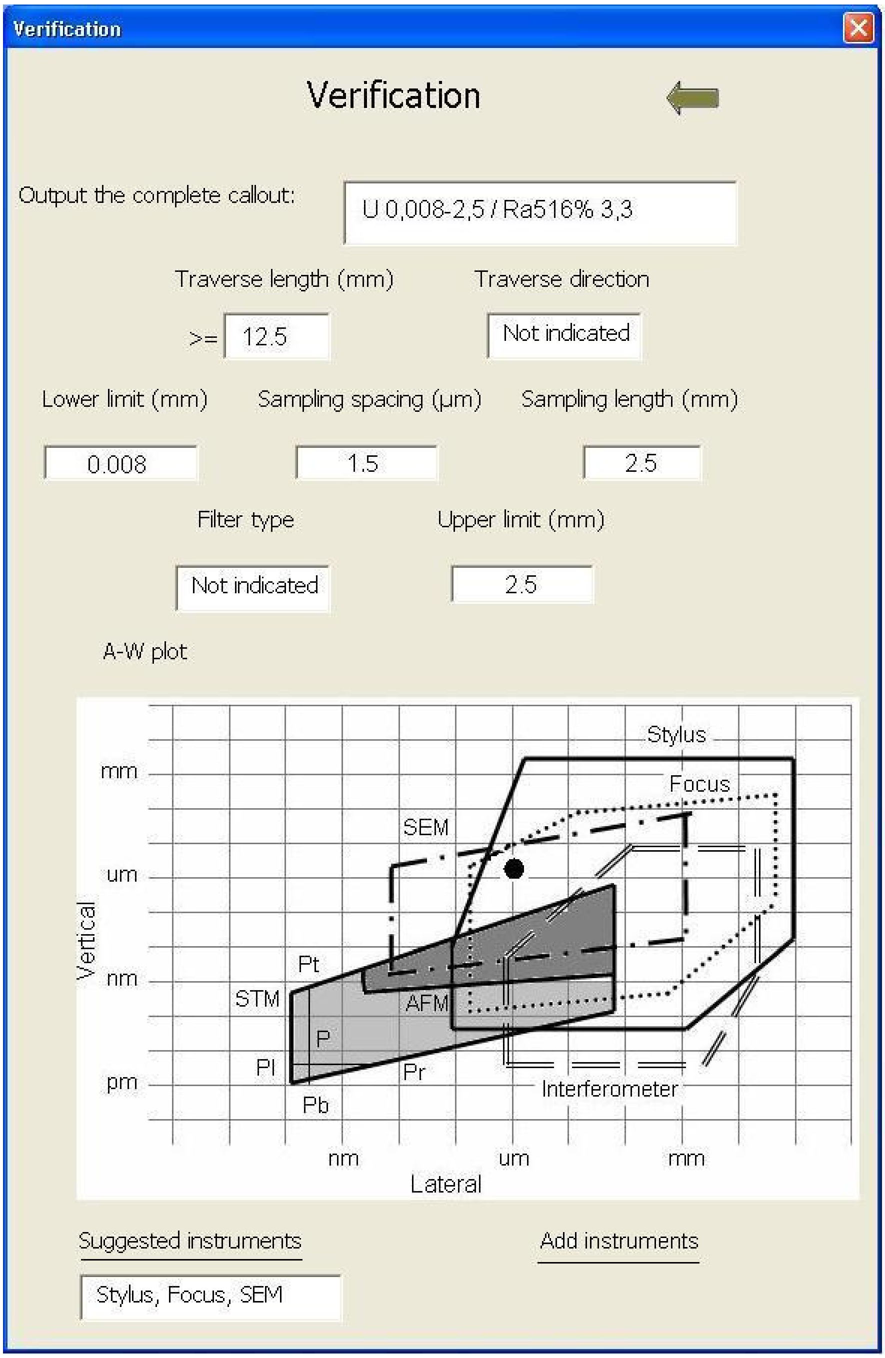


Figure 6.25 Interface of verification

6.5 Summary

This Chapter describes the detailed design procedures for verification knowledge-base, including:

Knowledge acquisition – the knowledge is mainly coming from ISO standards [4-5], [7-11] and divided into five parts: partition, extraction, filtration, measured value and instrument. The refinement processes of each part are provided, which constitutes the final verification category model.

Knowledge representation –The complete category model represents the knowledge structure and rules for verification knowledge-base. The verification knowledge-base mirrors the specification knowledge-base, and the detailed mirror relations are explained. Main functions and the interface for the verification knowledge are provided.

References

[1] Richard Leach (2001), *Measurement Good Practise Guide No.37-The Measurement of Surface Texture using Stylus Instruments*, National Physical Laboratory, ISSN 1368-6550

[2] ISO 12179 (2000), *Geometrical Product Specifications (GPS) — Surface texture: Profile method — Calibration of contact (stylus) instruments*, International Organisation for Standardisation, Geneva

[3] ISO 5436 (2000), *Geometrical Product Specifications (GPS) — Surface texture: Profile method; Measurement standards — Part 1: Material measures*, International Organisation for Standardisation, Geneva

[4] ISO 1302 (2002), *Geometrical Product Specifications (GPS) — Indication of surface texture in technical product documentation,* International Organisation for Standardisation, Geneva

[5] ISO 3274 (1996), *Geometrical Product Specifications (GPS) - Surface texture: Profile method - Nominal characteristics of contact (stylus) instruments*

[6] http://www.taylor-hobson.com/faqsurfacedetail.php?faqid=27

[7] ISO 4287 (1997), *Geometrical Product Specifications (GPS) — Surface texture: Profile method — Terms, definitions and surface texture parameters*, International Organisation for Standardisation, Geneva

[8] 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 Organisation for Standardisation, Geneva

[9] ISO 13565-3 (2000), *Geometrical Product Specifications (GPS) — Surface texture: Profile method; Surfaces having stratified functional properties — Part 3: Height characterization using the material probability curve*, International Organisation for Standardisation, Geneva

[10] ISO 12085 (1996), *Geometrical Product Specifications (GPS) — Surface texture:*

Profile method — Motif parameters, International Organisation for Standardisation, Geneva

[11] 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 Organisation for Standardisation, Geneva

[12] R.J. Hocken, N. Chkraborty and C. Brown (2005), *Optical Metrology of Surfaces*, Scientific Technical Committee-s meeting

[13] D J Whitehouse (1997), *Surface metrology*, Measurement Science and Technology **8** 955-972

[14] K.J. Stout (1994), *Three-Dimensional Surface Topography-Measurement, Interpretation, and Applications: A Survey and Bibliography*, Penton Press, Bristol, PA, ISBN-10: 1857180046

[15] Benjamin C. Pierce (1991), *Basic Category Theory for Computer Scientists*, MIT press, ISBN 0-262-66071-7

[16] M. Stedman (1987), *Basis for comparing the performance of surface measuring machines*, Precision Engineering 9(3), 149-152

[17] M. Stedman (1987), *Mapping the performance of surface-measuring instruments*, Proceedings of SPIE, 803, 138-142

[18] M. Stedman (1990), *How to compare different surface texture measuring instruments*, Quality Today, April 1990, 35-36

Chapter 7

Function Refinement

7.1 Introduction

This chapter aims to provide a detailed explanation of refinement processes for the function knowledge-base. The function knowledge-base is used to determine relationships between functional performances for different surfaces and specific surface roughness parameters.

Surface can be divided into two categories: functional and non-functional [1]. A non-functional surface can be either mirror smooth or sand paper rough without influencing the quality of the part. A functional surface is required to perform a function that is related to the users' perception of the part's quality. For example, the outside of an engine block in figure 7.1 is a non-functional surface, which has no specific function; while the contact area between the cylinder liner and the piston rings are functional surfaces, performing the function sealing.

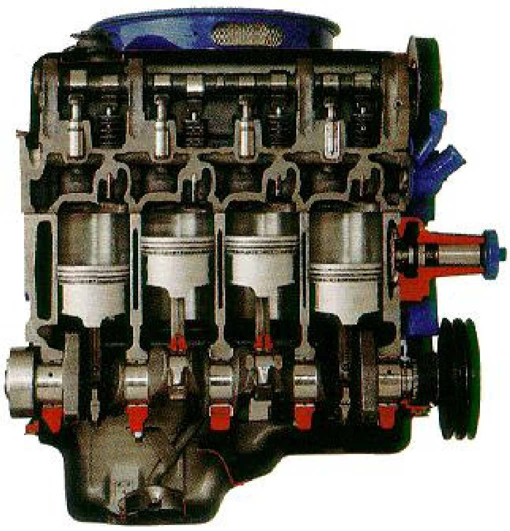


Figure 7.1 Engine block

For the functional surface, the surface texture has a direct influence on the quality of the part. When the relationship between surface topography and part function is known, quality can be optimised [1]. Therefore, the determination of the surface roughness parameters of a certain functional surface is very important and would help a manufacturer to produce a suitable workpiece matching requirements.

However, there is no easy way to select appropriate surface roughness parameters and values specified on technical drawings, because there are few references available to help the designer.

A function knowledge-base is being developed to assist designers during the initial stages of the engineering design process and to address the selection of corresponding surface roughness parameters with functions. This system will incorporate available resources about parameters selection rules and also would be an open platform for experts or designers to add their expertise and specific examples as well.

The functions of the system are described below:

- For designers:

  Job description: requesting the parameters necessary to specify the functional requirements on a technical drawing.

  Input: surface requirements, function performance, wear types, specific examples.

  Output: suggested procedures to obtain surface roughness parameters and limit values.

- For experts:

  Job description: adding new rules or examples into the knowledge-base

  Input: corresponding surface roughness parameters and limit values with certain function performance or specific examples

  Output: successfully add, and expand the knowledge-base

Chapter 4 mentioned that the relationship between parameters selection and functional performances is a qualitative, fuzzy classification scale. The system will try to incorporate available resources about parameters selection rules and also develop an open platform for experts or designers to add their expertise and specific examples later. Therefore, a proper data model is needed to represent this structure.

The popular used relational data model is not adequate enough to represent hierarchical structures. And the object-oriented data model lacks a theoretical foundation to secure the reliability. A pattern language [2] is adopted to represent the knowledge in this knowledge-base, which not only provides an open platform structure, but also takes a tree structure matching the structure of function knowledge, (further comparison of data models refers to section 7.3).

7.2 Parameters selection for different functions

7.2.1 Functions

The most important functional demand is related to the interaction of workpieces, which is called tribology. Tribology is the science and technology of friction, wear and lubrication [3]. It is important to understand the relationships between functional performance and the aspects of tribology, which would make it easier to specify the surface roughness parameters for different surfaces and then manage the optimal manufacturing of these surfaces.

Table 7.1 lists some common functional surfaces with the quality of the part. The surface texture has a direct influence on the quality of the workpiece. The following section introduces the different situation of qualities, i.e. the interaction of workpieces, because it is helpful to understand the correlation between function and surface topography.

Function	Criterion for quality	Example
Bearing	Low friction, wear	Crankshaft journal
Conduction	Large contact area	Electrical switch
Decoration	Consistent light reflection	Automotive body panel
Adhesion	Roughness, surface area	Sheet steel
Friction	Sharp peaks, low contact area	Driving roller
Sealing	Large contact area, low wear	Piston ring

Table 7.1 Relationship between surface function and quality [1]

7.2.1.1 Tribology

Tribology is the science and technology of friction, wear and lubrication [3].

Friction is defined as the resistance to relative motion between two bodies in contact, under a normal load [3], including friction in metals and friction in plastics and ceramics. Surface texture has an influence on the coefficient of friction, and as a result affects the friction.

Wear is defined as the progressive loss or removal of material from a surface. Wear has a great technologic and economic significance because it changes the shapes of tools and dies, and consequently affects the size and quality of the parts produced [3]. Wear may alter a part’s surface texture and result in severe surface damage. Therefore, it is important to select appropriate surface parameters in order to characterise an appropriate surface in which the effect of wear is reduced and ensure the long life of products. Wear is usually classified as following [3]:

- "Adhesive wear is caused if a tangential force is applied to the model and shearing takes place either at the original interface or along a path below or above it.
- Abrasive wear is caused by a hard, rough surface sliding across another surface.
- Corrosive wear is caused by chemical or electrochemical reactions between the surfaces and the environment.
- Fatigue wear is caused when the surface of a material is subjected to cyclic loading.
- Erosion wear is caused by loose abrasive particles abrading a surface.
- Fretting corrosion wear occurs at interfaces that are subjected to very small reciprocal movements.
- Impact wear is the removal, by impacting particles, of small amounts of material from a surface."

Lubrication is the process of applying fluids to effectively reduce friction and wear. There are four types of lubrication [3]: Thick-film lubrication; thin-film lubrication; mixed lubrication; and boundary lubrication.

The surfaces of workpieces are subjected to forces and contact pressures, relative speeds and also temperatures. In addition, the valleys in the surface roughnesses of contacting bodies can serve as local reservoirs or pockets for lubricants, thereby supporting a substantial portion of the load. For example, the recommended surface roughness (e.g. Ra) on most dies is about 0.4μm [3].

7.2.1.2 Characterization of Function

The classification of function is not an easy job because it is so large and diverse and impossible to carry out a systematic approach. Figure 7.2 is a simplistic generic approach to provide guidance. It classifies tribological applications and in particular, contact, friction, wear, lubrication and failure mechanisms, which make up the bulk of engineering functions [4].

The classification of function in figure 7.2 is broken down into two variables. The lateral axis represents the relative velocity of two surfaces, while the vertical axis represents the gap between two surfaces. The scales are omitted in the diagram. It is supposed that the vertical axis is in micrometers and the lateral axis has a maximum realistic value of 5m/sec. The diagram can be used to fit surface characteristics.

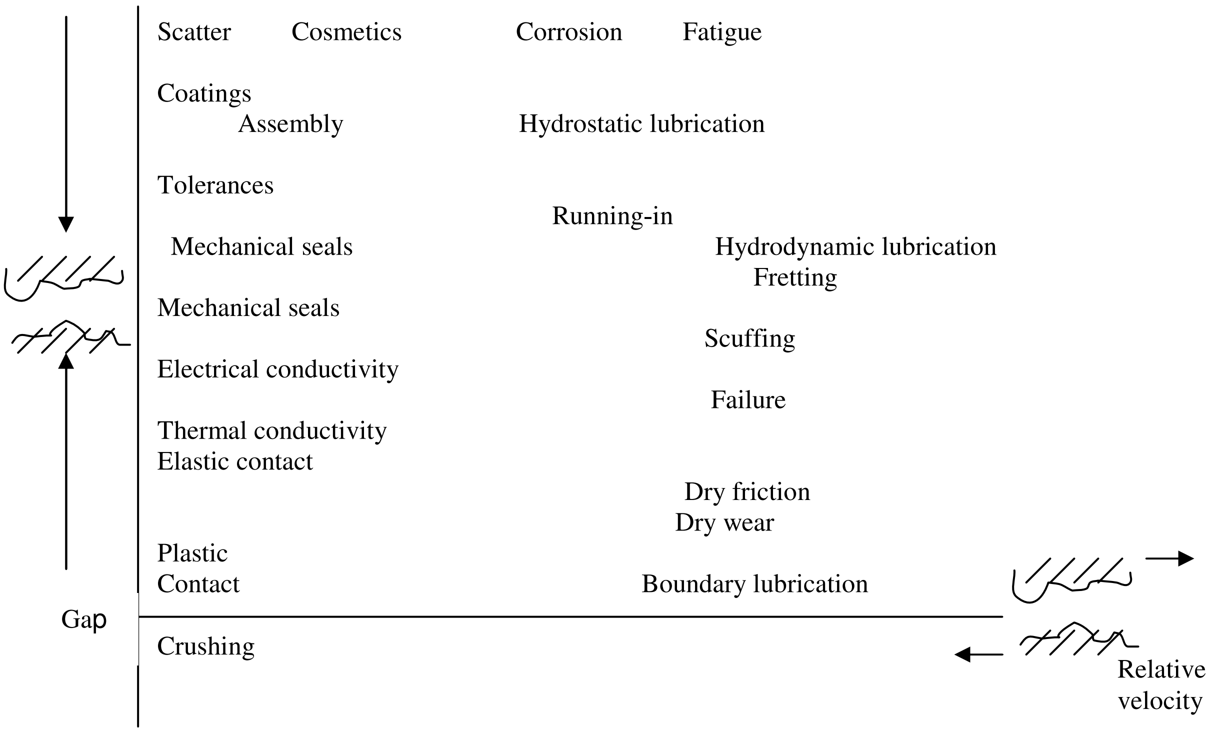
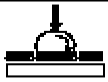
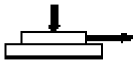
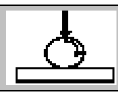
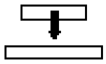
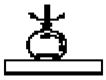
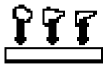
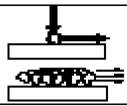
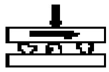
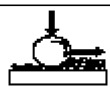
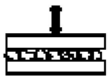
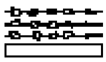
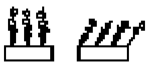
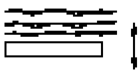
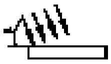
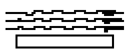


Figure 7.2 Function map [4]

Figure 7.3 shows another view of classifying functions applied to surfaces, which links contact bodies, relative motions and types of wear together [5]. It is a further illustration of the figure 7.2. In this diagram, the two attributes used to classify the function are also the separation of two surfaces and the relative velocity as shown in figure 7.2. The diagram takes a tree structure: on the first level, the surfaces are classified by their properties and contact elements; and then on the second level, the surfaces are further classified by the relative motions between them; and on the third level, the types of wear are suggested for each kind of contact surfaces.

This diagram provides guidance to link the surface requirements with functional performances. For example, see the highlight in figure 7.3, the surface requirements are two solid bodies contact with a rolling motion between them. Then the most important functional demands for the surfaces are fatigue wear.

Elements	Relative motions		Type of wear	Mechanism of wear			
	Type	Schemes	Type	AD	AB	WF	TC
Solid body Lubricant Solid body	Sliding		Hydrodynamic			●	○
Solid body Solid body	Sliding		Sliding wear	●	○	○	●
	Rolling		Rolling wear	○	○	●	○
	Impact		Impact wear	○	○	●	○
	Vibration		Fretting	●	●	●	●
Solid body Particles	Impact		Abrasion		●	●	○
	Sliding		Abrasion		●		○
Solid body Particles Solid body	Sliding		Abrasion	○	●	●	○
	Rolling		Abrasion	○	●	●	○
	Impact		Abrasion	○	○	●	○
Solid body Particles Fluid	Flow		Erosion		●	●	○
Solid body Particles Gas	Flow		Erosion	○	●	●	○
			Erosion	○	●	●	○
Solid body Fluid	Flow		Cavitation Erosion			●	○
	Impact		Erosion			●	○
	Flow		Erosion by fluids			○	●
Solid body Gas	Gas Erosion		Cavitation Erosion				●

AD – Adhesion, AB – Abrasion, WF – Wear Fatigue, TC – Tribocorrosion,
●More important ○ Less important

Figure 7.3 Examples of function [5]

7.2.2 Parameters selection

Surface texture parameters are used to reflect surface differences. A surface texture parameter should only be used if it is sensitive to the important surface characteristics, i.e. the variation of the parameter values would change the main characteristics of the surface. The selection of surface texture parameters should be carried out with this criterion in mind [1].

A surface texture value is specified that [1]:

- is possible to measure
- is possible to manufacture
- will allow the part to function
- is not smoother than necessary
- conforms to the preferred roughness values shown in table 7.2

Roughness Ra (µm)				
0.006	0.012	0.025	0.05	0.1
0.2	0.4	0.8	1.6	3.2
6.3	12.5	25	50	

Table 7.2 Preferred roughness values Ra [1]

For example, figure 7.4 and figure 7.5 give two examples showing how to choose surface parameters according to functions. They are only a very small contribution to this subject area. There is little information available at the moment to link surface texture to function performance. A lot more work needs to be done in this direction.

The relationship between motif parameters and function of surfaces contained in ISO 12085 1996 [6] is reproduced in figure 7.4. The left part of the table lists the classification of surface function, which takes a tree structure. The surface function is classified using the two variables: the gap between surfaces and the relative velocity as shown in figure 7.2. The right part of the table is a relationship table for motif parameters, including the relations between parameters and different functions, and relations between parameters. This table links the surface requirements, functional performance of the surface and parameters selection for the surface together. For example, see the highlight in figure 7.4, the surface requirements are two parts contact with relative displacement between them. The function applied to the surface is the rolling wear. Then the most important motif parameters are roughness parameter R (mean motif height) and waviness parameter Wx (maximum motif height). If parameter W is not specified, the upper tolerance on W is equal to the upper tolerance on R multiplied by 0,8.

