Computer-Aided Design 57 (2014) 41–53

Contents lists available at ScienceDirect

Computer-Aided Design

journal homepage: www.elsevier.com/locate/cad

Design and implementation of an integrated surface texture information system for design, manufacture and measurement[☆]

Qunfen Qi, Paul J. Scott, Xiangqian Jiang [*], Wenlong Lu

EPSRC Centre for Innovative Manufacturing in Advanced Metrology, School of Computing and Engineering, University of Huddersfield, Huddersfield, HD1 3DH, UK

HIGHLIGHTS

- We developed a surface texture information system.
- Profile and areal surface texture modules each with five components is constructed.
- Category theory based knowledge representation mechanism is devised.
- We developed a platform to integrate the information system with CAx systems.

ARTICLE INFO

Article history:
Received 9 January 2014
Accepted 20 June 2014

Keywords:
Information system
Profile surface texture (PST)
Areal surface texture (AST)
Computer-Aided Design (CAD)
Geometrical Product Specifications (GPS)

ABSTRACT

The optimized design and reliable measurement of surface texture are essential to guarantee the functional performance of a geometric product. Current support tools are however often limited in functionality, integrity and efficiency. In this paper, an integrated surface texture information system for design, manufacture and measurement, called "CatSurf", has been designed and developed, which aims to facilitate rapid and flexible manufacturing requirements. A category theory based knowledge acquisition and knowledge representation mechanism has been devised to retrieve and organize knowledge from various Geometrical Product Specifications (GPS) documents in surface texture. Two modules (for profile and areal surface texture) each with five components are developed in the CatSurf. It also focuses on integrating the surface texture information into a Computer-aided Technology (CAx) framework. Two test cases demonstrate design process of specifications for the profile and areal surface texture in AutoCAD and SolidWorks environments respectively.

1. Introduction

The trend in global manufacturing, along with the emergence of computer-aided technologies (CAx), urges a rigorous and systematic common language to characterize geometrical products throughout the product supply chain. An international technical language, called Geometrical Product Specifications and Verification (GPS), has created a synergy for design, manufacture and measurement. It uses rigorous mathematical definitions of geometric specifications mapped to verification, and is intended to save design modification and manufacture time and to reduce scrap material in manufacture and measurement cost [1,2]. Comprehensive implementations of the GPS-language globally, will promote future manufacturing moving to a knowledge driven economic environment, where design, manufacture and measurement are integrated into a single engineering process that enables 'right first time' every time fabrication of customized products [3]. Such evolutions will force product technical specification and verification to be much more precise and with a clearer implementation methodology.

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

The implementations of some geometric characteristics were hindered. One example is surface texture, one of the most complicated geometrical specification and verification systems in GPS.

[☆] This paper has been recommended for acceptance by Vijay Srinivasan.

[*] Corresponding author. Tel.: +44 1484473634; fax: +44 1484472161.
E-mail address: x.jiang@hud.ac.uk (X. Jiang).

http://dx.doi.org/10.1016/j.cad.2014.06.013

Table 1
Status of specification design status for profile surface texture (PST) in commercial CAD systems.

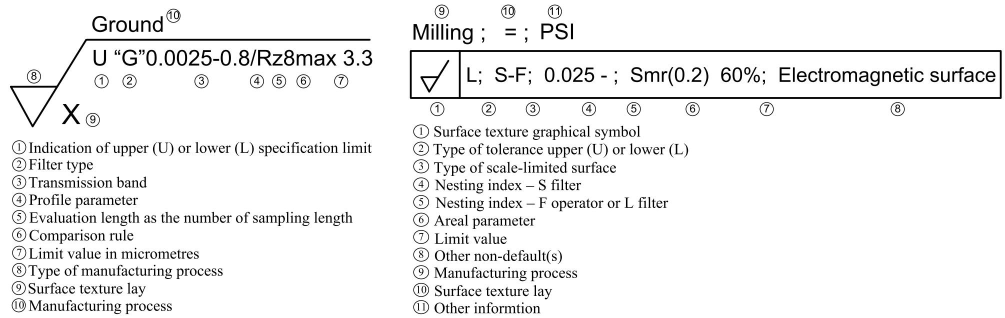
Commercial CAD/CAM/CAE systems		Surface texture specification design	PST standards		Database support
			Versions	Indications	
Autodesk	AutoCAD	None	None	None	None
	AutoCAD mechanical	Surface texture symbol tool	A simplified version from ISO 1302:2002	milling Ra 3.2	None
Dassualt Systemes	CATIA	Roughness symbol tool	ISO 1302 1965 version	3.2	None
	SolidWorks	Surface finish symbol menu	A simplified version from ISO 1302:2002	milling Ra 3.2	None
PTC	Pro/Engineer (PTC Creo)	Surface finish tool menu	ISO 1302 1965 version	3.2	None
Siemens	NX (Unigraphics)	Surface finish symbol tool	ASME Y14.36M-1996	Mill 2.5/Rz 6.3	None