Surface		Functions applied to the surface			Parameters								
					Roughness profile			Waviness profile				Primary profile	
		Designations		Symbol[a]	*R*	*Rx*	*AR*	*W*	*Wx*	*Wte*	*AW*	*Pt*	*Psc*
two parts in contact	with relative displacement	Slipping (lubricated)		FG	●			≤0,8R			○		●
		Dry friction		FS	●		○		●		○		
		Rolling		FR	●			≤0,8R	●		○		○
		Resistance to hammering		RM	○		○	○			○		●
		Fluid friction		FF	●		○				○		
		Dynamic sealing	with gasket	ED	●	○	○	≤0,6R	●		○		
			without gasket		○	●		≤0,6R					●
	without displacement	Static sealing	with gasket	ES	○	●		≤R		○	○		
			without gasket		○	●		≤R		●			
		Adjustment without displacement with stress		AC	○								●
		Adherence (bonding)		AD	●							○	
Independent surface	with stress	Tools (cutting surface)		OC	○		○	●			●		
		Fatigue strengths		EA	○	●	○						○
	without stress	Corrosion resistance		RC	●	●							
		Paint coating		RE			○				○		
		Electrolytic coating		DE	●	≤2R	●						
		Measures		ME	●			≤R					
		Appearance (aspect)		AS	●		○	○			○		

● Most important parameters: specify at least one of them.
○ Secondary parameters: to be specified if necessary according to the part functions.
The indication ≤ 0,8R, for example, means that, if the symbol FG is indicated on the drawing, and W not otherwise specified, the upper tolerance on W is equal to the upper tolerance on R multiplied by 0,8.

[a] The symbols (FG, etc.) are acronyms of French designations.

Figure 7.4 Parameters selection example [6]

Figure 7.5 lists some common functions for real examples, together with suggested values of the surface roughness parameter Ra, Leigh Mummery (1990) [1].

Function	**Example**	**Suggested value of the surface roughness parameter Ra (µm)**
Non-functional	Engine block	50
Fluid friction	Ship hull plate	25
Joint face (soft gasket)	Pipe flange	12.5
Assembly face	Gearbox housing	6.3
Bearing (slow speed)	Hand crank	3.2
Clearance fit	Bolt shank	1.6
Lubricated bearing seal	Gear teeth	0.8
Bearing (high speed)	Engine cylinder bore	0.4
Rolling bearing, Metallic seal	Valve follower, Hydraulic cylinder	0.2
Measuring reference	Gauge	0.1
Performance seal	Fuel injector	0.05
Wring surface	Gauge block	0.025

Figure 7.5 Selection of roughness values according to function [1]

The information above shows that there is no systematic approach covering all aspects of functions and the associated surface texture. The organisation of function is developed in this thesis and represents a considerable step forward and will contribute to the subject area by providing a framework to which functional knowledge can naturally expand. The knowledge stored is large, fuzzy and takes a tree structure as shown in figure 7.3 and figure 7.4 (see Chapter 4).

The data model for the function knowledge is first refined to an abstract knowledge representation model as shown in figure 7.6, which is derived from the references and information mentioned above.

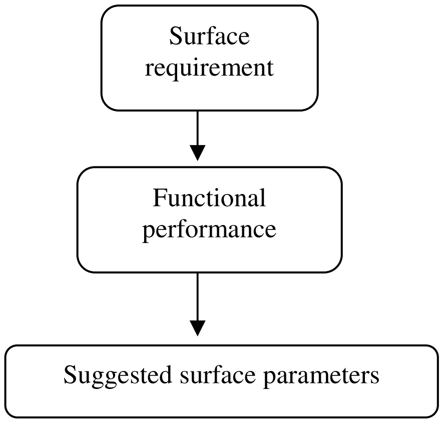


Figure 7.6 Abstract model for the Function knowledge-base

7.3 Pattern Language to represent the model

As mentioned before the relationship between parameters selection and functional performances is a qualitative, fuzzy classification scale. A new structure – pattern language is applied to represent this relationship [2], see section 2.3.3.

7.3.1 Introduction of Pattern Language

The term 'Pattern language' was initially introduced in Christopher Alexander's book 'A Pattern Language: Towns, Buildings, Construction' [2], which describes common problems of civil and architectural design, from how towns should be laid out to where windows should be placed in a room. The idea was soon expanded into popular diverse fields, such as computer science patterns used in software engineering, interaction design patterns in human computer interaction and pedagogical patterns in education.

A single design pattern in a pattern language is defined as a common problem (or decision) in a design process, together with its best solution. Each pattern has a name, a descriptive entry, and some cross-references to other patterns, much like a dictionary entry. Just as words must have grammatical and semantic relationships to each other in order to make a spoken language useful, design patterns must be related to each other in order to form a pattern language. The patterns should be organized into a logical or naturally intuitive structure. Each pattern should indicate its relationship to other patterns and to the language as a whole [7].

A pattern language is a linked structure of patterns within a particular domain. It is characterized by

1) Naming the common problems in a field of interest
2) Describing the key characteristics of effective solutions for meeting some stated goals
3) Helping the designer move from problem to problem in a logical way
4) Allowing for many different paths through the design process

A pattern language is not a kind of programming language, it is a piece of literature that describes an architecture, a design, a framework or other structure. A pattern is applied to document expertise. The typical pattern form includes the following items [8]:

Name: A word or simple phrase to describe the pattern.
Context: When to apply the pattern.
Problem: A statement of the problem.
Solution: A solution to the problem. Many problems have more than one solution. The applicability of a solution is affected by the context or circumstances in which the problem exists.
Force: The often contradictory considerations to be considered when choosing a solution to a problem. Each solution considers certain forces. It optimizes some and may totally ignore others. The relative importance of the forces is determined by the context.

An example from Christopher Alexander 1977 is the "ARCADES" pattern [2]:

Pattern name: Arcades
Context: connections between buildings, etc.
Problem: how to use the arcades?
Solution: covered walkways at the edge of buildings, which are partly inside, partly outside-play a vital role in the way that people interact with buildings.

Next pattern: try Columns, Roof, Ceiling height.

7.3.2 Comparison with other alternative models

The model representing function knowledge-base should be able to take a hierarchical tree structure, and support fuzzy classification scale as well. And most importantly, since the knowledge now available is far less than adequate for a full solution, the model should support the open platform model and be able to incorporate more and more expertise to make it more complete.

The popular used relational data model is not suitable because of its significant drawback to adequately represent hierarchical structures [9].

The object-oriented data model has the flexibility for knowledge to take any structure and supports the building of complex objects, which form a more natural and realistic representation of real-world objects, but it lacks a theoretical foundation [9].

The pattern language satisfies all the above requirements very well. Figure 7.7 shows a structure of a pattern language. A Pattern consists of at least: a name of the pattern, a context for the pattern, a reoccurring problem in that context, the core solution to the problem, links to sub- and super- patterns. A pattern language is a linked structure of patterns as shown below.

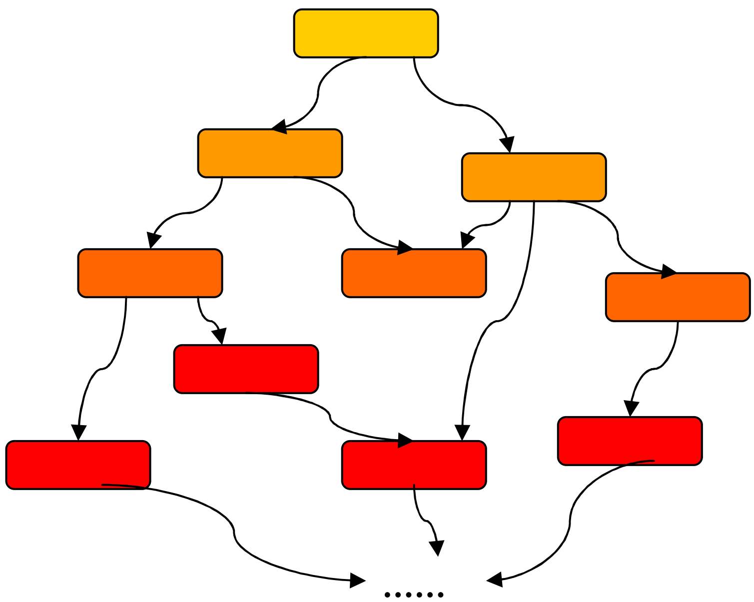


Figure 7.7 Diagram of a pattern language

A pattern language is a structured method of describing good design practices or applications within a particular domain. It is more like a structured documentation. It is possible to add more patterns of each level and to collect and unify these patterns. The pattern language won't solve the problems directly, it provides some possible solutions in documentation, but still requires users' opinion and judgment. Pattern language can be applied to the problem which does not have a specific solution. Fuzzy sets consider the probability and generate a function to represent the solutions [10]. Whereas pattern language gives several solutions under different contexts and forces, and then users can choose the most suitable one matching the current circumstance.

The function pattern language is designed to help designers find the best way to specify surfaces, by providing inexperienced designers with substantial material to carry out the task, guiding them during the procedure, providing alternative solutions as necessary and giving orientation of the next steps to be explored. It can record all valuable methods and link them together in a logical structure [11].

7.3.3 Function pattern language

A detailed description of example patterns for the start of a function knowledge-base is given below:

Pattern 1 — Surface requirements

<u>Context</u>: For each surface on a workpiece, certain surface requirements should be specified for designers to select the suitable specifications.

<u>Problem</u>: Specify the surface requirements.

<u>Solution</u>: Investigate the counterparts, the properties and the relative motion of the workpiece, and then specify the surface requirements to match those attributes.

<u>Forces</u>: In some circumstances, it is hard to specify a particular surface requirement because of lack of consideration and expertise. More research and references are needed to make this pattern more efficient.

<u>Example</u>: For a cylinder liner on an engine block, the counterpart is the piston ring, and surface requirements are that it needs to have a good bearing surface but also retain a reservoir of oil for lubrication [12].

<u>Next pattern</u>: After specifying the surface requirements, try Pattern 2 — Functional performance.

Pattern 2 — Functional performance

Context: The function knowledge-base is used to find out the relationships between functional performances for different surfaces and specific surface roughness parameters. It is an important step to analyze the functional performance for a certain workpiece according to the design requirements.

Problem: Determination of the functional performance according to Pattern 1 — Surface requirement.

Solution: For each surface requirement of the workpiece, find out a suitable functional performance. The relevant data information can be found on table 7.1, figure 7.2, figure 7.3, figure 7.4 and figure 7.5.

Forces: There are not enough references to describe the relations between suitable functional performances and surface requirements. More examples are needed to complement this pattern.

Example: The surface requirements for a cylinder liner on an engine block are that it needs to have a good bearing surface but also retain a reservoir of oil for lubrication. Therefore, the most important functional demands are oil consumption, blow-by, and wear specially at the TDC (top-dead centre) [12]. Partial information concerning the functional performances listed in figure 7.3, figure 7.4 and figure 7.5 would be concluded in this part.

Next pattern: After the analysis of functional performances, try Pattern 3 — Surface parameters selection.

Pattern 3 — Surface parameters selection

Context: The designers need to select the suitable specification for a surface in order to ensure the surface functions correctly.

Problem: Determination of the surface parameters to satisfy Pattern 2 — Functional performance of the surface.

Solution: There are two basic approaches:

1) Establish Pattern 4 — Functional correlation with texture parameters;
2) Establish a stable surface generation process that produces acceptable surfaces and then Pattern 5 — Monitor for surface changes.

Forces: The functional correlation approach is superior in quality of results but is more expensive in time and cost to establish correlation, and more sophisticated measuring equipment is required than establishing a stable surface generation process and monitoring for surface changes.

Example: The surface requirements for a cylinder liner on an engine block are that it needs to have a good bearing surface but also retain a reservoir of oil for lubrication.

1) The texture parameters Rk and friends have been shown to have a functional correlation with the desired surface tasks.
2) One approach for manufacture is with a plateau-honed surface. Rq & Rsk can be used to monitor for surface changes.

Next pattern: After the surface parameters selection, try Pattern 4 — Functional correlation and Pattern 5 — Monitor for surface changes.

Pattern 4 — Functional correlation

Context: The designers need to find the relationships between surface texture parameters and the functional performances.

Problem: Determination of the functional correlation between surface parameters and the functional performance of the surface, in order to give the suggestions to Pattern 3 — Surface parameters selection.

Solution: Using factorial designed experiment (FDE) to find the correlations between surface roughness parameters and functional performance indicators, and then analyse the correlation coefficient from the results, to determine the relationships between surface texture parameters and the functional performances.

Forces: The functional correlation approach is superior in quality of results but is more expensive in time and cost to establish correlation and more sophisticated measuring equipment is required.

Example: The surface requirements for a cylinder liner on an engine block are that it needs to have a good bearing surface but also retain a reservoir of oil for lubrication. The texture parameters Rk and Rz have been shown to have a functional correlation with the desired surface tasks [12]. Partial information concerning the surface texture parameters listed in table 7.2, figure 7.4 and figure 7.5 would be concluded in this part.

Next pattern: After the surface parameters selection, try Pattern 6 — suggestion of limit values.

7.4 Category model to represent function pattern language

In this thesis, a hybrid method using the pattern language is used for functional decomposition and function to structure mapping process. A category model based on the particular concepts of Category Theory: Partial order set and Product order [13], is employed to represent and record decomposition alternatives.

7.4.1 Partial order set of the “Contexts”

Partial order [13]: A partial order is a binary relation *R* over a set *P* which is reflexive, antisymmetric, and transitive, i.e., for all *a*, *b*, and *c* in *P*:

- *aRa* (reflexivity);
- if *aRb* and *bRa* then *a* = *b* (antisymmetry);
- if *aRb* and *bRc* then *aRc* (transitivity).

A set with a partial order is called a partially ordered set (also called a poset).

An example of poset is the set of subsets of a given set (its power set) ordered by inclusion, see figure 7.8.

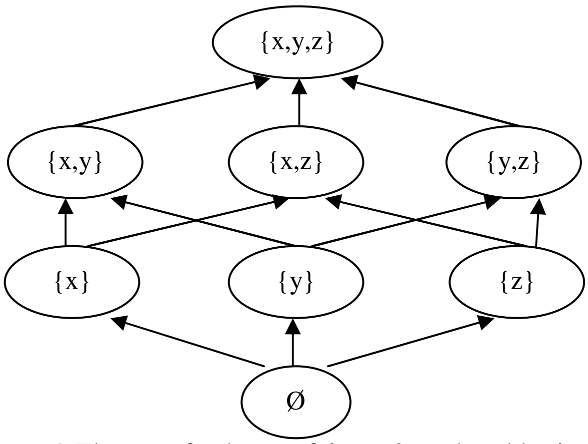


Figure 7.8 The set of subsets of {x,y,z}, ordered by inclusion

The patterns can be represented in a partial order set, which are connected with each other by the context of each pattern, and ordered by design sequence, see figure 7.9.

- Pattern 1 solves the problem to specify the surface requirements;
- After design requirements are specified, Pattern 2 then analyses the functional performances according to the surface requirements;
- Given the functional performance of the surface, surface parameters can be selected to match different functions in Pattern 3;

- In order to give the corresponding parameters, it is necessary to find the relations between functional performance and surface parameters, Pattern 4 suggests a functional correlation approach to achieve that target;
- Pattern 5 suggests another approach – surface changes monitor to find the relations between functional performance and surface parameters;
- After the surface parameters are specified, it would be possible to specify the limit value for each parameter in Pattern 6;
- …

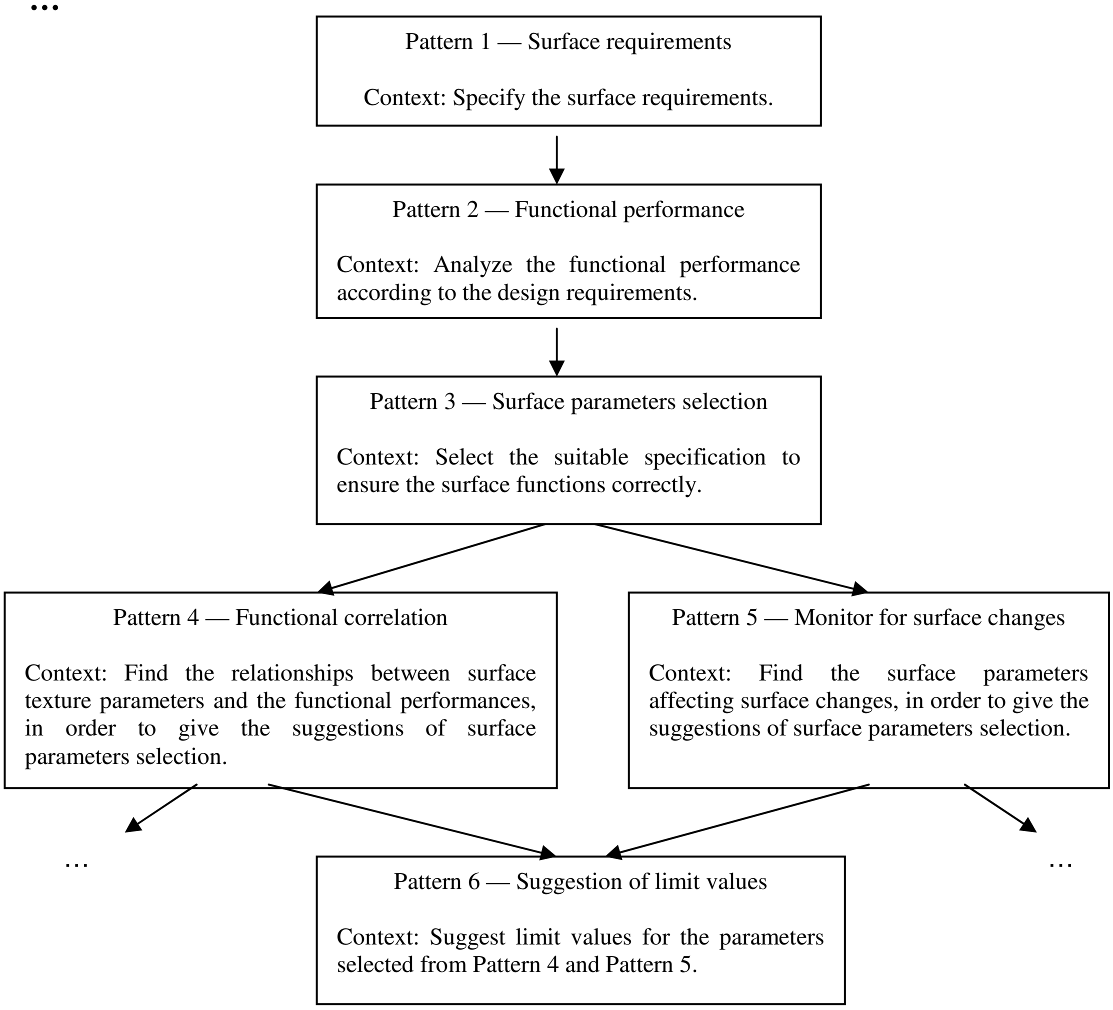


Figure 7.9 Poset of "Contexts"

7.4.2 Product order of the Posets

Product order connects two posets together and generates an ordered product, i.e. Let $(X, \leq^*)$ and $(Y, \leq')$ be finite posets, Consider the Cartesian product $X \times Y = \{(x, y) \mid x \in X, y \in Y\}$, and define a binary operation on $X \times Y$ by $(x_1, y_1) \leq (x_2, y_2)$, if and only if $x_1 \leq^* x_2$, $y_1 \leq' y_2$. This operation is called a product order [13]. Figure 7.10

gives an example, X and Y are two given posets, while X · Y is the product order of X and Y [13]:

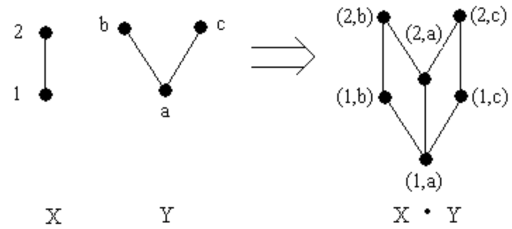


Figure 7.10 Example of Product order

As stated before, the Contexts of each pattern constitute a poset Y, and the other elements in a pattern form constitute another poset X, therefore, the Function pattern language is able to be represented as the product order of X and Y, shown in the following data model, see figure 7.11:

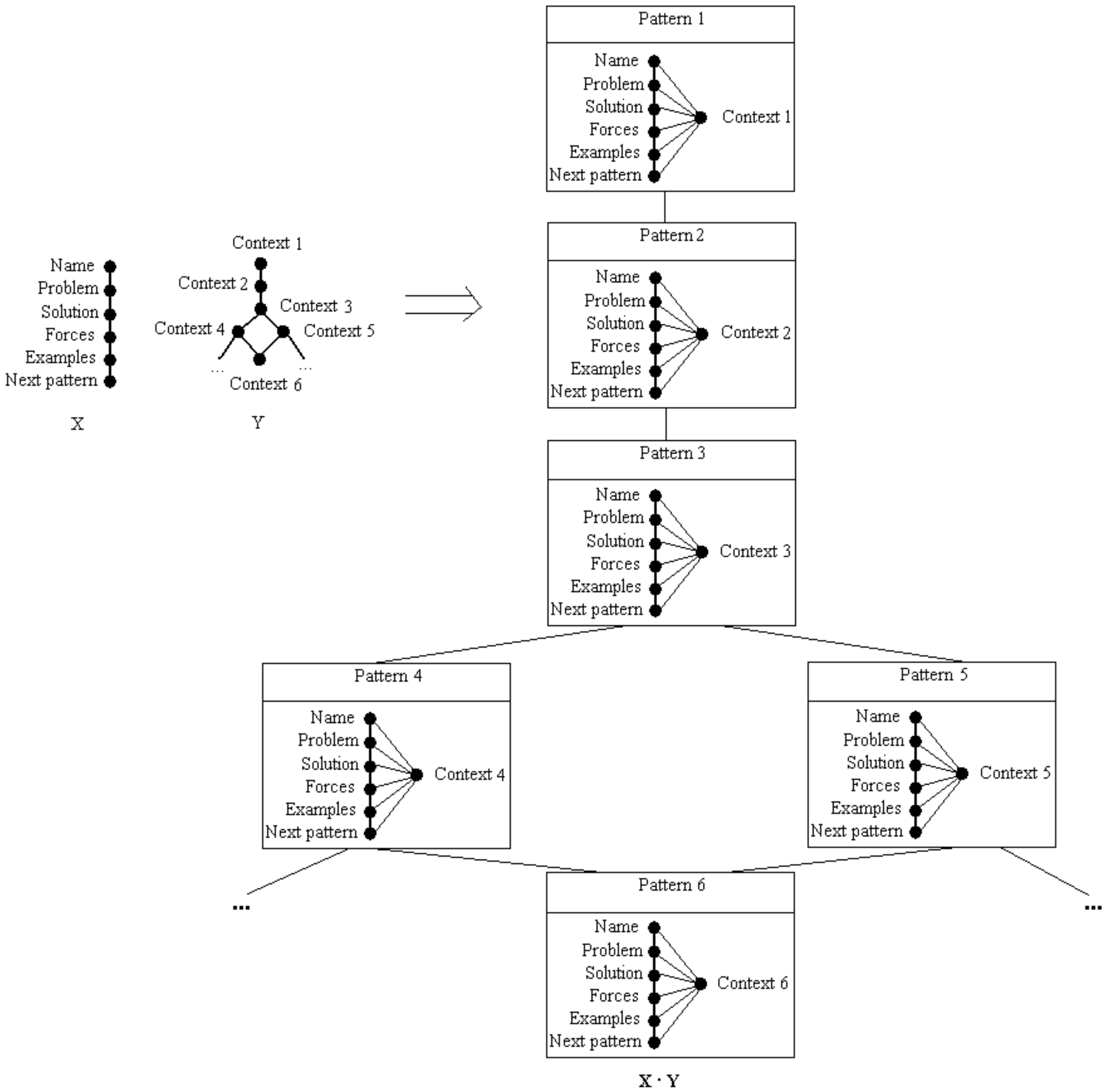


Figure 7.11 Product order of Function pattern language

The data model of the Functional knowledge-base is finally refined to the complete implementable model as shown in figure 7.11, which takes a tree structure and maintains an open platform for further information to be added.

7.4.3 Interfaces of the Function knowledge-base

Figure 7.12 shows the interface of Function knowledge-base. It allows users to view different patterns and also insert the new patterns. The left textbox lists the context of each pattern. When users click the particular one, the corresponding pattern form will display in the right textbox, including name, problem, solution, forces, examples and next patterns. Users could go through the patterns they want and find the best solutions for their problems.

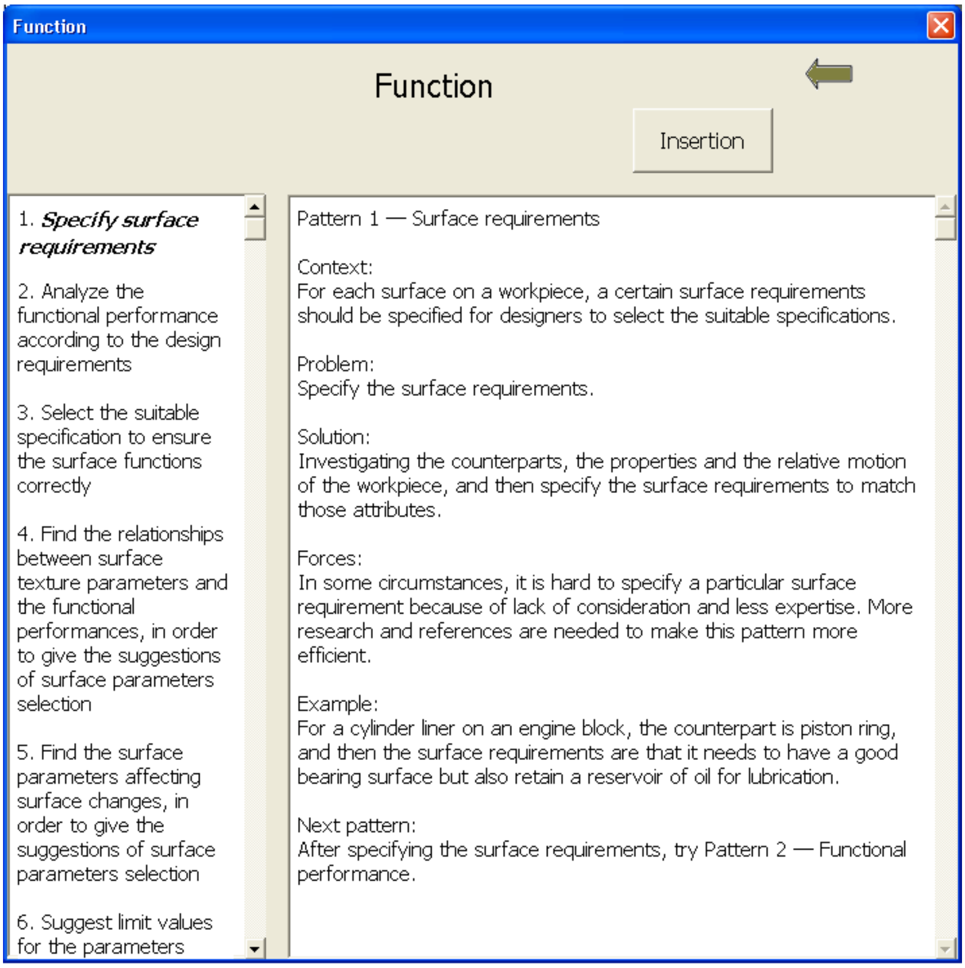


Figure 7.12 Interface of the Function knowledge-base

Take a cylinder liner in an engine block such as in figure 7.1 for example, the function knowledge-base will suggest to designers that the surface requirements for the cylinder liner are that it needs to have a good bearing surface but also retain a reservoir of oil for lubrication. And then the next pattern will suggest to designers that most relevant motif parameters are roughness parameter R (mean motif height) based on resources from figure 7.4.

7.4.4 Insertion of the new pattern

Since there is little information available at the moment to link surface texture to functional performance, the model now present is only a basic structure of patterns, it is needed to add more patterns in each level and to collect and unify these patterns. New patterns can be added into the model, following the steps below:

I. Insertion of a new example:

1) Find the pattern which the example belongs to;

2) Insert the example.

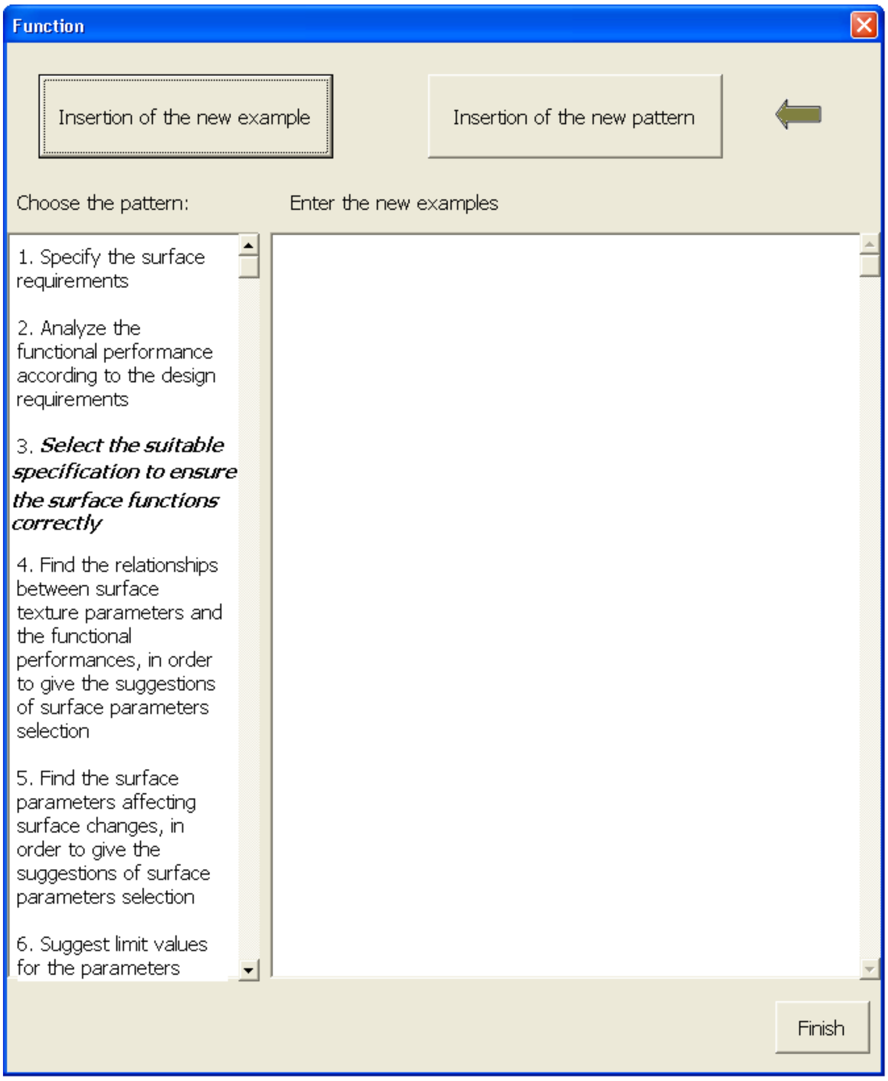


Figure 7.13 Interface of the insertion of a new example

Figure 7.13 shows the interface of the insertion of a new example. Choose the pattern in the left textbox, and enter the new pattern in the right textbox.

II. Insertion of a new pattern:

1) Consider when to apply the new pattern, i.e. write the context of the new pattern;
2) Find the right position in the Contexts poset for the new pattern;
3) Obtain the new product order of the poset X (Name, Problem, Solution, Forces, Examples and next pattern) and Contexts poset.

Figure 7.14 shows the interface of the insertion of a new Pattern. Write the context of the new pattern in the upper left textbox, and then find which pattern to follow, finally complete the pattern form on the right.

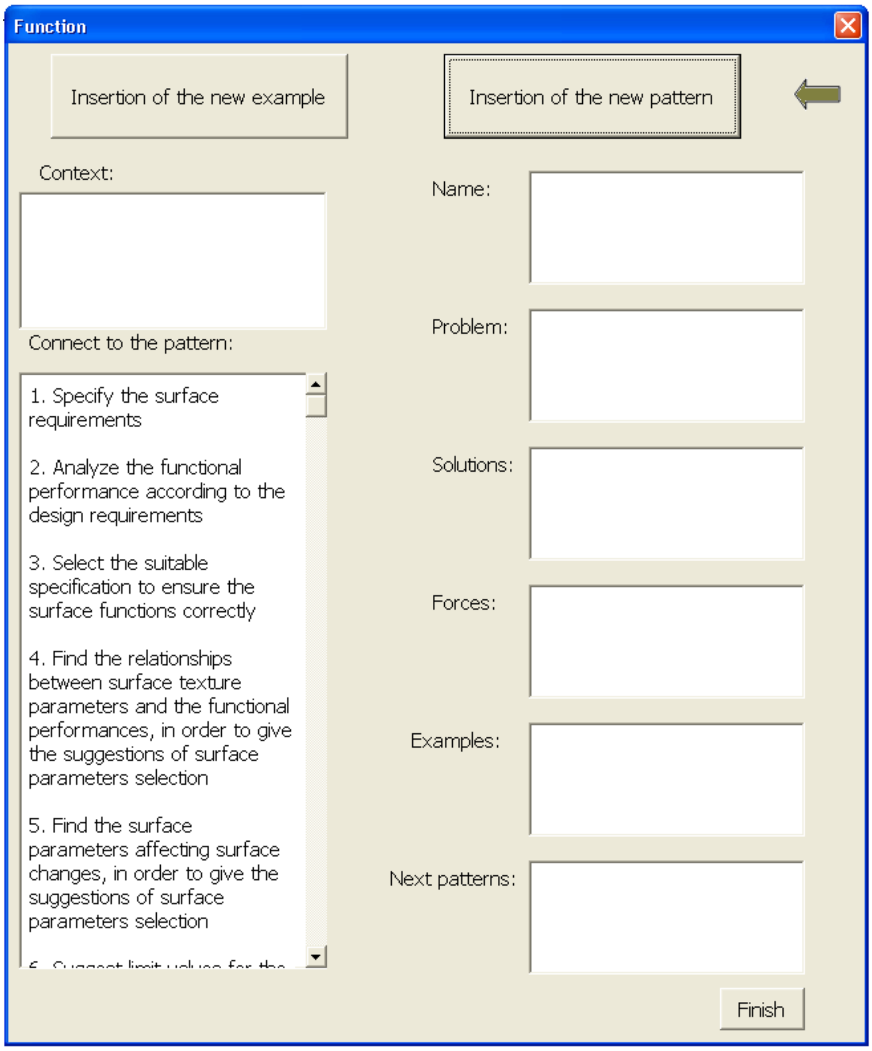


Figure 7.14 Interface of the insertion of a new Pattern

7.5 Summary

This Chapter describes the detailed design procedures for function knowledge-base, including:

Knowledge acquisition – The relationship between functional performance and surface texture parameters are analysed, and the hierarchical structure is adopted to represent this relationship in the function knowledge-base.

Knowledge representation – The pattern language is applied to represent this hierarchical structure, which also supports the open platform model. A category model based on the particular concepts of Category Theory: Partial order set and Product order, is employed to represent and record the Function pattern language. Interfaces are given to further illustrate the implementation of the knowledge-base.

References

[1] Leigh Mummery (1990), *Surface Texture Analysis - The Handbook*, Hommelwerke

[2] Christopher Alexander (1977), *A Pattern Language: Towns, Buildings, Construction*, New York: Oxford University Press

[3] Serope Kalpakjian & Steven R. Schmid (2000), *Manufacturing Engineering and Technology*, Prentice Hall, ISBN 0-201-36131-0

[4] David Whitehouse (2002), *Surfaces and their measurement*, Hermes Penton Ltd, ISBN 1 9039 9601 5

[5] T. Filetin (2002), *The knowledge bases for selecting the surface modification treatment,* MT2002

[6] ISO 12085 (1996), *Geometrical Product Specifications (GPS) — Surface texture: Profile method — Motif parameters*, International Organisation for Standardisation, Geneva

[7] *Pattern language*, Wikipedia, the free encyclopedia, http://en.wikipedia.org/wiki/Pattern_language

[8] Linda Rising (1998), *The patterns handbook: techniques, strategies, and applications,* Cambridge University Press, ISBN 0-521-64818-1

[9] Thomas Connolly and Carolyn Begg (1998), *Database Systems: A Practical Approach to Design, Implementation, and Management,* Addison Wesley, ISBN 0-201-34287-1

[10] Peter Jackson (1999), *Introduction to expert systems*, Addison Wesley Longman Limited, ISBN 0-201-87686-8

[11] Rosana T. Vaccare Braga, Fernão S. R. Germano, Paulo Cesar Masiero (1999), *A Pattern Language for Business Resource Management*, PLoP 1999 Conference

[12] Robert Ohlsson, Bengt Goran Rosén and John Westberg (2003), *The interrelationship of 3D Surface Characterisation Techniques with Standardised 2D Techniques*, Advanced Techniques for Assessment Surface Topography, pp. 197-220, Kogan Page Science, ISBN 1 9039 9611 2

[13] J. Neggess & Hee Sik Kim (1998), *Basic Posets*, World Scientific, ISBN 981-02-3589-5

Chapter 8

Manufacture Refinement

8.1 Introduction

This chapter aims to provide a detailed explanation of refinement processes for the manufacture knowledge-base. The manufacture knowledge-base is used to determine the manufacture procedures: to select suitable manufacture processes to match the specification of the designed product.

8.1.1 Introduction of manufacturing processes

Manufacturing, in its broadest sense, is the transformation process of raw materials into finished products. It encompasses (1) the design of the product, (2) the selection of raw materials, and (3) the sequence of processes through which the product will be manufactured [1]. The manufacturing process is one of the important components in manufacturing industry. Generally, the manufacturing process can be classified into several units below:

Casting processes

Casting is a process by which a material is introduced into a mould while it is liquid, allowed to solidify in the shape inside the mould, and then removed producing a fabricated object or part [2]. The typical casting processes include sand casting, plastic mould, shell mould, investment, permanent mould, centrifugal, die, slush or slurry, full mould, low pressure, continuous moulding etc.

Moulding processes

Moulding is a strip of material with various cross sections used to cover transitions between surfaces or for decoration [2]. The typical moulding processes include powder metallurgy, plastics (injection, compression, transfer, extrusion, blow, rotational, thermoforming, laminating, expandable bead, foam etc.).

Forming processes

Forming is a process to change the shape of an existing solid body [2]. The typical forming processes include forging, rolling, extrusion, pressing, bending, piercing, miscellaneous other, shearing etc.

Machining processes

Machining involves using a power-driven machine tool, such as a lathe, milling machine or drill, to shape metal. Machining is a part of the manufacture of almost all metal products [2]. The typical machining processes include milling, turning, drilling, sawing, broaching, shaping, planing, honing, finishing, routing, hobbing, ultrasonic, electrical discharge, electron beam, electrochemical, chemical, photochemical, laser beam etc.

Joining processes

Joining is an all-inclusive term, covering processes such as welding, brazing, soldering, adhesive bonding, and mechanical fastening [2]. The typical joining processes include welding, brazing, soldering, sintering, adhesive bonding, fastening, stitching, stapling, press fitting etc.

Rapid manufacturing

Rapid manufacturing is a technique for manufacturing solid objects by the sequential delivery of energy and/or material to specified points in space to produce that solid [2]. The typical rapid manufacturing processes include stereo lithography, selective laser sintering, fused deposition modelling, three dimensional printing, laminated object manufacturing, laser engineered net shaping etc.

8.1.2 General manufacturing processes selection procedures

Since there are several types of manufacturing processes, the selection of a proper process for products becomes an important issue for manufacturers. Some engineers have practical experience of production, and understand the limitations and capabilities they must work within. Unfortunately, there are many who do not [5]. The manufacture knowledge-base would help both designers and manufactures to make the right decision and select the most suitable candidate manufacturing process. Figure 8.1 shows the process selection flowchart [5].

First of all, the candidate processes would be selected according to the specification requirements of products. The specification requirements include the material quantity, the texture lay, the limit value, etc. The surface texture related to manufacturing processes is discussed in section 8.3. Then the standard format of these candidate processes would be provided for manufacturers to carry out the comparison. The items listed in the format include economic considerations, typical applications, and design aspects.

Finally after considering and comparing all the necessary requirements, such as surface tolerances, materials, the costs and so on, manufacturers are able to make the right decision with the help of the knowledge-based system.

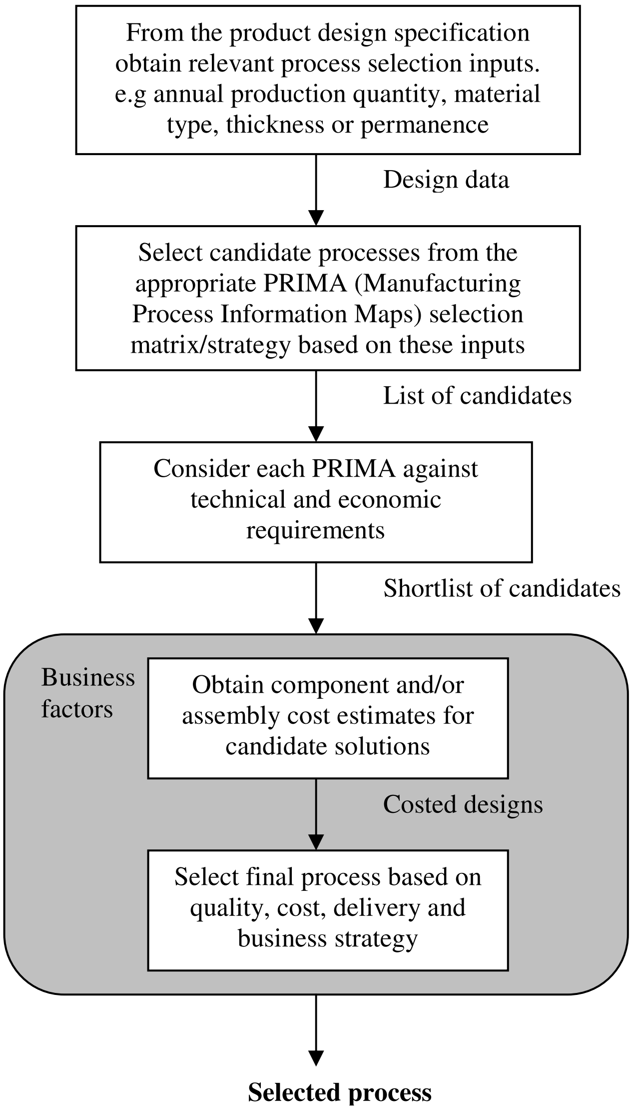


Figure 8.1 General process selection flowchart [5]

8.1.2.1 PRIMA

A set of so-called PRIMAs (Manufacturing Process Information Maps) [5] has been developed to enable the correct process selection. PRIMAs give detailed data on the characteristics and capabilities of each process in a standard format under general headings including: material suitability, design considerations, quality issues, general economics and process fundamentals and variations. The information includes not only design considerations relevant for the respective processes, but quite purposefully, an overview of the functioning of the process so that a greater overall understanding may be achieved. Within the standard format a similar level of detail is provided on each of the processes included.