It is relevant for the whole surface manufacture chain from design through manufacture and qualification, and plays a significant role in determining the functional performances of a workpiece, e.g. friction, wear and lubrication. In recent years, the characterization of surface texture has experienced a paradigm shift moving from profile measurement to areal measurement thanks to the rapid development of advanced measurement instruments and information technology [4,5]. Surface design, manufacturing and metrology are however disconnected, becoming a very complicated and ambiguous system, especially since the necessary skills/expertise are often not available in global supply-chains, SMEs and multi-country manufacturing.

One of the essential reasons for this disconnect is the complexity of surface texture knowledge in GPS. Currently, there are 29 GPS published standards for profile and areal surface texture, a set of new standards, including ISO 25178 series, are being issued. Those paper-based documents which contain a wealth of information under the GPS matrix structure have been recognized as being too complicated to be comprehended and implemented without an effective implementation methodology. To tackle this issue it requires a deeper understanding of the underlying reasons that can be summarized in five aspects:

1. *Limitations in existing surface texture information systems*. It has been found that majority of surface texture information systems were developed either for design support [6,7] or measurement support [8–13] only, until a so-called "VirtualSurf" project was undertaken by Wang [14] and Xu [15] to develop a knowledge-based system for the design and measurement of profile surface texture (PST). A preliminary framework of the VirtualSurf system has been proposed. However, it has not been developed as a comprehensive functional system for practical implementation. For example, the 'Function' part of the system was a 'function performance report' rather than having practical correlation with specification. Moreover, a comprehensive surface texture system with the support of AST is required due to the high functional demand of surface texture in industry, and with the rapid development of areal characterization and standards publication.
2. *Integration problems between surface texture information systems and CAx systems*. Current practice in manufacturing still relies on design based on traditional standards, which utilize a symbolic language, this being driven by the limitations of CAD/CAM/CAE multi-platform software suites including CATIA, AutoCAD, Pro/Engineer (PTC Creo), SolidWorks and NX, see Table 1. All roughness symbol tools in the CAD systems are requiring manual input of all the symbol elements without any assistance, such that they have no support to help CAD users choosing the appropriate elements and to prevent them generating a wrong roughness symbol.
3. *Restrictions of existing data representation methods for surface texture*. It was discovered by Wang, Xu and Lu [14–16] that traditional data models such as relational and object-oriented models had limitations to efficiently support complex data structures and to reflect the complicated relationships among engineered artefacts and surface texture GPS standards. The VirtualSurf system utilized category theory to develop an object-based modelling mechanism, since category theory is an alternative to the foundations of mathematics and can represent any mathematical object very efficiently (more efficient than set theory). It has been proved that the devised categorical DBMS (database management system) has been proved to be on an average 10 times faster than an analogue MySQL product when processing a query operation, as well as an average 1/3 memory cost of traditional relational DBMS when containing more than 500k data in memory [15]. This formalism, however, has still thrown up some issues. One of the essential problems is the rigorous application of category theory. The major definitions of category theory, are based on the categories, objects and arrows in/between them. The object-based categorical model [14,15] was focused on the objects rather than categories and relationships between categories. It significantly limited the effectiveness of category theory in dealing with complex relationships. A more rigorous categorical model is required to completely utilize the advantages of category theory.
4. *Limited correlation between function and surface texture specification*. The number of applications in which surface texture is involved is so large and diverse that a systematic approach has been difficult. There is an urgent need of function support tools to ensure the functionality of the assigned specification.
5. *Knowledge gap between specification and verification*. Some of the definitions in the surface texture standards still leave a room for several different interpretations [17,18]. Misunderstanding caused by the ambiguities and imperfections can result in significant information loss between specification and verification, especially when there are vast quantities of information for exchange. Currently there is also lack of support tools to help metrologists for measurement strategy, imprecise interpretation of specification can produce unmatched verification [19]. Assigning unambiguous specifications and mapping them rigorously with verification are essential tools to bridge the gap.