The format is very deliberate. First an outline of the process itself - how it works and under what conditions it functions best. Second a summary of what it can do - limitations and opportunities it presents, and finally an overview of quality considerations including process capability charts for relating tolerances to characteristic dimensions [5]. According to the PRIMA standard format, each manufacturing process is divided into seven categories as listed and defined below [5]:

- Process Description: an explanation of the fundamentals of the process together with a diagrammatic representation of its operation.

  For example the drilling is a process that removal of material by chip processes using rotating tolls of various types with two or more cutting edges to produce cylindrical holes in a workpiece, see figure 8.2.

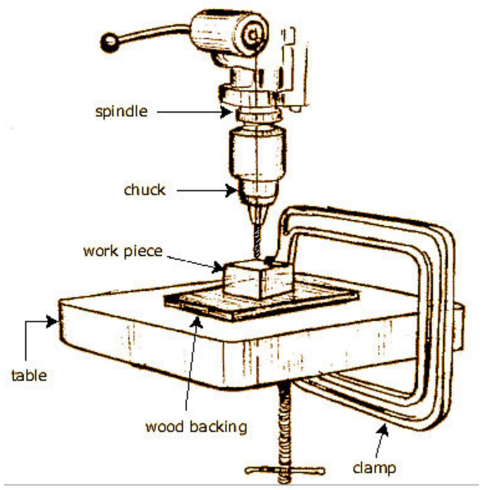


Figure 8.2 Example of drilling process

- Materials: describes the materials currently suitable for the given process.

  For example, the materials suitable for the drilling process are all metals and some plastics and ceramics.

- Process Variations: a description of any variations of the basic process and any special points related to those variations.

  For example, wide range of cutting tool materials is available for the drilling process.

- Economic Considerations: a list of several important points including: production rate, minimum production quantity, tooling costs, labour costs, lead times and any other points which may be of specific relevance to the process.
  For example, the tooling costs and finishing costs for the drilling process are low.
- Typical Applications: a list of components and parts that have been successfully manufactured using the process.
  For example, one of the typical applications for the drilling process is any component requiring cylindrical holes.
- Design Aspects: any points, opportunities or limitations that are relevant to the design of the part as well as standard information on minimum section, size range and general configuration.
  For example, flat-bottomed holes should be avoided for the drilling process.
- Quality Issues: standard information includes a process capability chart, surface roughness and detail, as well as any information on possible faults, etc.
  For example, surface roughness values ranging 0.4 – 12.5 µm Ra are obtainable for the drilling process.

8.1.2.2 PRIMA selection matrix

A PRIMA selection matrix has been devised based on two basic variables [5]:

- Material type
- Production quantity

This is a simple method based on material and production quantity which is designed to enable a user to focus attention on the most relevant PRIMAs. There are many cost drivers in process selection, such as component size, geometry, tolerances, surface finish, capital equipment and labour costs. The justification for basing the matrix on material and production quantity is that they mix PRIMA technological and economic issues of importance. The boundaries of economic production can be vague when so many factors are relevant, therefore, the matrix concentrates rather more on the use of materials [5].

Table 8.1 gives an example of part of the Manufacturing process PRIMA selection matrix. For example, if the material type is Irons and the production quantity is under 100, then the suitable manufacturing processes would be centrifugal casting, investment casting, ceramic mould casting and manual machining, see the highlights in table 8.1. Further comparison of other important issues in PRIMA forms of these suggested processes would be carried out to reach the final decision. The complete selection matrix please see K.G. Swift & J.D. Booker (2003) [5].

Quantity / Material	Very low 1 to 100	Low 100 to 1,000	Low to medium 1,000 to 10,000	Medium to high 10,000 to 100,000	High 100,000+	All quantities
Irons	[1.5] [1.6] [1.7] [4.M]	[1.2] [1.5] [1.6] [1.7] [4.M] [5.3] [5.4]	[1.2] [1.3] [1.5] [1.6] [1.7] [3.11] [4.A] [5.2]	[1.2] [1.3] [3.11] [4.A]	[1.2] [1.3] [3.11] [4.A]	[1.1]
Steel (carbon)	[1.5] [1.7] [3.10] [4.M] [5.1] [5.5] [5.6]	[1.2] [1.5] [1.7] [3.10] [4.M] [5.1] [5.3] [5.4] [5.5]	[1.2] [1.3] [1.5] [1.7] [3.1] [3.3] [3.10] [3.11] [4.A] [5.2] [5.3] [5.4] [5.5]	[1.9] [3.1] [3.3] [3.4] [3.5] [3.11] [3.12] [4.A] [5.2] [5.5]	[1.9] [3.1] [3.2] [3.3] [3.4] [3.5] [3.12] [4.A]	[1.1] [1.6] [3.6] [3.8] [3.9]

Key to manufacturing process PRIMA selection matrix:

[1.1] Sand casting
[1.2] Shell moulding
[1.3] Gravity die casting
[1.4] Pressure die casting
[1.5] Centrifugal casting
[1.6] Investment casting
[1.7] Ceramic mould casting
[1.8] Plaster mould casting
[1.9] Squeeze casting

[3.1] Closed die forging
[3.2] Rolling
[3.3] Drawing
[3.4] Cold forming
[3.5] Cold heading
[3.6] Swaging
[3.7] Superplastic forming
[3.8] Sheet-metal shearing
[3.9] Sheet-metal forming
[3.10] Spinning
[3.11] Powder metallurgy
[3.12] Continuous extrusion (metals)

[4.A] Automatic machining
[4.M] Manual machining

[5.1] Electrical discharge machining (EDM)
[5.2] Electrochemical machining (ECM)
[5.3] Electron beam machining (EBM)
[5.4] Laser beam machining (LBM)
[5.5] Chemical Machining (CM)
[5.6] Ultrasonic machining (USM)
[5.7] Abrasive jet machining (AJM)

Table 8.1 Example of Manufacturing process PRIMA selection matrix [5]

8.2 Surface texture related to common manufacturing processes

Surface texture is the local deviations of a surface from its ideal shape. The measure of the surface texture is generally determined in terms of its roughness, waviness and form [6].

<u>Roughness</u>: a quantitative measure of the process marks produced during the creation of the surface and other factors. This is usually the process marks or witness marks produced by the action of the cutting tool or machining process, but may include other factors such as the structure of the material [6].

<u>Waviness</u>: a longer wavelength variation in surface away from its basic form (e.g. straight line or arc). This is usually produced by instabilities in the machining process, such as an imbalance in a grinding wheel, or by deliberate actions in the machining

process. Waviness has a longer wavelength than roughness which is superimposed on the waviness [6].

Form: this is the general shape of the surface, ignoring variations due to roughness and waviness. Deviations from the desired form can be caused by many factors. For example [6]:

- the part being held too firmly or not firmly enough
- inaccuracies of slides or guideways of machines
- stress patterns in the component

Surface texture knowledge is important across a very wide spectrum of technical activities, from the design function to specification on a drawing, from the manufacturing process to verification. It has been recognized as being significant in many fields. In particular, surface texture is an important factor in production processes. It is used to monitor the production process, prevent any failure of the products and ensure surface quality. It is also one of the important factors to be considered when selecting the manufacturing processes.

The surface texture information is normally provided in the design through a specification callout, which includes surface texture lay, surface roughness value, surface waviness value, etc, see Chapter 5. The relationships between surface texture and manufacturing processes are discussed below:

Texture lay

Texture lay is the directionality of the surface, which is an important factor to reflect the interaction between the surface and the environment. Table 8.2 lists some examples of typical manufacturing processes appropriate to different texture lays [7].

Lay symbol	Interpretation	Typical Manufacturing processes
$=$ $\perp$	Parallel to plane of projection of view in which symbol is used Perpendicular to plane of projection of view in which symbol is used	milling, drilling, turning, shaping
X	Crossed in two oblique directions relative to plane of projection of view in which symbol is used	cross-honing
M	Multi-directional	lapping, abrading
C	Approx. circular relative to centre of surface to which symbol applies	facing, parting-off
R	Approx. radial relative to centre of surface to which symbol applies	face-grinding
P	Lay is particulate, non-directional, or protuberant	EDM, ECM, peening

Table 8.2 Texture lay with typical manufacturing processes [7]

Surface roughness values

A particular manufacturing process is capable of producing a limited range of surface roughness values [8]. Table 8.3 lists surface roughness values produced by common production processes. For example, normally the process drilling is used to produce the surface roughness value *Ra* between 1.6 µm – 6.3 µm, as highlighted in table 8.3 [9].

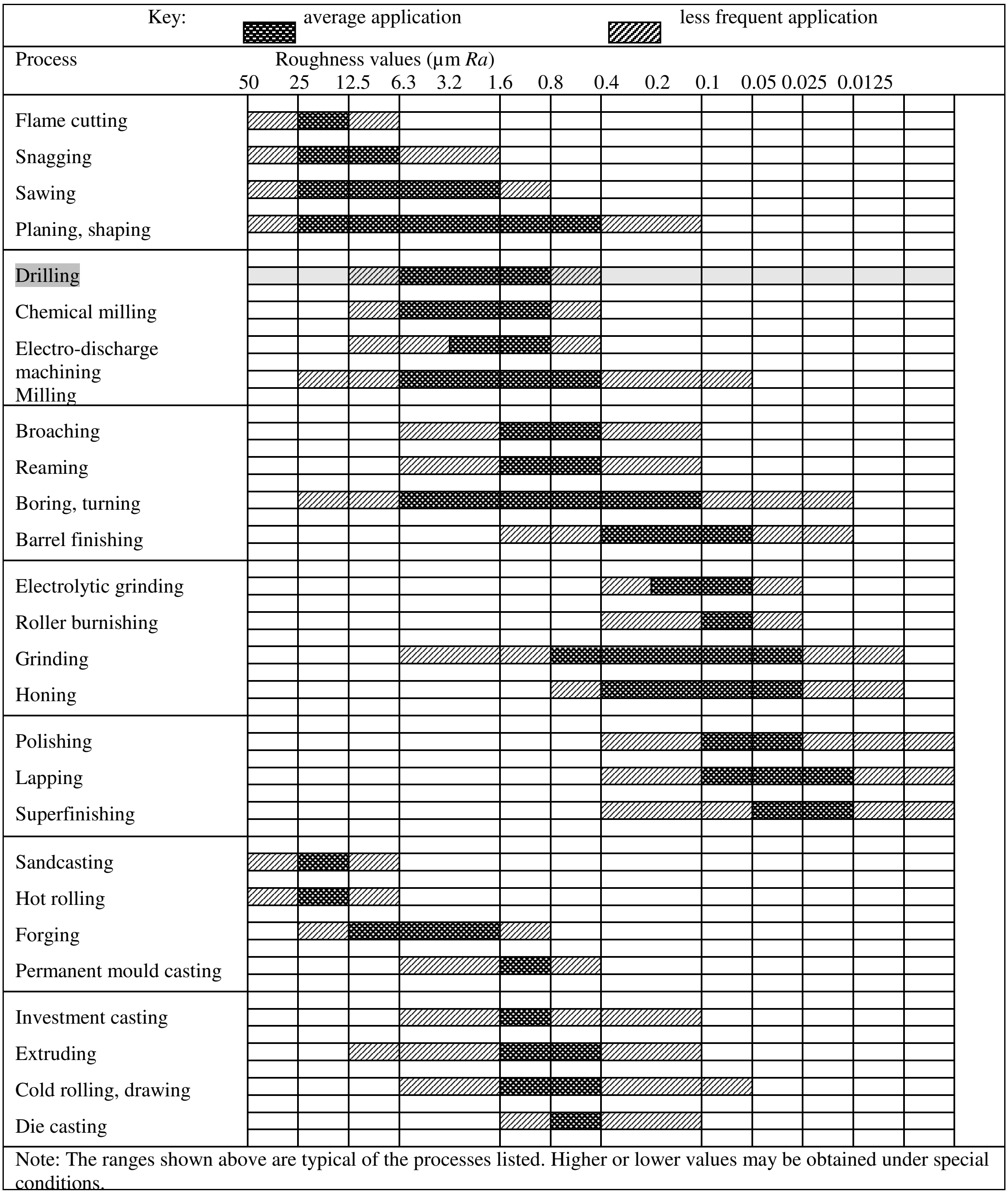


Table 8.3 Surface roughness values produced by common production processes [9]

Cut-off wavelength

The cut-off wavelength is used to distinguish between roughness values and waviness values. Table 8.4 gives the suitable cut-off wavelength for different processes which will be used in measurement of the finished products [10].

	Cut-off wavelength (mm)				
Process	**0.25**	**0.8**	**2.5**	**8.0**	**25.0**
Milling		√	√	√	
Turing		√	√		
Grinding	√	√	√		
Shaping		√	√	√	
Boring		√	√	√	
Planning			√	√	√
Reaming		√	√		
Broaching		√	√		
Diamond boring	√	√			
Diamond turning	√	√			
Honing	√	√			
Lapping	√	√			
Super finishing	√	√			
Buffing	√	√			
Polishing	√	√			
Electro discharge	√	√			
Burnishing		√	√		
Drawing		√	√		
Extruding		√	√		
Moulding		√	√		
Electro polishing		√	√		

Table 8.4 choice of cut-off wavelength for a number of common machining operations [10]

8.3 Manufacturing processes structure

After discussing manufacturing processes and their relations with surface texture, the manufacturing process selection procedure is developed in this thesis, as shown below:

As mentioned before, engineers normally select suitable procedures according to the process selection guide as shown in figure 8.1. Firstly, obtain relevant process selection inputs from the product design specification; Secondly, select candidate processes from the appropriate PRIMA selection matrix (see table 8.1) based on these inputs; and finally consider each PRIMA against technical and economic requirements. The selection process related to surface texture is summarised in the following lists:

1. Select the most relevant procedures using PRIMA selection matrix
2. Match the surface texture lay with the processes
3. Match the limit value with the processes
4. Match the cut-off wavelength with processes
5. Provide the suitable candidate manufacturing processes
6. Consider other factors listed in PRIMA

The material and the quantity in specification determine the candidate processes in PRIMA matrix, while the texture lay, the limit value and the cut-off wavelength are also considered. These factors finally determine the most appropriate candidate processes. Other main items in PRIMA forms of candidate processes will then be provided for the engineers to compare and make the final decision of the manufacturing process. The refinement structure of the manufacturing processes selection is shown in figure 8.3.

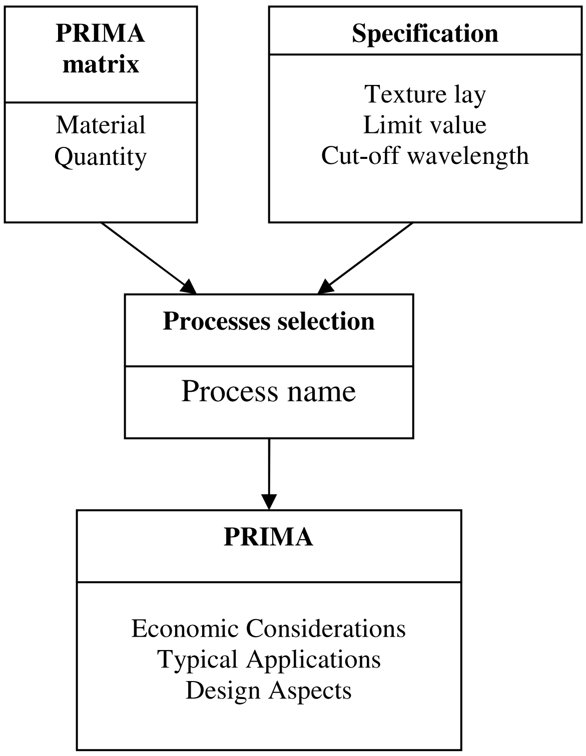


Figure 8.3 Structure of manufacturing processes

Vacuum forming [2.5]

Economic Considerations

- Process cycle times range from 10 to 60 seconds.
- Set-up times and change-over times are low.
- Production volume trends vary from small batches (10) to high volume, 1,000+.
- Tooling costs are low to moderate, depending on complexity.
- Equipment costs are low to moderate, but can be high it automated.
- Labour costs are low to moderate.
- Finishing costs are low.

Typical Applications

- Open plastic containers.
- Electronic enclosures.
- Bath tubs.

Design Aspects

- Shape complexity limited to mouldings in one plane.
- Open forms of constant thickness without re-entrant angles.
- Maximum section 3mm.
- Minimum section = 0.05mm to 0.5mm, depending on material used.
- Sizes range from 25mm2 to 7.5m x 2.5m in area.

Figure 8.4 gives an example of detailed PRIMA form which covers the technical and economic factors other than material and quantity of a process 'vacuum forming'. This extra information will be used to match the requirements and to compare with other candidate processes.

8.4 Category model to represent Manufacturing knowledge-base

Chapter 5 mentioned that the category model [11] is adopted to represent the knowledge in the manufacturing knowledge-base, which not only has the ability to represent this flowchart and has object-oriented capabilities to represent the detailed PRIMA forms.

Following the refinement process described above, the category model of manufacture knowledge-base is developed in figure 8.5. It includes five entities: Matrix, Criteria 1, Criteria 2, Criteria 3 and PRIMA, which are refined from figure 8.3. The entity Matrix is refined from PRIMA matrix, the Criteria 1 to 3 are refined from relations between three specification attributes and manufacturing processes, and finally PRIMA is refined from PRIMA form which output the detailed PRIMA form of candidate processes.

- The entity Matrix includes the information list in the table of manufacturing process PRIMA selection matrix, in which the names of candidate processes are determined by material types and quantities. The object c_name1 includes the first list of candidate processes.
- The entity Criteria 1 includes the information of the relation between the texture lay and manufacturing processes. The object c_name2 includes the second list of candidate processes, and the object direction is determined by the lay direction in entity Partition of specification knowledge-base, see table 8.2.
- The entity Criteria 2 includes the information of the relation between the cut-off wavelength and manufacturing processes. The object c_name3 includes the third list of candidate processes, and the object cut-off wavelength is determined by the up limit in entity Filtration of specification knowledge-base, see table 8.4.
- The entity Criteria 3 includes the information of the relation between the limit value and manufacturing processes. The object c_name4 includes the fourth list of candidate processes, and the object limit value is determined by the value in entity Measurand of specification knowledge-base, see table 8.3.
- The product of candidate processes c_name1 in Matrix, together with the processes c_name2 in Criteria 1, c_name3 in Criteria 2 and c_name 4 in Criteria 3 determines the final suggested processes in the entity PRIMA and also the standard PRIMA forms of suggested processes are provided for engineers to carry out the further comparison.

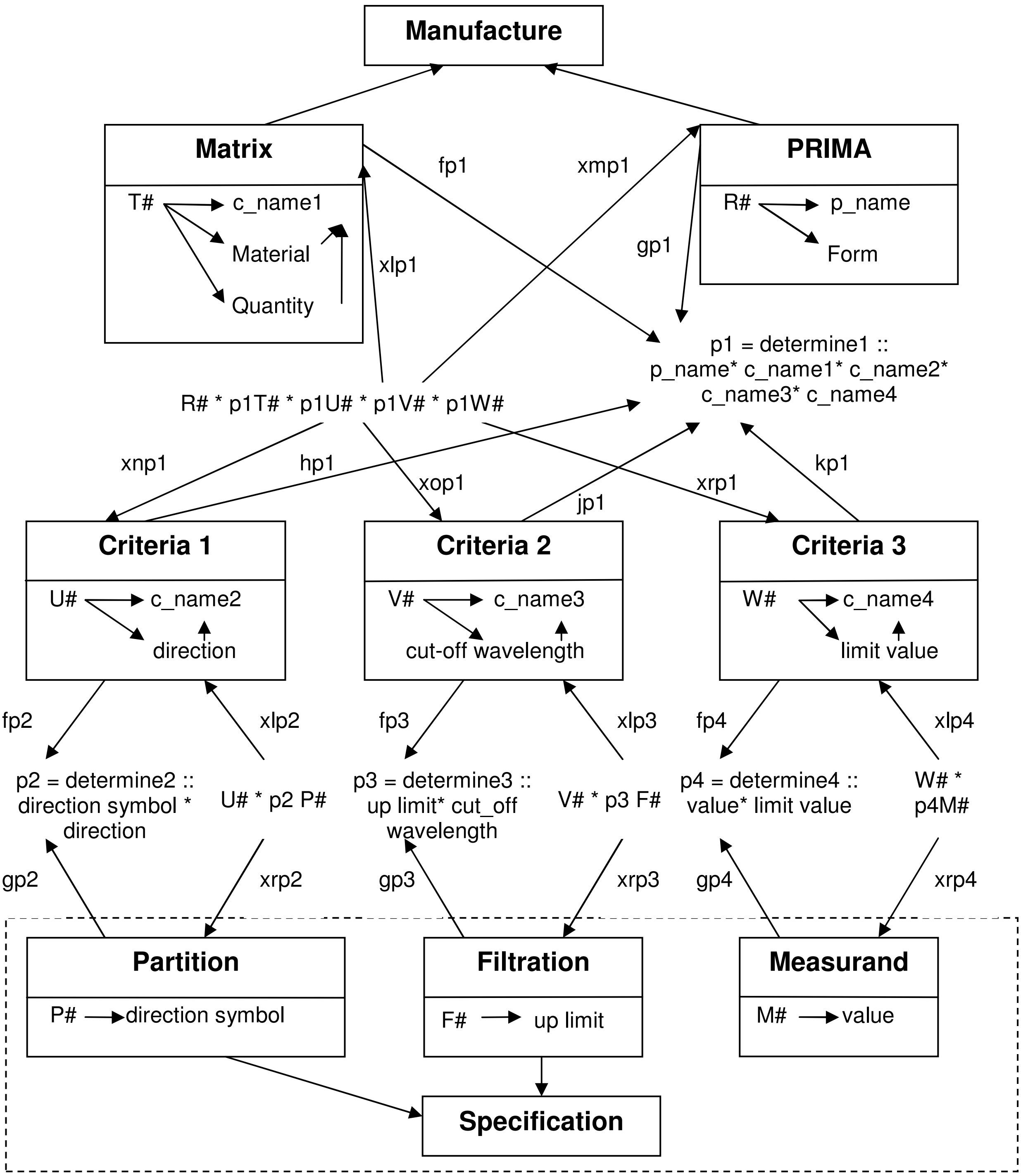


Figure 8.5 Category model of Manufacturing Knowledge-base

8.5 Interfaces of the Manufacturing knowledge-base

The sample interface of the manufacturing knowledge-base is shown in figure 8.6. Engineers could enter the requirements of the material type and the quantities; the default value of the texture lay, the cut-off wavelength and the limit value would be determined by the callout symbol in Specification knowledge-base, otherwise engineers can indicate the values they want.

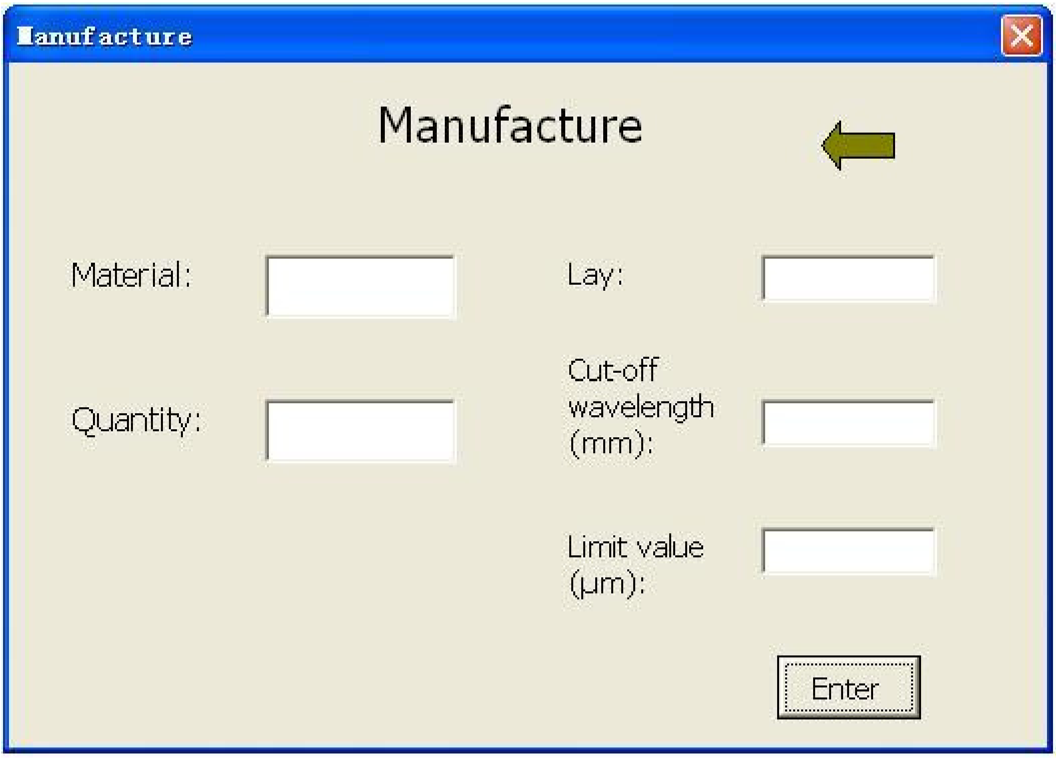


Figure 8.6 Interface of Manufacturing knowledge-base

Figure 8.7 shows the example of the output of the manufacture knowledge-base, which lists the appropriate manufacturing processes matching the requirements engineers entered. The output also gives the detail descriptions of each suggested manufacturing process in PRIMA form, engineers then could compare them and choose the most suitable one.

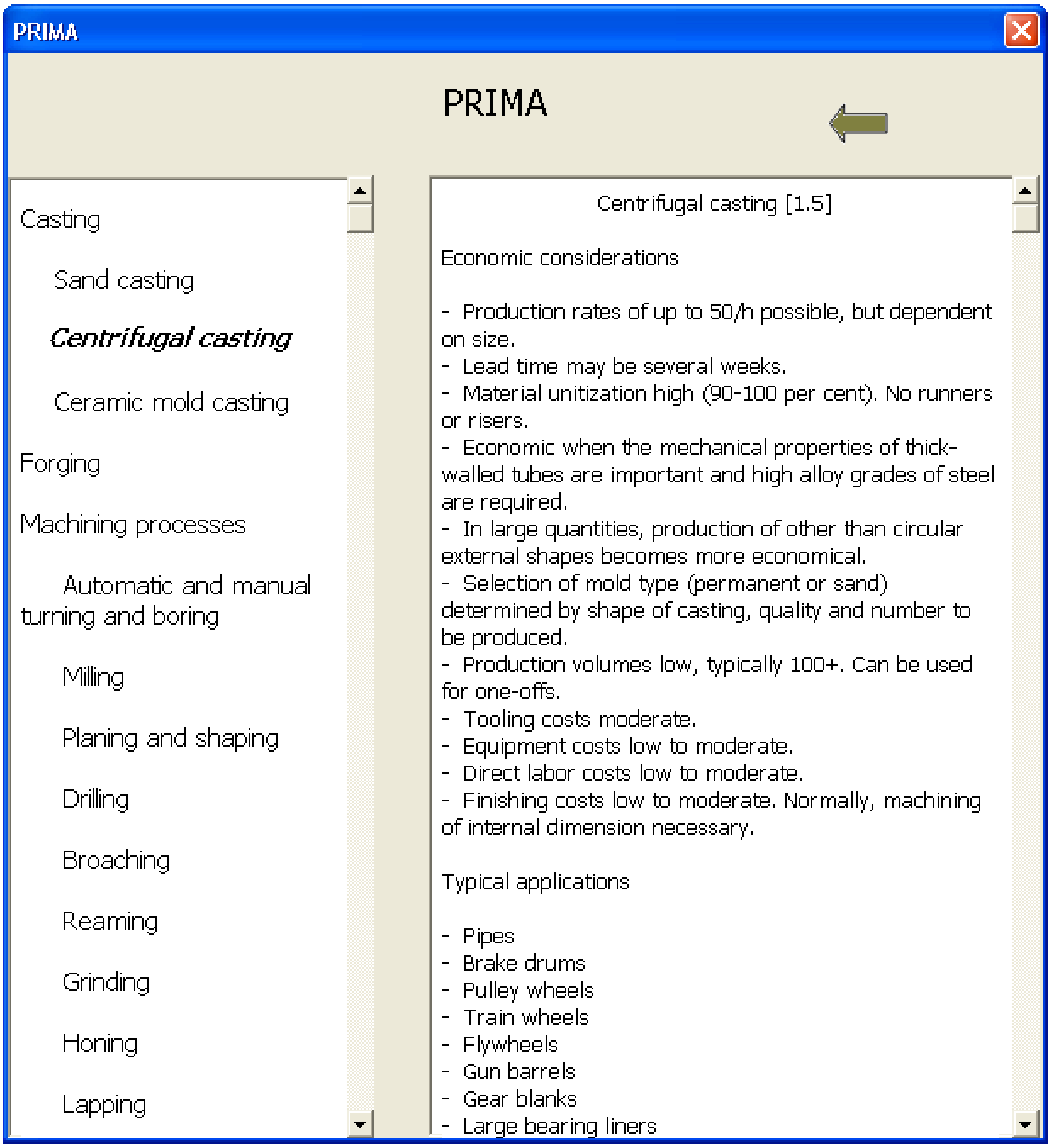


Figure 8.7 Output Interface of Manufacturing knowledge-base

8.6 Summary

This Chapter describes the detailed design procedures for manufacture knowledge-base, including:

Knowledge acquisition – The manufacture processes selection procedure is provided, and the relationship between manufacture processes and surface texture parameters is analysed.

Knowledge representation –The category model is applied to represent the knowledge in this knowledge-base, which not only has the ability to represent this flowchart and has object-oriented capabilities to represent the detailed PRIMA forms. Interfaces are given to further illustrate the implementation of the knowledge-base.

References

[1] Serope Kalpakjian & Steven R. Schmid (2001), *Manufacturing Engineering and Technology*, Prentice Hall, ISBN 0-201-36131-0

[2] http://en.wikipedia.org/wiki/Taxonomy_of_manufacturing_processes

[3] N. Hopkinson, R. Hague, P. Dickens (2005), *Rapid Manufacturing (Abstract)*, Germany: Wiley-VCH

[4] Paul K. Wright (2001), *21st Century manufacturing*, New Jersey: Prentice-Hall

[5] K.G. Swift & J.D. Booker (2003), *Process selection: from design to manufacture*, Elsevier Butterworth-Heinemann, ISBN 0750654376

[6] http://www.taylor-hobson.com/faqsurface.asp

[7] Brian Griffiths (2001), *Manufacturing Surface Technology – Surface Integrity & Functional Performance*, Penton Press, ISBN 1857180291

[8] Leigh Mummery (1990), *Surface Texture Analysis – The handbook*, Hommelwerke Gmbh

[9] BS 1134-2 (1990), *Assessment of surface texture — Part 2: Guidance and general information*

[10] Richard Leach (2001), *Measurement Good Practise Guide No.37 – The Measurement of Surface Texture using Stylus Instruments*, National Physical Laboratory, ISSN 1368-6550

[11] N.N Rossiter & D.A Nelson (1994), *The Categorical Product Data Model as Formalism for Object-Relational Databases*, Technical Report 505, Department of Computing Science, University of Newcastle, 1995

Chapter 9

Case Studies

9.1 Introduction

In this chapter, two case studies using the Virtualsurf system are presented. The first shows the application of metrology and design for surfaces of a total hip replacement system, and the second shows the application of designing a cylinder liner. The case studies will connect Function, Specification, Manufacture and Verification knowledge-bases together and provide examples of the implementation procedures for a Virtualsurf system.

9.2 Total hip replacement

Total hip replacement is the most common and well-established procedure in joint replacement [1]. For patients with painful and debilitating diseases such as osteoarthritis, rheumatoid arthritis and avascular necrosis, total joint replacement is an effective way of restoring and improving a patient's quality of life, often considerably.

Figure 9.1 shows a typical hip replacement system, which includes three components: acetabular, femoral head and femoral stem. During total hip replacement, the head of the femur is removed and replaced by a prosthetic ball (femoral head), this is typically secured in place by the insertion of a femoral stem into a cavity in the femur (medullary canal). The ball or femoral head articulates with a replacement acetabular cup, which is inserted into the affected acetabulum in the pelvis. A representation of the positioning of these components can be seen in figure 9.2 [1].

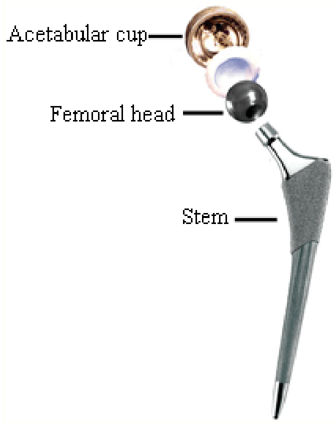


Figure 9.1 – Hip replacement system (http://www.hipreplacementinfo.com/hip/treatmentoptions/hipreplacement/hip_replacement_material_and_technology.cfm)

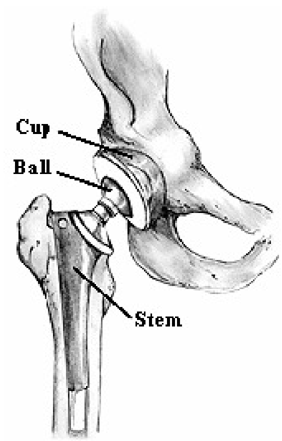


Figure 9.2 Schematic of the replacement hip joint in situ (www.hipsandknees.com)

9.2.1 Wear of total hip replacement

With the increase in life expectancy, and an increase in the number of younger patients requiring joint replacement, there is an emphasis on increasing the longevity of total hip replacement systems [1]. Aseptic loosening resulting from wear debris has been highlighted as a major factor in total hip replacement failure [2]. The current primary reason for aseptic loosening is bone resorption due to particulate debris being released during wear of the replacement components. Consequently the limitation of wear in hip implant components is critical to their long-term survival [2].

The primary cause of failure is wear at the bearing surface [3]. There are two main interfaces at which wear occurs [1]:

- The interface where the femoral head articulates against the acetabular cup

A number of researches have indicated that the wear rate of the polymer acetabular cup is greatly affected by the surface texture characteristics of the femoral head [4]. Figure 9.3 shows an example of severe wear of the femoral cup and head. With new advances this source is being reduced by employing novel materials which reduce the scope for wear particles such as ceramics and CoCr alloys with excellent bearing properties [1].

- At the stem/cement interface

  Debris at the stem cement interface is attributed to fretting wear due to micromovements of the stem in vivo [1]. Therefore, the surface finish of the stem interface plays an important role in the function performance of the stem. The assessment and analysis of the surface parameters have given a significant advantage to engineers and tribologists in gaining a deeper understanding of surface functionality.

Figure 9.3 – Femoral head and cup showing an area of severe wear

9.2.2 Links surface texture with functional performance

Surface texture can be utilised as a tool to identify modes of failure in prosthetic implants, and also as a comparative tool to identify function and success of bearing surfaces aiding the optimized design process [1]. In section 9.2.1 the effect of the surface texture characteristics on the functional performance of the system is discussed. Designed experimentation can help select relevant surface texture parameters for total hip replacement prostheses with a view to describing the topography of the bearing surfaces, and improving the functional performance. For example, in assessing the surface topography of femoral heads in relation to the wear of the acetabular cup, relevant roughness parameters would be derived. The methodology used would be to monitor for surface changes of the femoral heads looking for distinct worn regions visible by naked eye after revision surgery [6].

9.2.2.1 Instrument selection

One of the initial considerations for monitoring the surface changes is selecting a suitable instrument which satisfies the metrology requirements for the surface. The two primary concerns for use with orthopaedic implants are measurement resolution limitations and potential damage to the measurement specimen [1].

Firstly, consider the measurement resolution limitations. Guidelines are given in BS 7251:4, ISO 7206-2 [7], as to the appropriate surface roughness values to give good function in terms of the articulating surfaces (head and cup) [1]. The standard reports that the surface finish should be no greater than 70μm average roughness (Ra) for metal components and no greater than 30μm average roughness (Ra) for ceramic components. In reality, surface finishes far better than this are achieved due to advances in manufacturing techniques and material developments. The need for a highly polished surface finish on these components is due to the necessity for excellent bearing properties between the two surfaces through minimisation friction, hence reducing wear between the components [8]. Standards state that the surface finish of a ceramic femoral head must be below 30nm and up to 50nm for metal heads [1].

Suppose the surface roughness of femoral head to be measured is 30nm average roughness (Ra). According to the A-W diagram in figure 9.4, the coordinate of callout 'Ra 0.03' in the A-W plot is the point (0.5,0.03), 0.5μm is the sampling space and 0.03μm is the limit value, the instruments 'STM, AFM, SEM, Stylus, Focus' are capable of carrying out the required measurement.

Secondly, consider the potential damage to the measurement specimen. The instruments can be split into two broad categories, according to their measurement method: contacting and non-contacting. Stylus method belongs to the contacting method; while optical focus detection and SEM (scanning electron microscopy) belongs to the non-contacting method; and STM (scanning tunnelling microscopy) and AFM (atomic force microscopy) belong to a category of their own as they can not be completely described as contacting or non-contacting.

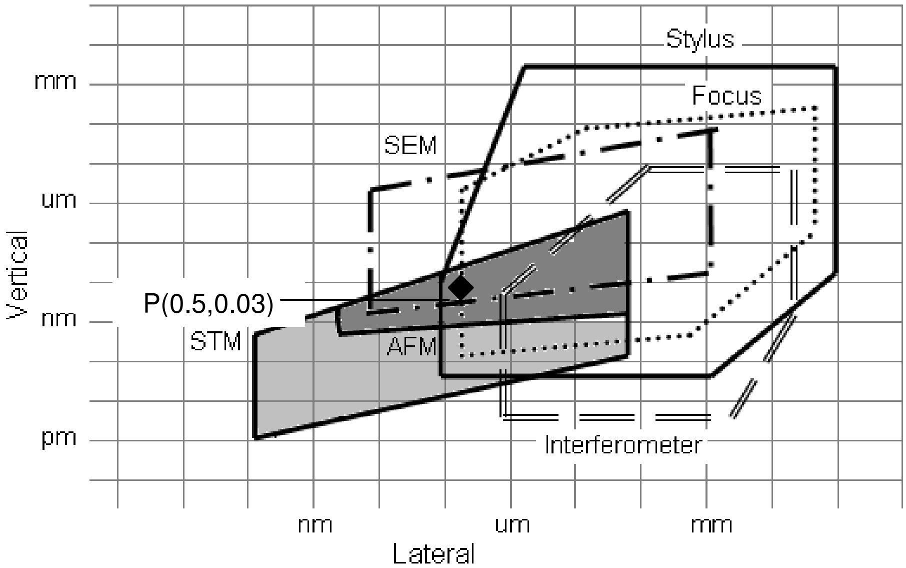


Figure 9.4 A-W diagram

The way Stylus instrumentation measures the surface topography may cause damage to the surface. It uses the stylus as the central component of the probe. The loads transmitted through the stylus tip would generate high pressure and damage to the surface. With the possibility of surface damage caused by the contacting method, the instruments suitable for the measurement are STM, AFM, SEM, and Focus detection.

9.2.2.2 Relevant surface roughness parameters

The second stage for monitoring the surface changes is carrying out the designed experiments and finding out the relevant surface roughness parameters in describing the topography of bearing surfaces.

After measuring worn regions of explanted femoral heads and calculating a range of surface parameters, amplitude parameters are proved to be the most relevant for characterising worn femoral heads. Table 9.1 lists three amplitude parameters calculated from femoral heads [2].

Heads	Ra (µm)	Rq (µm)	Rsk
No.1	0.0094	0.0129	0.1780
No.2	0.0053	0.0168	-15.572
No.3	0.0111	0.0178	-2.000
No.4	0.0035	0.0077	-10.017

Table 9.1 A comparison of surface roughness parameters calculation on femoral heads [2]

- **Ra** is the most relevant roughness parameter used to discriminate roughly the four scratched regions observed on retrieved femoral heads. However, Ra alone is less useful as a predictor of functional performance, as different tribological characteristics may have the same Ra value.
- Although **Rq** also measures the average height of roughness-component irregularities from the mean line, it uses a different formula and is more sensitive to peaks and valleys than Ra.
- **Rsk** can distinguish between asymmetrical profiles with the same Ra or Rq. Negative values indicate a predominance of valleys, while positive values indicate a predominance of peaks. For example, in table 9.1, region No.1 has more peaks while regions No.2, No.3 and No.4 are dominated by valleys.

9.2.3 Application of Virtualsurf

The Virtualsurf system would assist designers and metrologists to carry out the above procedures more efficiently. The aim for the measurement of total hip replacement system is monitoring surface changes and identifying the most relevant parameters with good functional prediction.

9.2.3.1 Flowchart of the implementation

Figure 9.5 shows the whole flowchart for the implementation. The function knowledge-base is first utilised to suggest procedures for the function assessment. After analysing functional performances of the surface, the proper specification parameters and values would be suggested. The specification knowledge-base is then applied to provide the essential information for a complete specification callout, and also mirrors the verification knowledge-base. The verification knowledge-base is invoked for the suggested specification, appropriate instruments are suggested for the measurement, together with the related measurement procedures. A range of surface parameters for the surface being measured will then be evaluated. Depending on the results of surface parameter values, the most relevant parameters with good functional prediction would be determined. Finally, return to the functional knowledge-base, new examples of relationships between parameters and functions would be added into the function knowledge-base.