These issues highlight the necessity of a comprehensive implementation of surface texture in design, manufacture and measurement. The development of support systems and integrating them with CAx is one of the most efficient ways to allow partners collaborating effectively in creating innovative products [20]. The proposed work is to develop a surface texture information system to bridge



(a) Profile surface texture indication. (b) Areal surface texture indication.

Fig. 1. PST and AST indications [21,22].

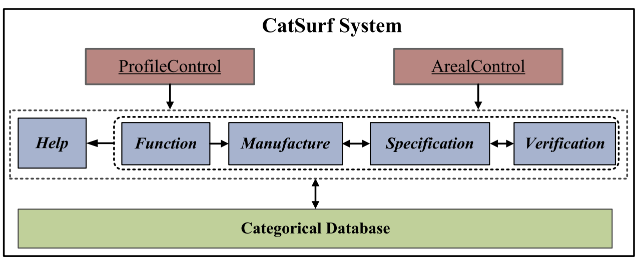


Fig. 2. Main components in the CatSurf system.

the knowledge gap between design, manufacture and measurement in surface texture. A prototype system, named “CatSurf” is designed and developed. It integrates the GPS information and GPS expertise methodologies (GPS expert’s guidance), covering PST and AST, specification design and related verification principles, measuring equipment and calibration requirements, including uncertainty and measurement traceability, into an intelligent system. The rest of the paper will address how the above five limitations are tackled.

2. Design system architecture

The CatSurf system focuses on integrating the profile and areal surface texture information and corresponding GPS realization methodologies into a CAx integratable framework. It spans knowledge domains from surface specification, related manufacturing processes, to verification principles, calibration requirements and uncertainty control. The GPS specifications should be designed based on the functional requirements, lower costs and shorter product lifecycles. Two of the essential functions within the system are:

- to procreate optimized and complete specifications that comply with ISO 1302:2002 (for PST) and ISO 25178-2:2012 (for AST), which defined 10/11 control elements in the indication of PST/AST requirements on engineering drawings (see Fig. 1);
- to provide unambiguous verification strategy and analysis tools that are rigorously based on the assigned specification.

2.1. System architecture

The architecture of the CatSurf system is constructed in accordance with the product chain in which surface texture is defined, also the profile and areal chains. As shown in Fig. 2, the main components of CatSurf system are presented with one database and two modules each with five components. *ProfileControl* is a module to deal with design and measurement of PST. *ArealControl* is developed to carry out the underdeveloped AST standards. Each module includes five components, which are *Function*, *Manufacture*, *Specification*, *Verification* and *Help*. The first three components are part of the design phase; the *Verification* component is designed for surface texture measurement; the *Help* component is developed to provide all the help information for the system. A categorical database is developed to support all the data and information store, manipulation, querying and reasoning in the two modules.

2.2. The categorical database

The knowledge modelling for PST and AST in the CatSurf is based on category theory, a relatively new and high-level (abstract) form of mathematics language that focuses on how things behave rather than on what their internal details are [23]. It has the capability for providing an effective and natural formalism for object-based databases [24,25]. One of the attractions of category theory is the ability to combine diagrammatic formalisms as in geometry with symbolic notations as in algebra: in computing science, diagrams are a common way of mastering complexity and symbolic notation is used for proofs and computation. With the facility to specify formally transformations between different types of mathematical structures, category theory provides a powerful way of modelling complex systems with heterogeneous structures.

Category theory is based on the concept of a morphism, which is an abstraction derived from structure-preserving mappings between two mathematical structures, generally thought of as an arrow and represented by “$\rightarrow$” [26]. The arrows can denote any static condition or dynamic operation and therefore can cope with descriptive, prescriptive equivalent views. For example, the arrow is a generalization of mathematical symbols such as $>$, $=$, $\subset$, $\in$ and $f(x)$ with the usual respective meaning of comparison, equality, partition, membership and functional image. The major definitions of category theory, are based on the categories, objects and arrows in/between them. Some good starting literature on category theory includes [23,27,28].