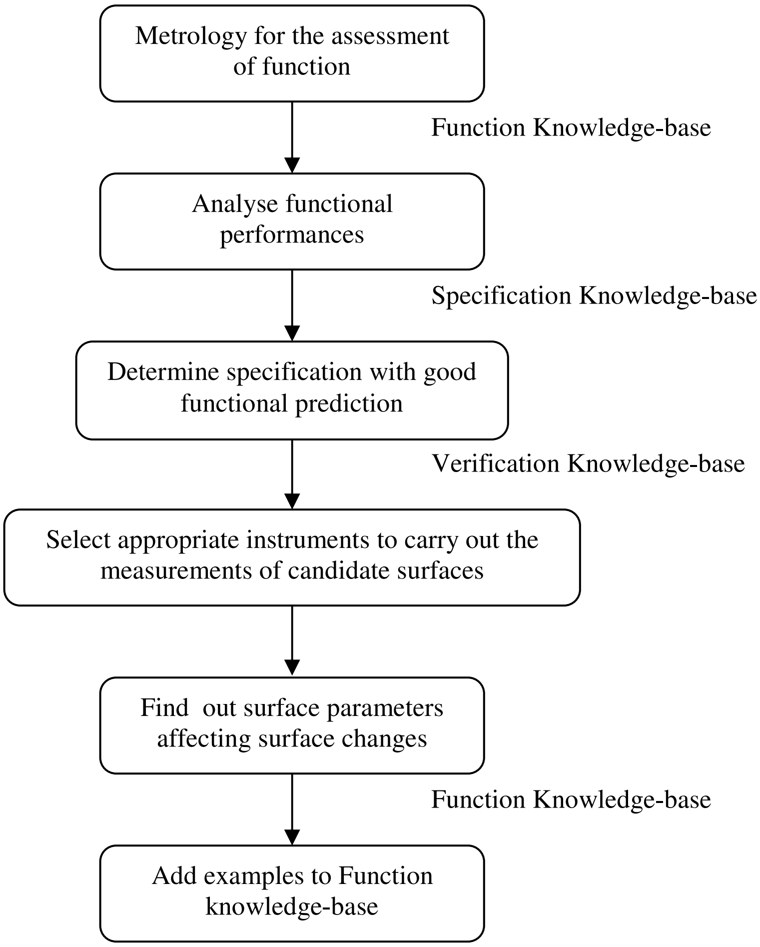


Figure 9.5 Flowchart of metrology procedures

9.2.3.2 Interfaces of of the Virtualsurf system

Figure 9.6 and 9.7 show the user-friendly interfaces provided for both designers and metrologists: the interface of Virtualsurf system and the interface of procedures for metrologists.

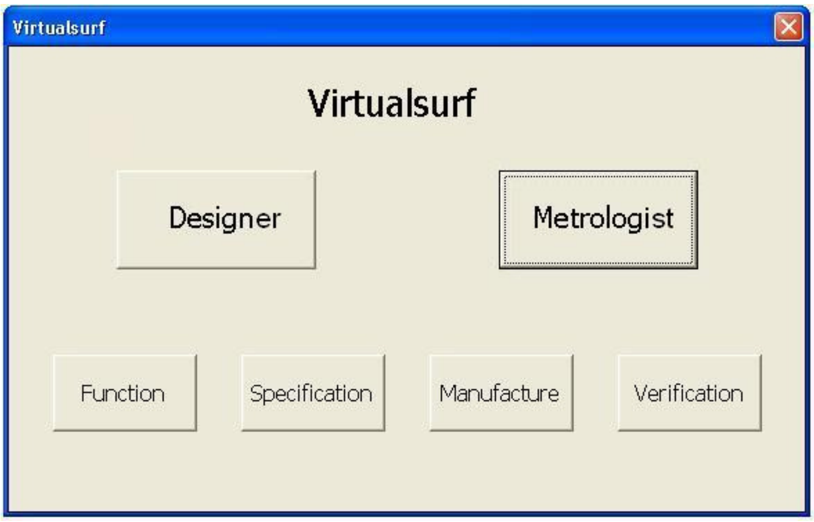


Figure 9.6 Interface of "Virtualsurf"

On the main interface metrologists click the "Metrologist" button and then go through into the Metrologist interface. There are three options available on the Metrologist interface, first click the "Function" button and then go through into the Function knowledge-base.

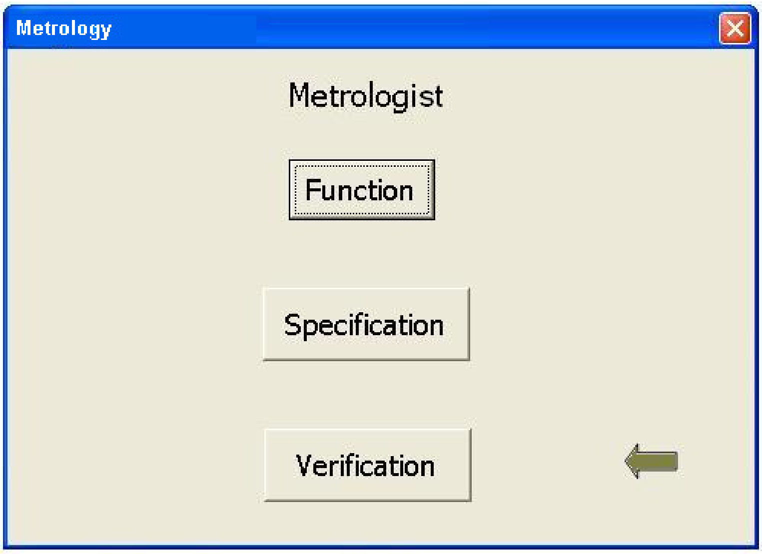


Figure 9.7 Interface of procedures for metrologists

Function knowledge-base

Figure 9.8 shows the interface of Function knowledge-base, which is followed with patterns below:

Pattern 1 — Surface requirements

Example: For the interface where the femoral head articulates against the acetabular, it requires a good bearing surface.

Pattern 2 — Functional performance

Example: The nature of bearing surfaces on femoral heads requires minimization of friction and maximization of lubrication conditions.

Pattern 3 — Surface parameters selection

Example: In order to find relations between relevant parameters with surface functions of femoral heads, it is better to establish a stable surface generation process that produces acceptable surfaces and then Pattern 5 — Monitor for surface changes. Standards BS 7251:4, ISO 7206-2 [7] state that the surface finish of a ceramic femoral head must be below 30nm and up to 50nm for metal heads.

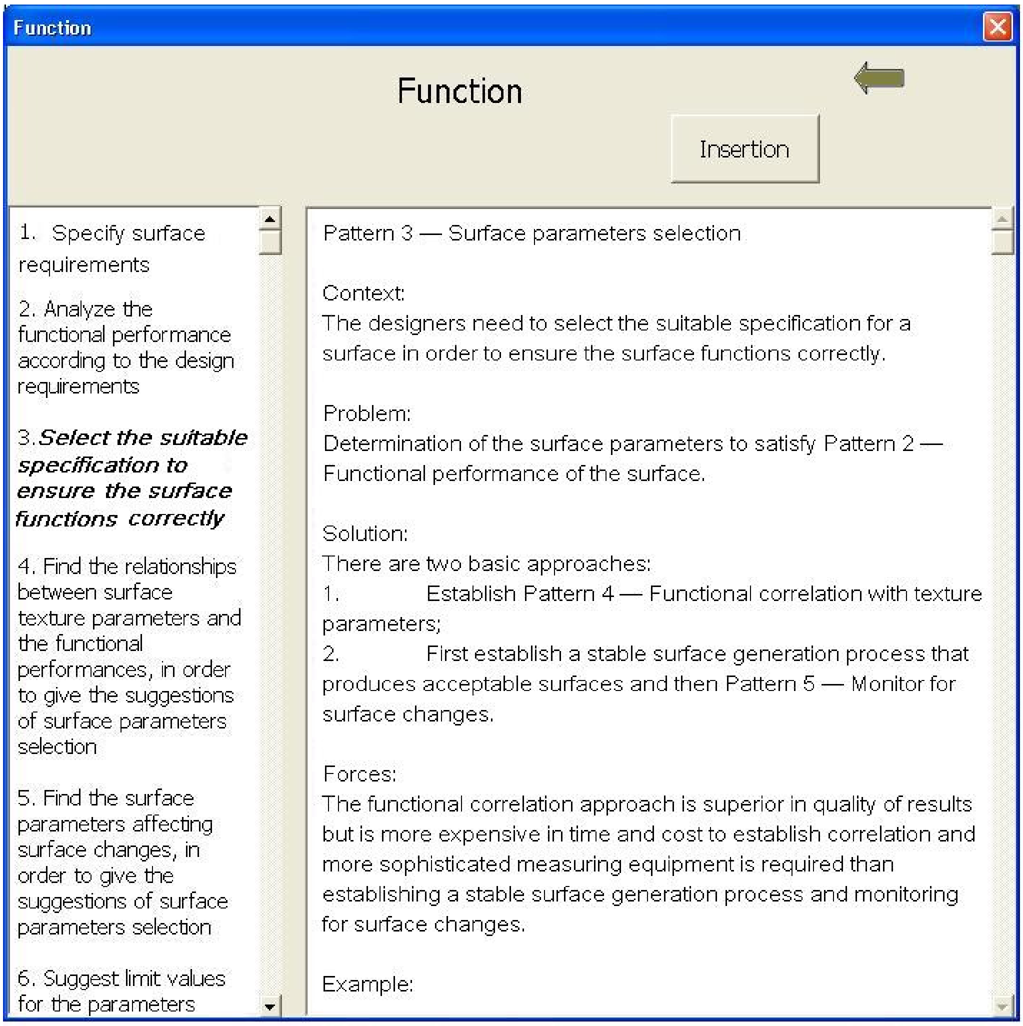


Figure 9.8 Interface of Function knowledge-base

The function knowledge-base suggests that metrologists should monitor surface changes of femoral heads showing distinct worn regions visible at naked eye after revision surgery, and also indicate the surface finish (e.g. Ra) of these femoral heads are within the tolerance of 30nm.

Specification knowledge-base

The second step is to implement specification knowledge-base, which would give the complete callout for the surface specification, and then the verification knowledge-base mirror this. Figure 9.9 shows the interface of Specification knowledge-base and figure 9.10 gives the output of specification knowledge-base for the femoral head. The indicated surface finish in figure 9.9 is Ra = 0.03µm, the same limit value to other surface parameters for femoral heads, as standards BS 7251:4, ISO 7206-2 [7] state that the surface finish of a ceramic femoral head must be below 30nm.

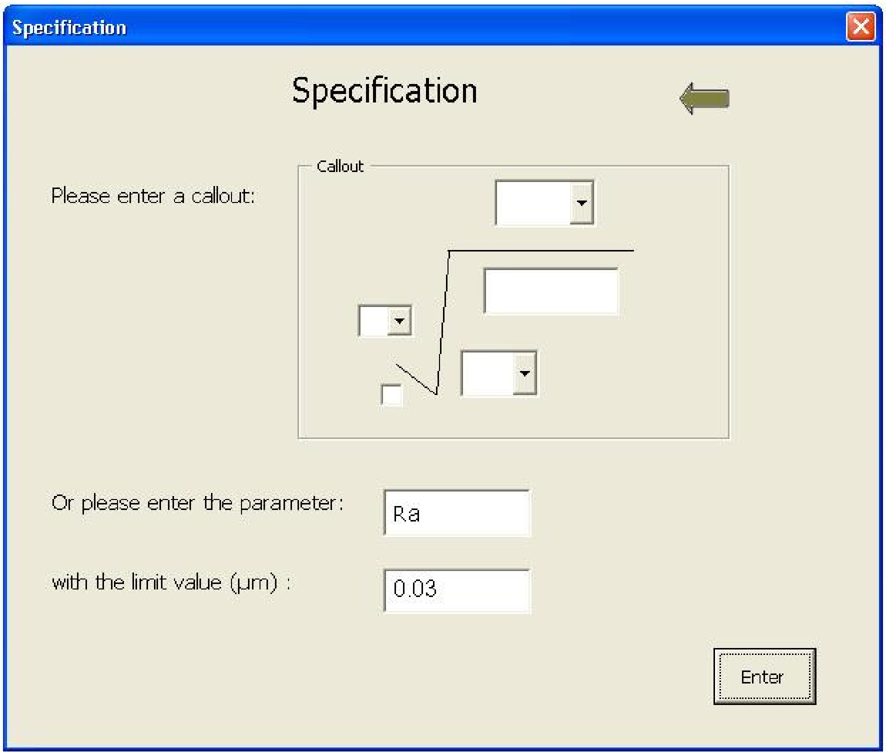


Figure 9.9 Interface of Specification knowledge-base

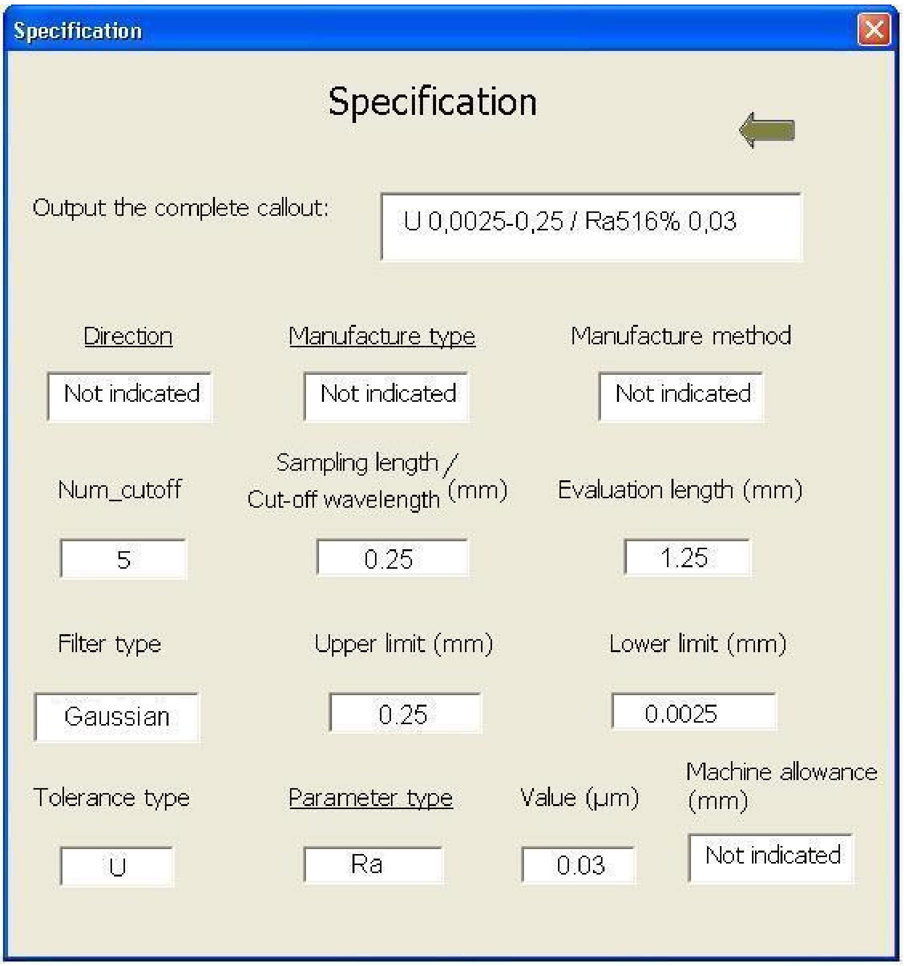


Figure 9.10 Output of Specification knowledge-base

Verification knowledge-base

Depending on the specification indicated in the specification knowledge-base, verification knowledge-base would help metrologists select suitable measurement processes.

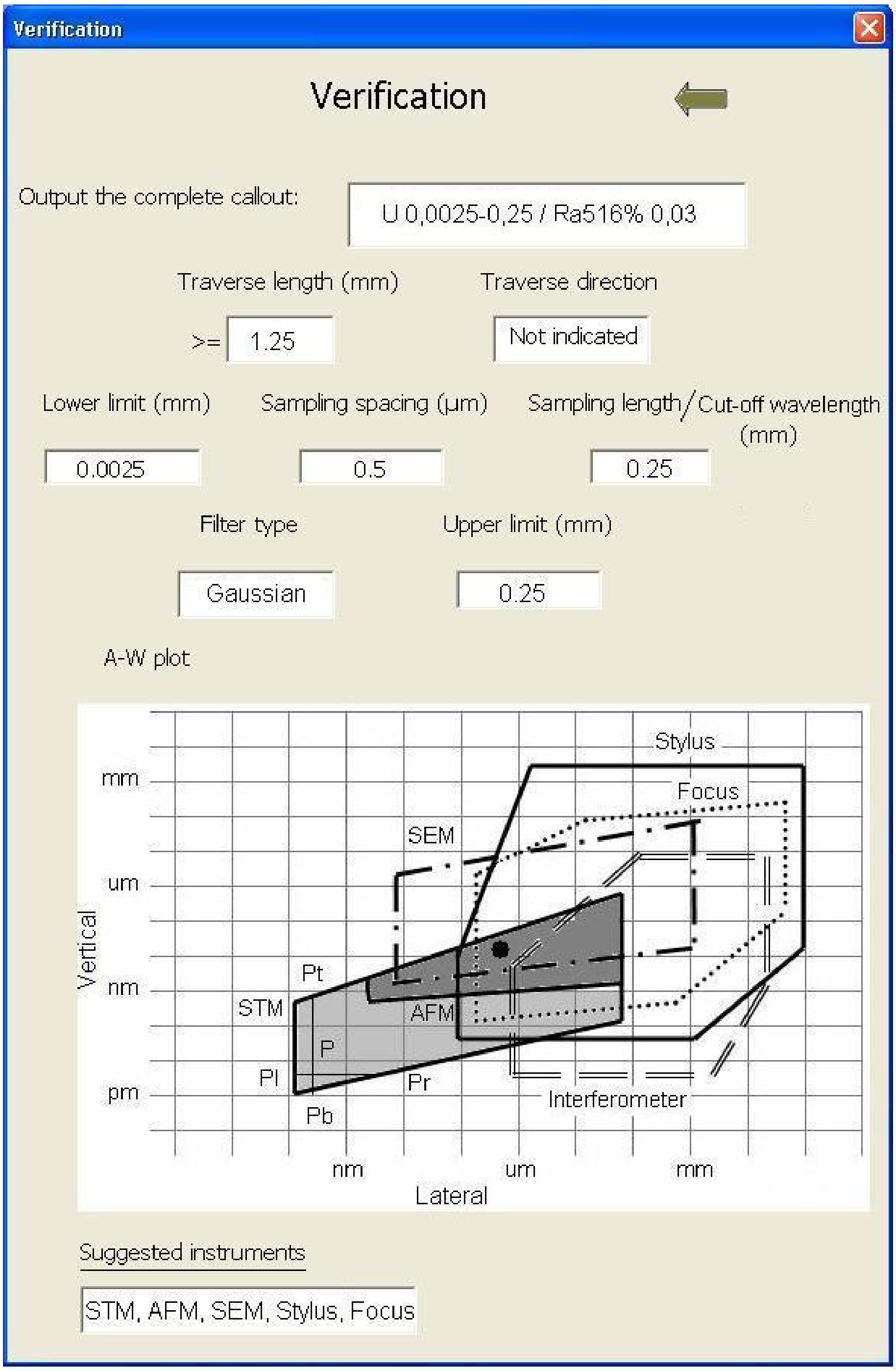
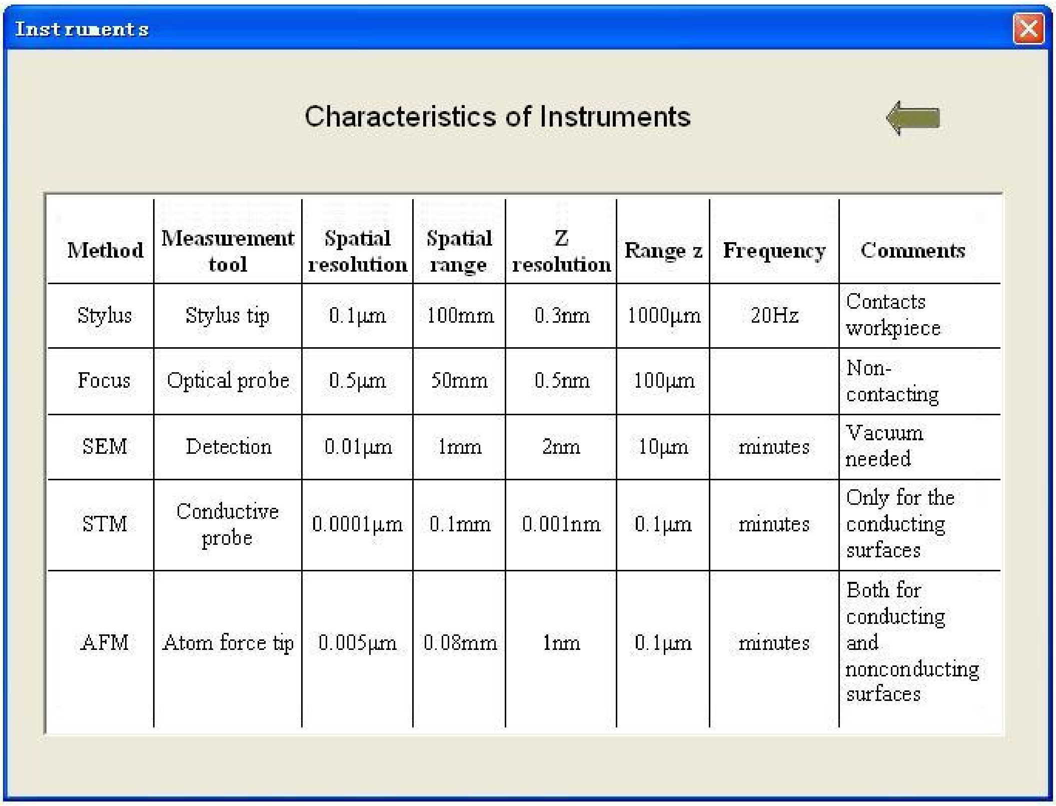


Figure 9.11 Output of the Verification knowledge-base

Candidate measuring instruments would be suggested by the A-W (amplitude-wavelength) plot (see section 7.2.5.1), according to the limit value and sampling spacing of the specified surface, see figure 9.11, the limit value for the femoral head is indicated as 0.03μm and the sampling spacing is 0.5μm derived from ISO 3274 [10]. The black dot ● in the A-W plot represents the parameter Ra 0.03. The polygons which cover the dot would be able to measure this parameter. Five instruments are eligible at this stage, which are STM, AFM, SEM, Stylus and Focus detection, then the table showing the detail characteristics would be provided. Metrologists could compare these instruments and choose the most suitable one from them, see figure 9.12.



Method	Measurement tool	Spatial resolution	Spatial range	Z resolution	Range z	Frequency	Comments
Stylus	Stylus tip	0.1μm	100mm	0.3nm	1000μm	20Hz	Contacts workpiece
Focus	Optical probe	0.5μm	50mm	0.5nm	100μm		Non-contacting
SEM	Detection	0.01μm	1mm	2nm	10μm	minutes	Vacuum needed
STM	Conductive probe	0.0001μm	0.1mm	0.001nm	0.1μm	minutes	Only for the conducting surfaces
AFM	Atom force tip	0.005μm	0.08mm	1nm	0.1μm	minutes	Both for conducting and nonconducting surfaces

Figure 9.12 Characteristics of Suggested Instruments

Insertion new examples into Function knowledge-base

With the help of Virtualsurf knowledge-based system, metrologists can easily find out the procedure to carry out the metrology of femoral heads. After measuring worn regions of the explanted femoral heads and calculating a range of surface parameters on them, amplitude parameters are concluded to be most relevant for characterising worn femoral heads [2]. In particular, parameters Ra, Rq and Rsk are useful for the assessment of function of used implants, see section 9.2.2.2. This information would be added into the function knowledge-base. Figure 9.13 shows the interface of inserting new examples into Pattern 5 – Monitor for surface changes.

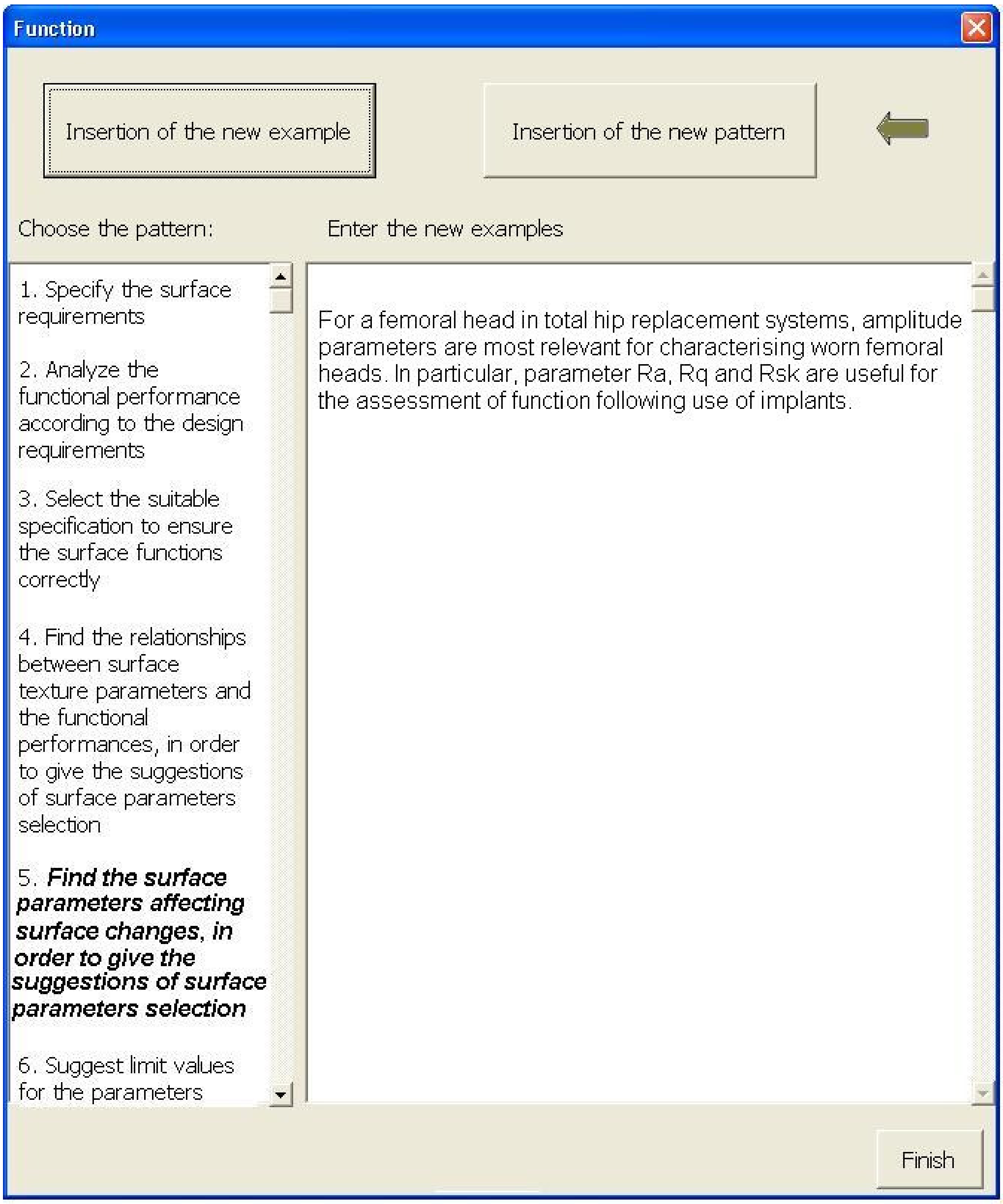


Figure 9.13 Interface of the insertion of a new example

9.3 Design of a cylinder liner

A cylinder liner is in the central working part of a reciprocating engine, and it is the space in which a piston travels. Figure 9.14 shows a typical cylinder liner with a piston of a steam engine [11]. The movement of a piston inside the cylinder makes the vehicle move.

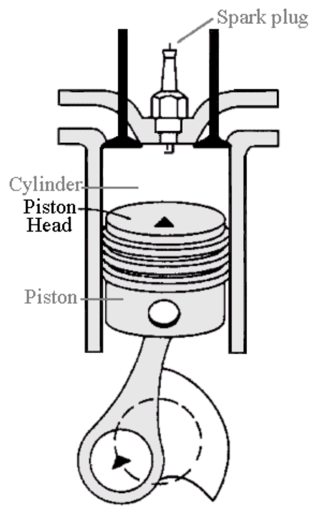


Figure 9.14 Cylinder liner with the piston

A piston is moving inside each cylinder with several metal piston rings which fit around its outside surface in machined grooves; typically two for compressional sealing and one to seal the oil, see figure 9.15. They are made of spring steel and make near contact with the hard walls of the cylinder bore, riding on a thin layer of lubricating oil which is essential to keep the engine from seizing up [12].

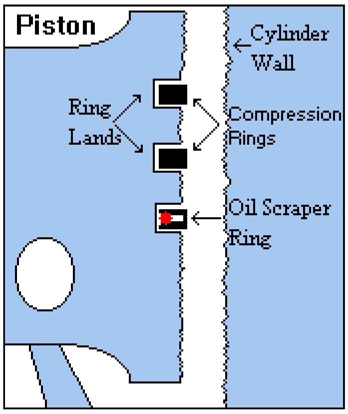


Figure 9.15 Piston rings

The contact between the cylinder liner and its counterpart piston rings, requires that the cylinder needs to have a good bearing surface but also retain a reservoir of oil for lubrication. Moreover, the space surrounded by the cylinder bore and piston rings needs to be a good seal to keep the compression of fuel and air mixture [13].

9.3.1 Functional performance

The most important functional demands of the cylinder and piston rings are oil consumption, blow-by, and low wear specially at the TDC (top-dead centre) [13].

Oil consumption: It is better if a cylinder can save oil. Oil consumption depends primarily on the piston rings. If the piston rings are worn, cracked, missing, broken or improperly installed, the engine will suck oil down the guides and into the cylinders. The engine may not have good compression, and will use a lot of oil [14].

Blow-by: Compressed fuel and air mixture burns in the cylinder at the top of the pistons. The crankcase, that which contains the crankshaft and connecting rods, is at the bottom of the engine. When the rings become tired and worn they allow some of this compressed and burning mixture to leak past and escape into the crankcase. This is called "blow-by" [15]. Blow-by inhibits performance because it results in a loss of compression. Less blow-by means less contamination, better fuel economy, and more power. The major factor that causes blow-by is wear: As rings and cylinder liners wear away they are less capable of maintaining this seal.

Wear: Both the cylinder and piston rings are subject to wear as piston rings rub up and down the cylinder bore.

9.3.2 Relation between surface parameters and functional performance

The surface texture has a direct influence on the functional performance of the part. In order to investigate the correlation between the surface parameters and the function of the part, a factorial designed experiment (FDE) was performed where surface roughness was correlated to functional performance indicators such as oil consumption, wear, and blow-by in a 10 litre truck engine [13] (Investigated by Volvo truck corporation).

Table 9.2 lists the parameters and their values chosen in the experiment, including the roughness of the piston ring Ra, the amplitude parameter of the cylinder liner Rz and the reduced value to core ratio Rvk/Rk of the cylinder liner. The measured responses in these experiments were the functional performance indicators:

A: Oil consumption, which was measured as the mean consumption for different loads and revolutions (g/kWh).

B: Blow-by, which measured as the gas flow that passes the ring pack (l/s).

C: Wear, which was measured after the test at TDC (top dead centre) of the cylinders. The wear was characterised by the maximum wear depth in micrometers.

Test no	Piston Ring Ra (µm)	Cylinder Liner Rz (µm)	Rvk/Rk
1	0,5-0,6	16	3
2	<0,2	4	3
3	0,3-0,4	10	1,5
4	0,5-0,6	4	1
5	0,3-0,4	10	1,5
6	<0,2	16	1
7	0,3-0,4	10	1,5

Table 9.2 The test plan used in the FDE [13]

Values for the correlation factors between the surface roughness parameters and the functional performance are shown in table 9.3. Values between 0,3 and -0,3 considered as not significant and are shadowed in the table. The sign relates to the variation of the parameter and the response. When the sign is positive, the parameter and the response have the same variation. When the sign is negative, the variation is contrary. For example, when Rz of the cylinder increased, oil consumption increased since they have the same variation [13].

	Ra	Rz	Rvk/Rk	A	B	C
Ra	1					
Rz	-0.1337	1				
Rvk/Rk	0.174711	-0.209439	1			
A	-0.240277	0.965298	-0.355595	1		
B	-0.838345	0.244934	-0.244315	0.321418	1	
C	0.625148	0.583006	-0.308528	0.481782	-0.397573	1

Table 9.3 The correlation factor between the surface roughness parameter value used and the functional performance, where A = oil consumption, B = blow-by, and C = wear [13]

According to results in the table 9.3, oil consumption (A) is strongly correlated to Rz measured on the cylinder liner. The biggest influence on blow-by (B) is Ra measured on the piston rings and shows a negative variation. The wear (C) is strongly correlated to the Ra measured on the piston rings followed by Rz measured on the cylinder; both have the same variation with the wear. It is possible to determine the relationships between the functional performance and the surface texture parameters by applying this correlation approach to the cylinder part, and by analysing the results of the correlation coefficient as listed in table 9.3.

9.3.3 Application of Virtualsurf

The above sections described approaches to obtain the key surface texture parameters for the cylinder and piston rings. The Virtualsurf system would help engineers design a cylinder liner more quickly and easily. The detailed procedures are explained in the following sections. Figure 9.16 and figure 9.17 show the user-friendly interfaces provided for engineers: interface of Virtualsurf system and interface of procedures for designers.

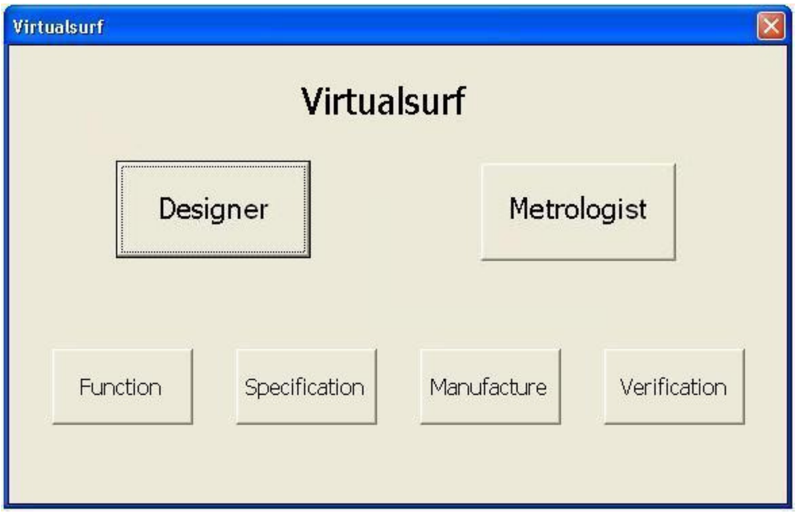


Figure 9.16 Interface of "Virtualsurf"

At the beginning, on the main interface, engineers click "Designer" button and then go through into the Designer interface. There are four steps available for the procedures, beginning with the Function knowledge-base.

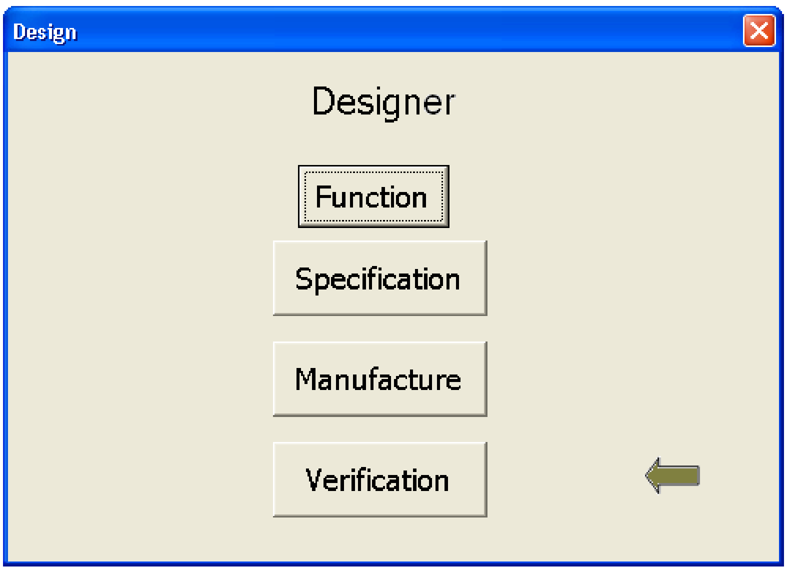


Figure 9.17 Interface of procedures for designers

Function Knowledge-base

The Function knowledge-base would suggest to engineers those surface parameters matching with functional performances (see Chapter 7), see figure 9.18, followed by the patterns below:

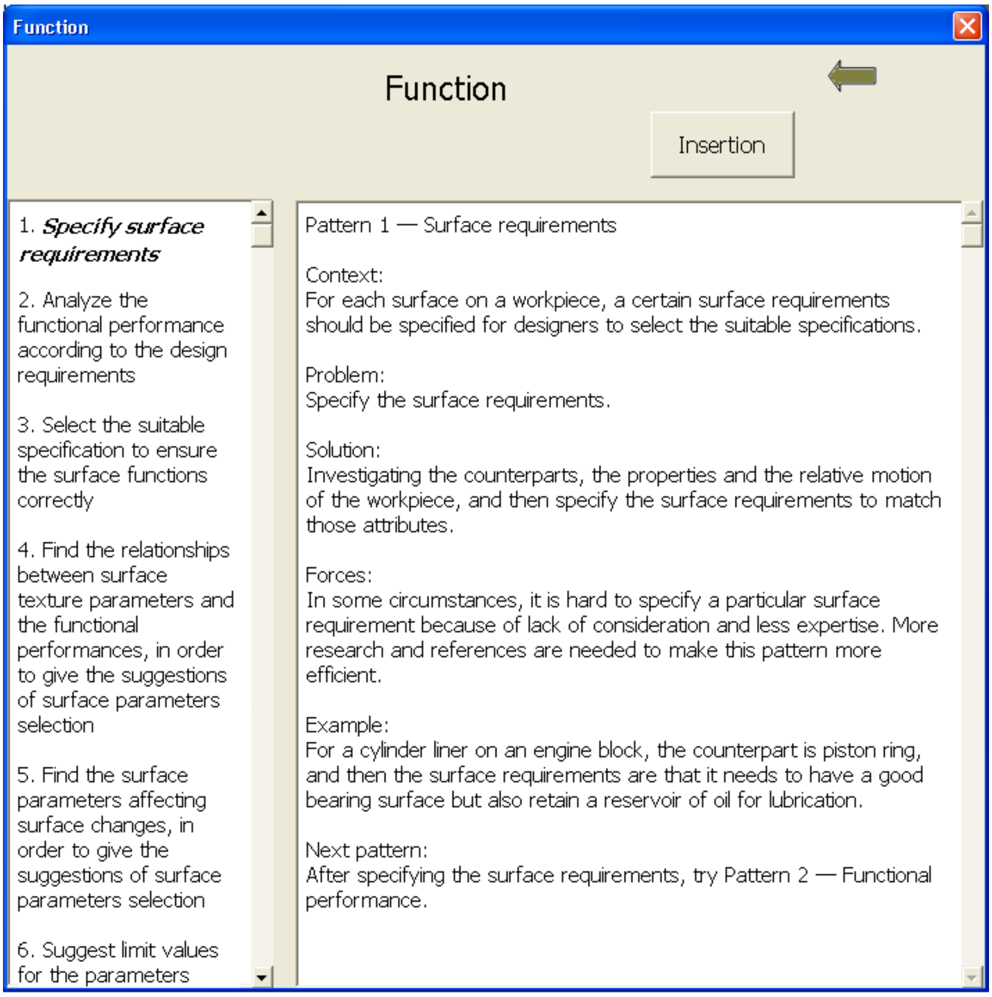


Figure 9.18 Interface of Function knowledge-base

Pattern 1 — Surface requirements

Example: For a cylinder liner on an engine block and its counterpart piston ring, the surface requirements are having a good bearing surface but also retaining a reservoir of oil for lubrication [13].

Pattern 2 — Functional performance

Example: The surface requirements for a cylinder liner on an engine block are having a good bearing surface but also retaining a reservoir of oil for lubrication. Therefore, the most important functional demands are oil consumption, blow-by, and low wear specially at the TDC (top-dead centre).

Pattern 3 — Surface parameters selection

Example: The surface requirements for a cylinder liner on an engine block are having a good bearing surface but also retaining a reservoir of oil for lubrication. The texture parameters Rk and its family have been shown to have a functional correlation with the desired surface tasks.

Pattern 4 — Functional correlation

Example: The surface requirements for a cylinder liner on an engine block are having a good bearing surface but also retaining a reservoir of oil for lubrication. The texture parameters Rk and Rz have been shown to have a functional correlation with the desired surface tasks.

Pattern 6 — Suggestion of limit values

Example: According to the factorial designed experiment (FDE), when Rz of the cylinder increased, oil consumption, blow-up and low wear all increased since they have the same variation. Depending on the results of experiments, the limit value of Rz is suggested to use 4 µm. Therefore, the suggested surface parameter for the cylinder liner is Rz with a tolerance of 4µm.

Specification Knowledge-base

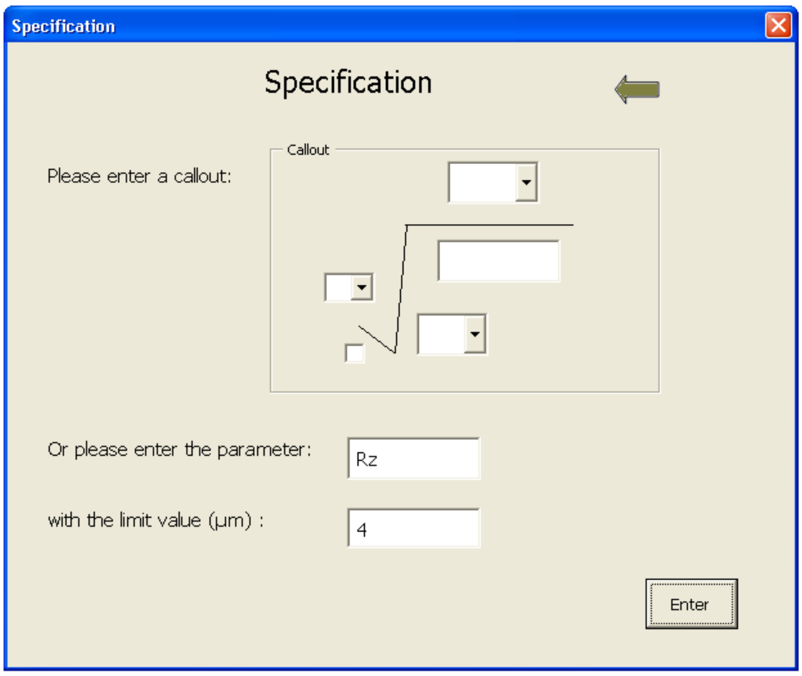


Figure 9.19 Interface of Specification knowledge-base

The second design step is the specification knowledge-base, which would give the complete callout for the surface, including the sampling length, evaluation length, bandwidth for the filter, etc (see Chapter 5). Figure 9.19 shows the interface of Specification knowledge-base. Experienced engineers can also input information such as surface texture lay, manufacture methods, etc into the callout. This example only provides the surface texture parameter with its limit value. And figure 9.20 gives the output of specification knowledge-base for the cylinder liner.