The knowledge of surface texture includes massive diverse concepts and structures which cover specification definitions, definition categories, semantic understanding, algebraic structures, structured entities and relationships between all of them. The range of knowledge covers mechanical design, manufacturing information, surface metrology and information technology. The diverse nature of the knowledge makes it hard to apply in computing science. Using the categorical constructions, a categorical model is constructed to capture the semantics of surface texture. Fig. 3 shows a category **ATD** defined in AST, including seven objects and nine arrows between them.

The categorical model for PST and AST developed in Refs. [29,30] are the foundation of the categorical database. The database is then developed with various sections to fit the purpose of the components for the two modules. The query language for the

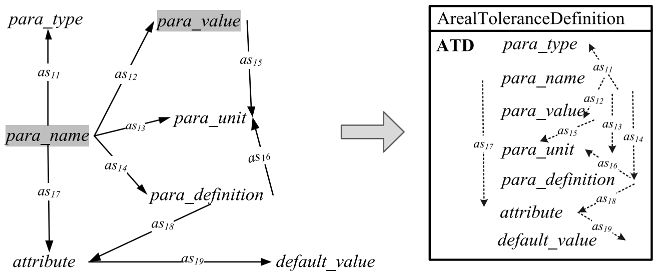


Fig. 3. Objects and arrows in a category **ATD** (Areal Tolerance Definition).

database is also based on the categorical model structured in [29, 30]. More details of the interaction between the four components of each module will be introduced later.

2.3. System components

The five components are designed to provide both designers and metrologists with related information. Designers are involved in *Function*, *Manufacture* and *Specification* components; metrologists are involved in the *Verification* component as shown in Fig. 4. The components are designed with related databases, interfaces, input and output data processing mechanisms. Depending on the external input of function and other requirements, all output data will be transferred to the next component.

2.3.1. The Function component

This component aims to provide all relevant information for the engineered artefact surface before the assignment of a specification. To ensure the unambiguity and functionality of the assigned specification, the information in this component is carefully selected from open-source information such as handbooks, research papers, published case studies and internal research outputs, which populates a universal categorical model which capsules the structure of *Function*. The output of this component to provide designers with optimized specification elements such as suggested parameters, limit values, applicable manufacturing processes, etc. Besides the common objectives with other components, the design of *Function* component is expected to:

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

The two databases which are *Function* database and other information database for storing and deducing related information are placed in the component. A *Function* interface is designed for gaining input data and outputting the deduced results. The interface provides various surface functions, component information, materials and other information used for selection. The designers input the requirements, and then the data will be sent to the *Function* database or other information database for related output information such as function-related parameter or limit value. The interface of the *Function* component for *ArealControl* module is shown in

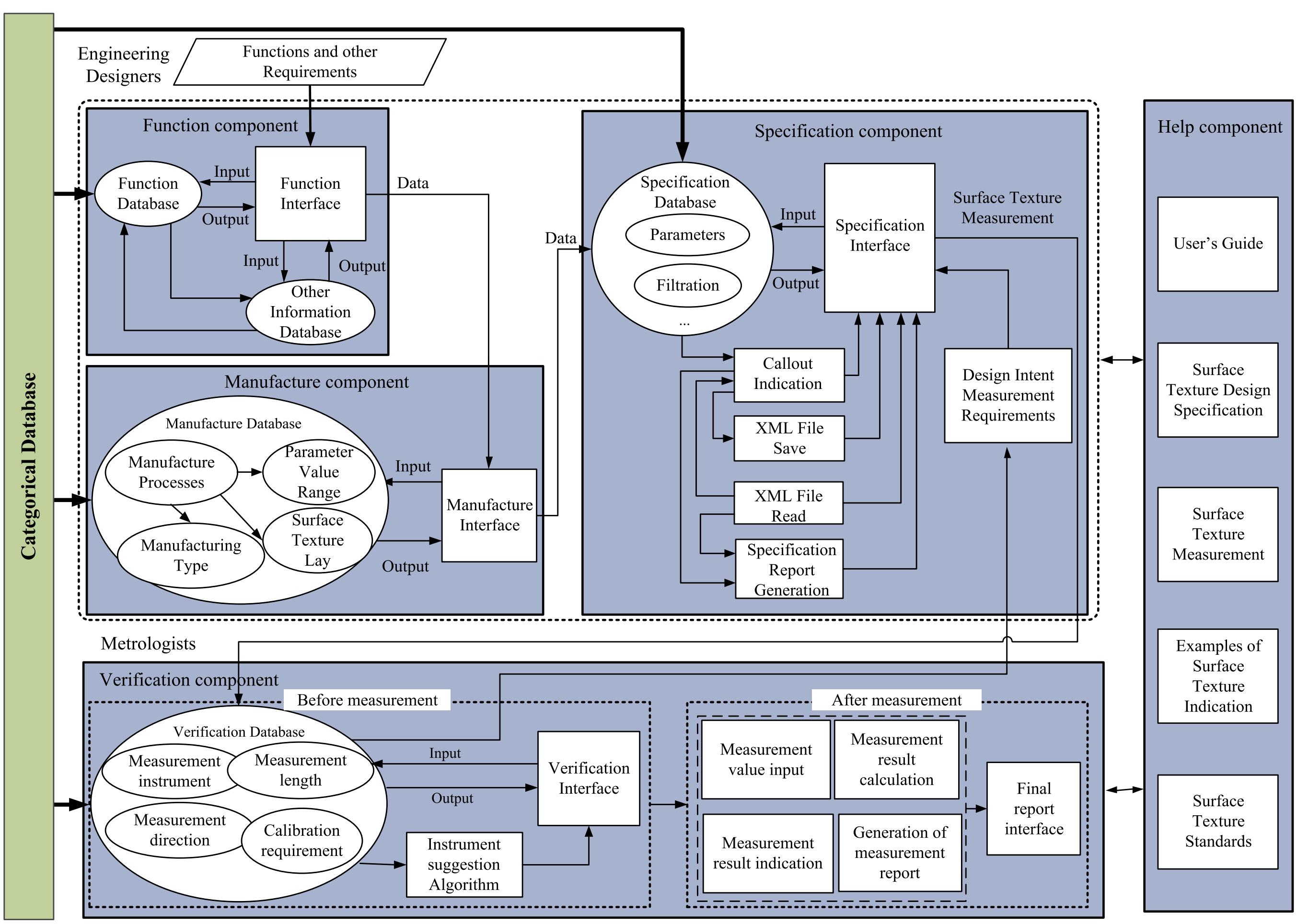


Fig. 4. The architecture of the CatSurf system.

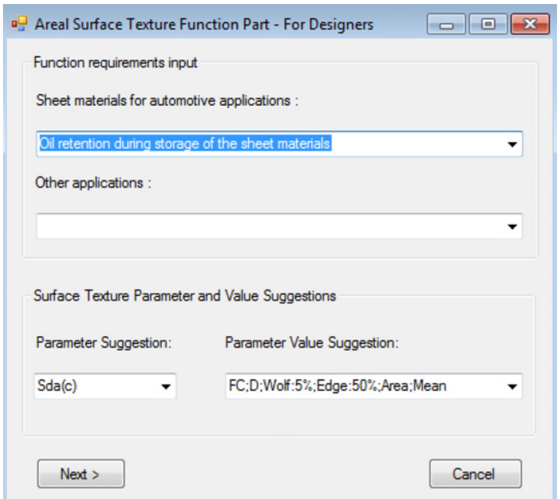


Fig. 5. The interface of the *Function* component in *ArealControl*.

Fig. 5. The selected functional requirement is "Oil retention during storage of the sheet materials". By analysing the input, the database calls the relationship AP_3 in the AST categorical model (will be detailed in Section 2.3.5), suggested parameter is $Sda(c)$ and value is "FC;D;Wolf:5%;Edge:50%;Area;Mean".

2.3.2. *The* Manufacture *component*

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

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

Transferring the function selection and output data in *Function* component, the *Manufacture* interface will link to the manufacture database for inferring the right manufacturing process and related information such as parameter value range and surface texture lay. For example, if the specified surface is designed to be manufactured by turning, the expected range of *Ra* is 0.025–25 μm and possible surface texture lay will be '=', '⊥' or 'C' if it is the end face of a cylinder. The interface of the *Manufacture* component in *ArealControl* is shown in Fig. 6.