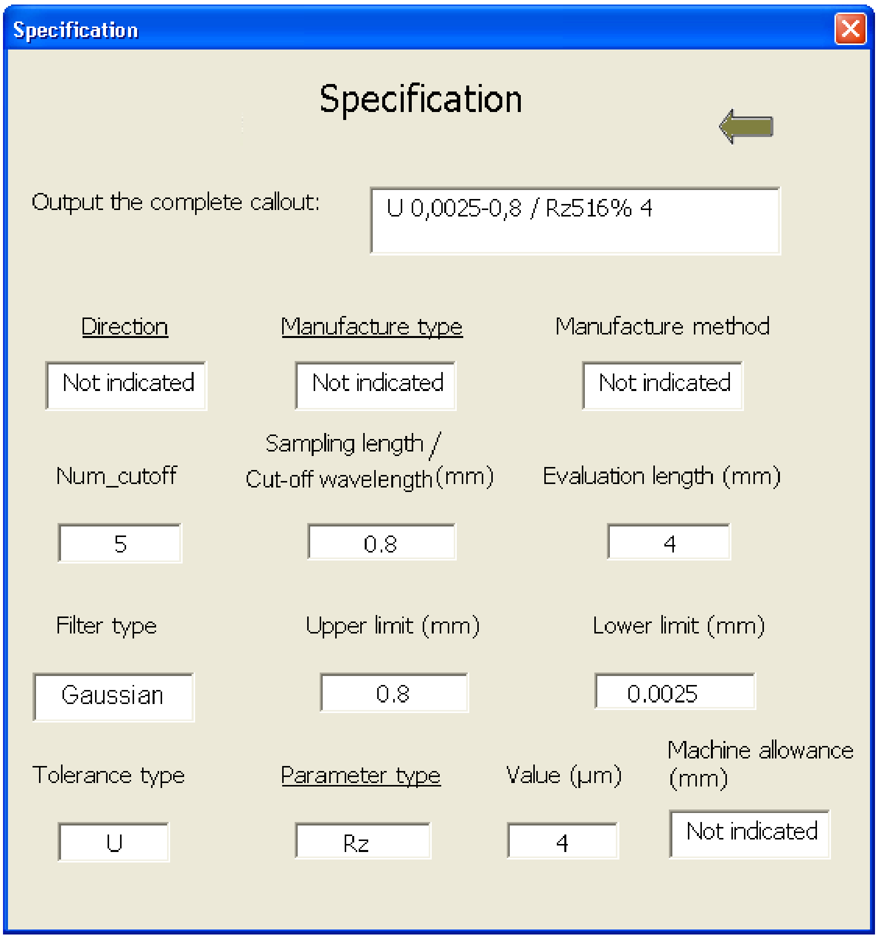


Figure 9.20 Output of specification knowledge-base

Manufacturing Knowledge-base

The third design step is manufacturing knowledge-base, which suggests appropriate manufacture processes satisfying surface requirements (see Chapter 8). Figure 9.21 shows the interface of Manufacture knowledge-base. Engineers could enter the requirements of material, quantity, texture lay and limit value on the interface. For this cylinder liner, the material chosen is steel and the limit value is 4 µm according to the specification knowledge-base.

Figure 9.22 shows the output of Manufacture knowledge-base, which lists the appropriate manufacturing processes matching the requirements engineers entered. For a cylinder liner, the suggested processes are casting and machining processes. The output also gives detailed descriptions of each suggested manufacturing process in PRIMA form, engineers then could compare them and choose the most suitable one.

Figure 9.21 Interface of Manufacture knowledge-base

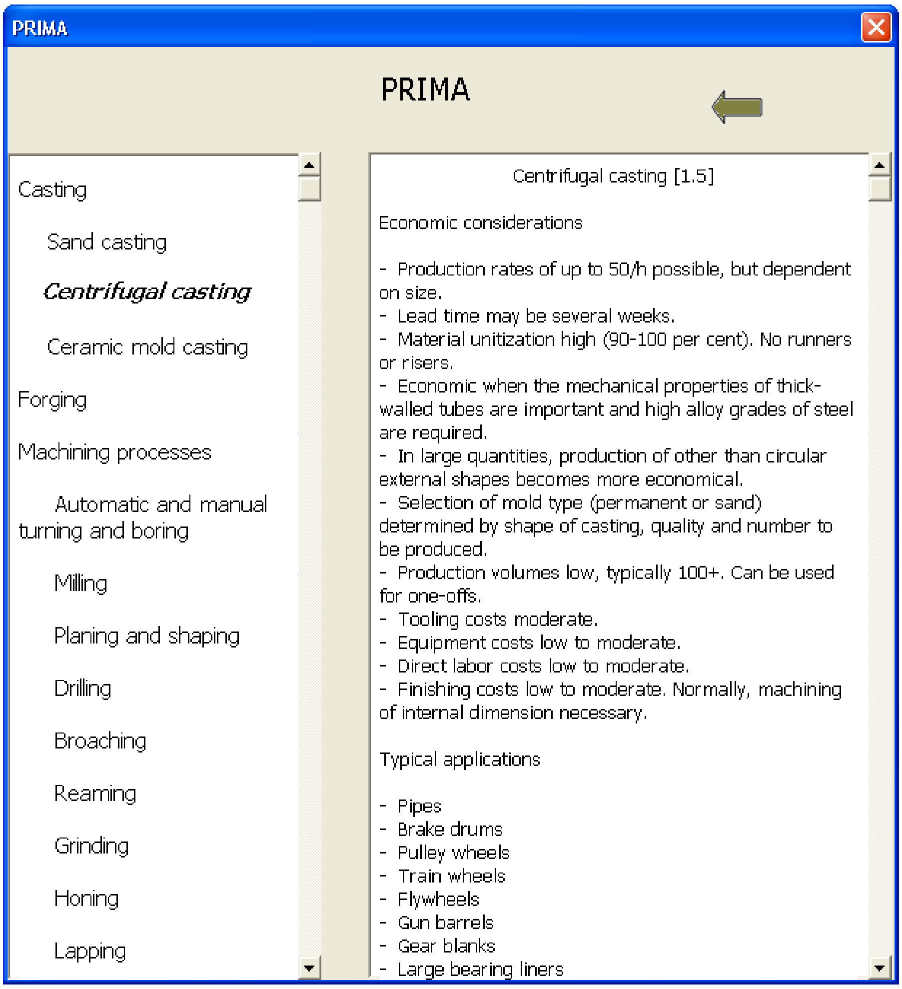


Figure 9.22 Output of Manufacture knowledge-base

Verification Knowledge-base

The final design step is verification knowledge-base, which suggest the suitable measurement processes for the cylinder liner, including the traverse length, the sampling space, measuring instruments, etc (see Chapter 6).

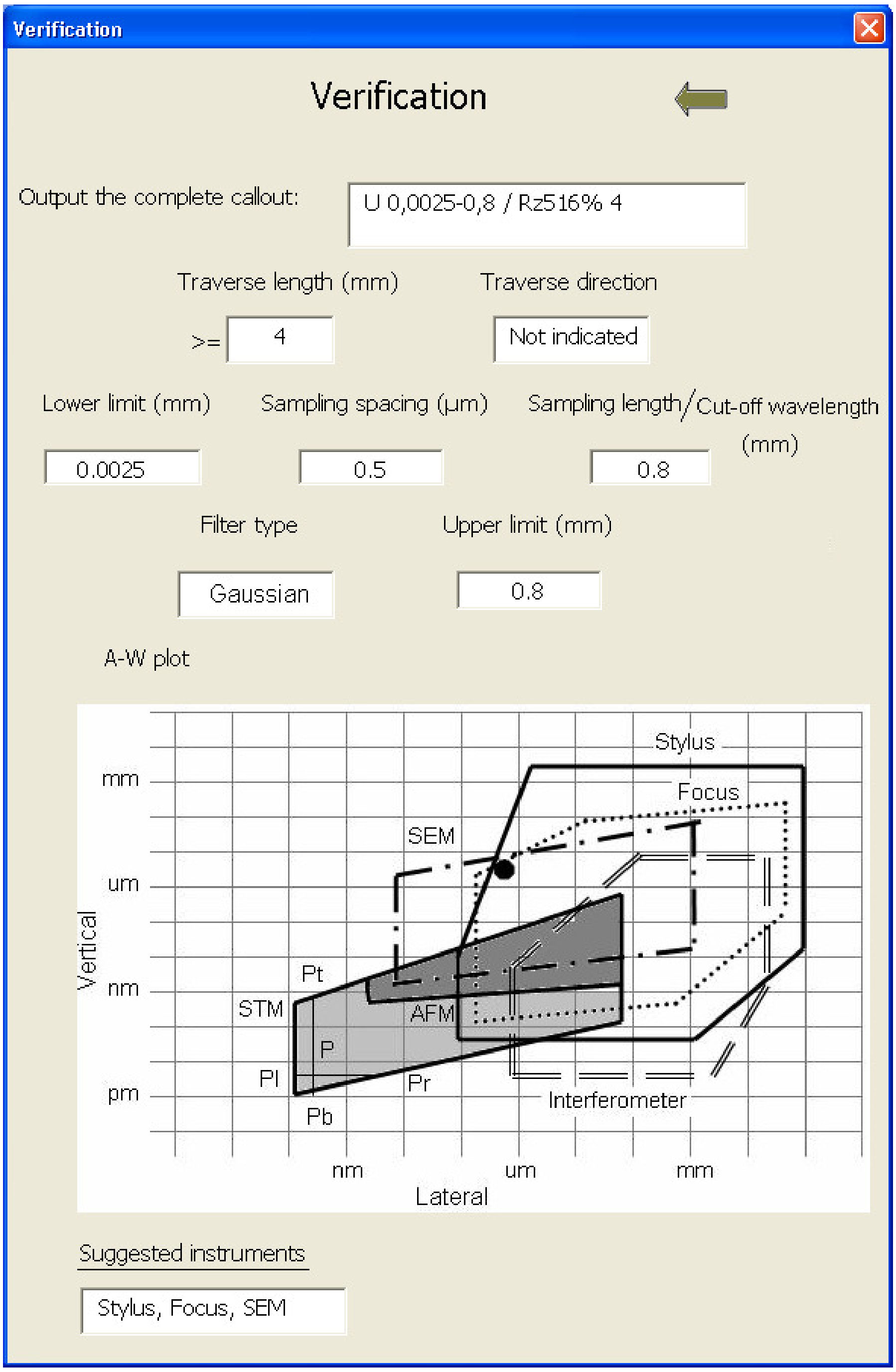


Figure 9.23 Output of Verification knowledge-base

Figure 9.23 shows the output of verification knowledge-base. The most important purpose is to suggest the measuring instruments. The A-W plot in the figure 9.23 which would help in making the decision. It is possible to locate the target parameter value on the plot and look for the instruments surrounded with the parameter. See the dot ● in the figure 9.23, which represents the target parameter value Rz 4. The x axis is located by the sampling spacing, and the y axis is located by the parameter value. In the example Rz 4, the sampling space is 0.5μm derived from ISO 3274 [10] and the parameter value is 4μm, thus the target dot is located in the position (0.5μm, 4μm). According to the A-W plot, three kinds of instruments - Stylus, Focus and SEM are surrounding with this dot, and may be capable of doing the measurement. Engineers can know more about characteristics of each suggested instrument by simply clicking the hyperlink of "Suggested Instruments".

9.4 Summary

These two case studies give a further insight into the application of Virtualsurf system. The case study of total hip replacement explains the implementation procedures for designers and metrologists in detail, and shows how the knowledge-based system works, including flowcharts and interfaces. The aim of this is to identify the most relevant parameters with functional performance of implants. The Virtualsurf system links Function, Specification, and Verification knowledge-bases together, provides guides for metrologists and designers, and helps them carry out the measurement and designed experiments step by step.

The case study of a cylinder liner focuses on the design functions of the Virtualsurf system, which covers four categories Function, Specification, Manufacture and Verification. The detailed design procedures are introduced, including the selection of parameters, indication of the specification, determination of manufacturing processes and verification procedures. This case study would particularly help engineers understand the design functions of this knowledge-based system.

References

[1] Leigh Brown (2006), *The Use of 3D Surface Analysis Techniques to Investigate the Wear of Matt Surface Finish Femoral Stems in Total Hip Replacement*, PhD Thesis, University of Huddersfield, UK

[2] Liam Blunt & Ciarán C. Murphy (2007), *Surface texture considerations on artificial hip implants – A review of the instruments and parameters available*

[3] *http://www.intellistem.com/science.htm*

[4] D.J.R. Cooper and J Fisher (1993), Clinical Materials 14 295

[5] D. Dowson (2001), *New Joints for the Millennium: Wear Control in Total*

Replacement Hip Joints, Proceedings of the Institution of Mechanical Engineers. Part H, Engineering in Medicine 215 (4): 335-358

[6] D. Najjar, M. Bigerelle, H. Migaud and A. Iost (2006), *About the relevance of roughness parameters used for characterizing worn femoral heads*, Tribology International 39 (2006) 1527-1537

[7] BS 7251:4 (1990), ISO 7206-2 (1987), *Part 4*: *Specification for bearing surfaces of hip joint prostheses*

[8] S. L. Smith, D. Dowson, A. A. J. Goldsmith (2001), *The Lubrication of Metal-on-Metal Total Hip Joints: A Slide down the Stribeck Curve*, Proceedings of The Institute of Mechanical Engineers – Part J; 215: 483-493

[9] M.A. Dagnall (1997), *Exploring surface texture – a fundamental guide to the measurement of surface finish*, Taylor Hobson Limited

[10] ISO 3274 (1996), *Geometrical Product Specifications (GPS) - Surface texture: Profile method - Nominal characteristics of contact (stylus) instruments*

[11] http://en.wikipedia.org/wiki/Cylinder_(engine)

[12] http://en.wikipedia.org/wiki/Piston_ring

[13] Robert Ohlsson, Bengt Goran Rosén and John Westberg (2003), *The interrelationship of 3D Surface Characterisation Techniques with Standardised 2D Techniques*, Advanced Techniques for Assessment Surface Topography, pp. 197-220, Kogan Page Science, ISBN 1 9039 9611 2

[14] http://www.aa1car.com/library/oil_consumption.htm

[15] http://www.misterfixit.com/blow-by.htm

Chapter 10

Conclusions and Future work

10.1 Conclusions

This thesis established the mathematical foundations for a virtual knowledge-based system for surface texture. The benefits of this surface texture knowledge-based system are concluded as firstly to improve communication — between designers, manufacturers and engineers; secondly to improve documentation — the knowledge-bases encapsulate surface texture information and produce a smaller, simpler, more maintainable document; and finally improve future designs — collective design experience can be applied to new projects through the knowledge-base.

A number of component studies were carried out and completed, the following conclusions can be drawn from those component studies:

1. Category theory has been investigated and developed as the mathematical foundation for the research, which was applied in three aspects:
   - Proved the stability of a measurement procedure, which ensures the consistency of knowledge acquisition for knowledge-based system;
   - Encapsulated relationships and structures within the surface texture knowledge, converted useful data structures into category structure models and to provide a framework for the data models of a virtual knowledge-based intelligent system for surface texture;
   - Established the consistency of data refinement processes. Data refinement is used to convert a simplified abstract data model into a concrete implementable data model. During the refinement processes, the initial knowledge-base is refined to produce a complex implementable system.

2. The integral framework and different structures contained in surface texture knowledge have been identified. Then the four knowledge-bases function, specification, manufacture and verification have been developed respectively:
   - Specification knowledge has been developed, and used to translate functional requirements into detailed geometrical specifications. The knowledge contained in the specification knowledge-base has been acquired; the relationships and structures have been identified and first represented by a

functional data model, and then transferred into a computer readable format based on category theory; interfaces were provided to illustrate the implementation.

- Verification knowledge has been developed, and used to determine the measurement procedures. The knowledge contained in the verification knowledge-base has been acquired; the relationships and structures have been identified and represented by a computer readable format based on category theory; the verification knowledge-base mirrors the specification knowledge-base, the detailed mirror relations have been explained; interfaces were provided to illustrate the implementation.
- Function knowledge has been initialized, and used to incorporate available resources about parameter selection rules and also provide an open platform for engineers to add their expertise and specific examples. The relationships between functional performance and surface texture parameters have been analysed; the pattern language was applied to represent the hierarchical structure contained in the function knowledge-base, and a category model was used to represent and record this pattern language; interfaces were provided to further illustrate the implementation of the knowledge-base.
- Manufacturing knowledge has been initialized, and used to determine the manufacture procedures. The relationships between manufacture processes and surface texture parameters have been analysed; the category model was applied to represent the knowledge, which not only has the ability to represent the selection flowchart, but also has object-oriented capabilities to represent the detailed PRIMA forms; interfaces were provided to further illustrate the implementation of the knowledge-base.

3. Two case studies have been carried out latterly to illustrate the implementation of the knowledge-based system for both designers and metrologists. The VirtualSurf system would provide a common language and act as a tool for designers, production engineers and metrologists to understand the latest surface texture knowledge.

10.2 Future Work

Areas have been highlighted from the literature review and also the studies completed which warrant further investigation, and these are outlined below:

- It would be desirable to incorporate more examples into the Function knowledge-base to provide a more general tool for the handling of the relationships between the functional requirements and surface specifications. For example, surface parameters selection for two parts in contact with rolling relative motions would be added to patterns 3-6 of Function pattern language.
- It would be desirable to incorporate more examples into the Manufacture knowledge-base to provide a more general tool for the handling of the relationships between the manufacturing processes and surface specifications. For example, new manufacturing processes such as electromagnetic forming would be added according to PRIMA format.
- It would be desirable to incorporate knowledge of the calibration and uncertainties into the Verification knowledge-base to accomplish a complete measurement procedure; and also incorporate more instruments to provide a more general tool for the handling of the relationships between the candidate instruments and surface specifications. For example, new instruments such as specific interferometer CCI, would be added with its main characteristics, including spatial and vertical measuring resolution and range.
- The system depends on a continuous update in order to stay current. Since areal measurement and analysis result in a better understanding of the functional performance of surfaces and a better control of their manufacturing [1], and ISO standards on areal measurement are still being developed, it will be possible to incorporate areal approach into the Virtualsurf system later.
- The system will be coded and programmed into an implemental software, based on the framework and data models developed in this thesis, and further connected with other analysis software, CAD software, next generation intelligent standard, etc, and finally applied in production industries.

  For example, the Virtualsurf system could be connected with the CAD system through a CAD interface, see figure 11.1. The knowledge stored in Virtualsurf system can be applied as a guide to the CAD system and help engineers in design, manufacture and metrology processes. For example, the function knowledge-base in Virtualsurf system will help engineers select the relevant surface parameters, while specification knowledge-base can be used to provide the indication of callouts and help the drawing of product surfaces.

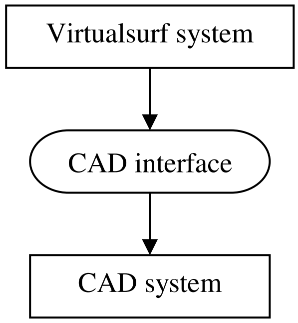


Figure 10.1 Integration with CAD system

The Virtualsurf system has several advantages when applied in production industry. It will be a good training tool for new and less experienced engineers and it can be applied as a knowledge-base and connected with other implemental software, such as the CAD system. Overall, this system links design, manufacture and verification of a product surface together, acting as an intelligent and handy surface texture library for engineers.

References

[1] E.J. Davis, K.J. Stout, and P.J. Sullivan (1990), *The Scope of Three Dimensional Topography*, Industrial Metrology 1, pp. 193-216

Appendix I

A Selection of publications resulting from this project

Y. Wang, Paul J. Scott and X. Jiang (2005), *The structure of surface texture knowledge,* Journal of Physics, Conference Series, 13(2005) 1-4

Yan Wang, Paul Scott and Xiangqian Jiang (2007), *Structures of Specification and Verification Knowledge-base in Surface Texture Knowledge-based System*,10^{th} CIRP conference on Computer Aided Tolerancing, March 21^{st}-23^{rd}, 2007 in Erlangen, Germany

Yan Wang, Paul Scott and Xiangqian Jiang (2007), *Surface texture knowledge-based system – Case studies from specification and verification*, Poster displayed at 11^{th} International conference on Metrology & Propertities of Engineering Surfaces, July 17^{th}-20^{th} 2007 in Huddersfield, UK

Yan Wang, Paul Scott and Xiangqian Jiang (2007), *Modeling of the surface texture knowledge-base*, Proceedings of DET2007, 4^{th} International Conference on Digital Enterprise Technology, Sep. 19^{th}-21^{st}, 2007 in Bath, UK. Selected for publication in International Journal of Computer Integrated Manufacturing.