2.3.3. *The* Specification *component*

This component aims to provide optimized and complete surface texture specifications for designers with the least amount of input information. The specification is the design step where all control elements are stated, accommodating the design requirements of workpiece and their functional surfaces commensurate with production capabilities for the use of design and engineering drawings. The data from both *Function* and *Manufacture* components will be sent to this component for generating a complete specification. The design of the component is expected to:

- avoid indiscriminate using of surface texture values that result in impractical and costly production requirements;

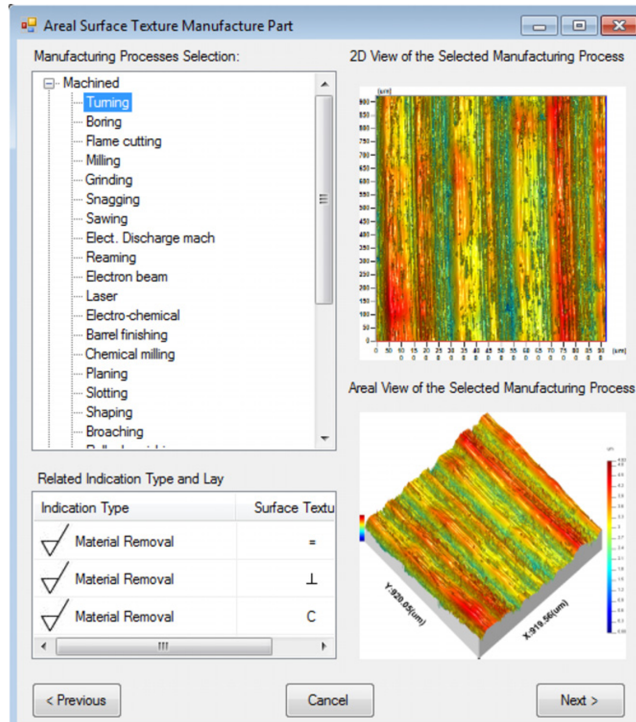


Fig. 6. The interface of the *Manufacture* component in *ArealControl*.

- generate a complete specification based on the information gained in previous components;
- provide the opportunity for designers to revise the specification details in accordance with their specialized requirements;
- generate and save indications and specification data;
- provide a specification report of interpretation for indications;
- provide basic measurement information for designers.

The process of generating a complete specification is carried out by the specification categorical model presented in [29,30]. In the interface shown in Fig. 7, the designers are allowed to change the details of certain specification elements under limited privileges. However, any revisions which are contrary to previous inputs such as functional requirements and manufacturing process, or any other inputs which is contrary to the relationship restriction in the specification models will not be allowed. The generated specification will be saved into an XML file; every detail of the specification will be explained in a specification report. Furthermore, the measurement database in the *Verification* component will be connected to this such that designers are provided with the required indications so that they have a straightforward understanding about the measurement requirements of the assigned specification.

2.3.4. *The* Verification *component*

This component is split into two different sections—the *Measurement strategy* and the *Final report*. The *Measurement strategy* is designed to:

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

The *Final report* is designed to:

- record the details about the measurement environment such as measurement time, humidity and operator;

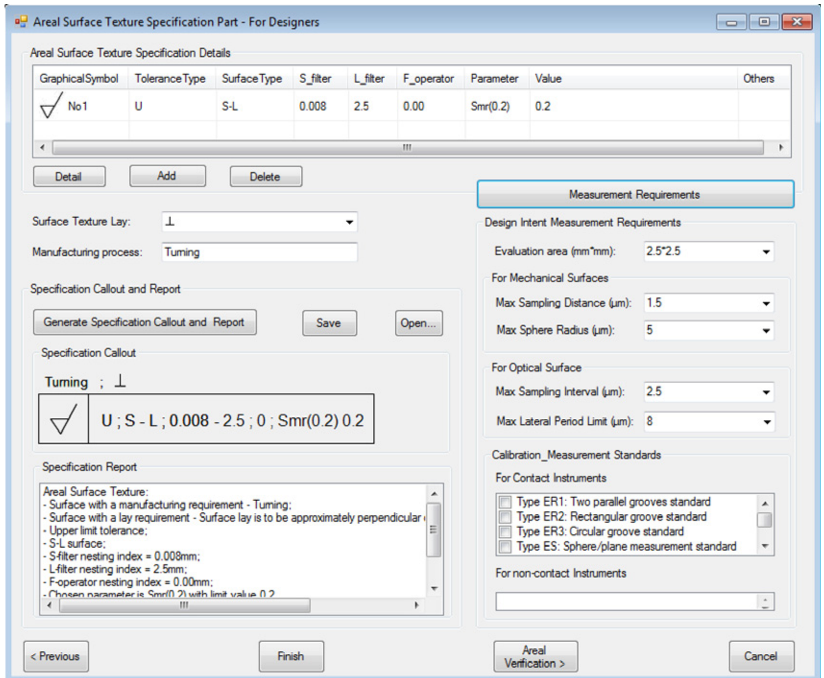


Fig. 7. The interface of the *Specification* component in *ArealControl*.

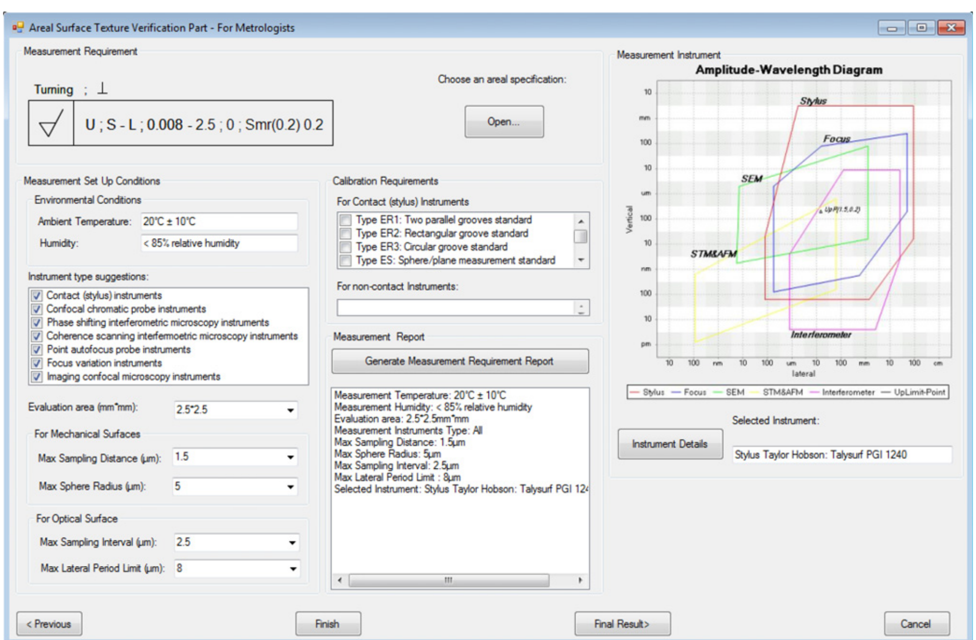


Fig. 8. The interface of *Verification* component in *ArealControl*.

- calculate the number of measurements;
- estimate the measurement uncertainty;
- indicate the measurement result;
- provide a conformance zone to make a measurement result decision with reference to the specification and uncertainty.

To provide the recommended instrument, an instrument suggestion algorithm is placed in the section. A main *Verification* interface is developed to provide both the *Measurement strategy* (see Fig. 8) and the attainment of *Final report* interfaces (see Fig. 9).

2.3.5. The interaction between the four components

The interactions in and between the four components are modelled rigorously by employing categorical factors such as categories, morphisms, functors and pullbacks structures as shown in Fig. 10, where each component is modelled by different sets of categories (each composed of a set of objects and morphisms) and pullback structures. The morphisms between objects in the same category are represented by dashed line arrows with labelled as_i (for Specification) and av_i (for Verification). The arrows AI_j show the direction of the inheritance between categories. The interactions between *Function*, *Manufacture* and *Specification* are represented by pullback structures AP_k as listed below:

AP_1 —the interaction between *Function* (categories **AI**) and *Specification* (category **AP**):
AI-object: *manufacturing_process* → **AP**-object: *manufacturing_method*;

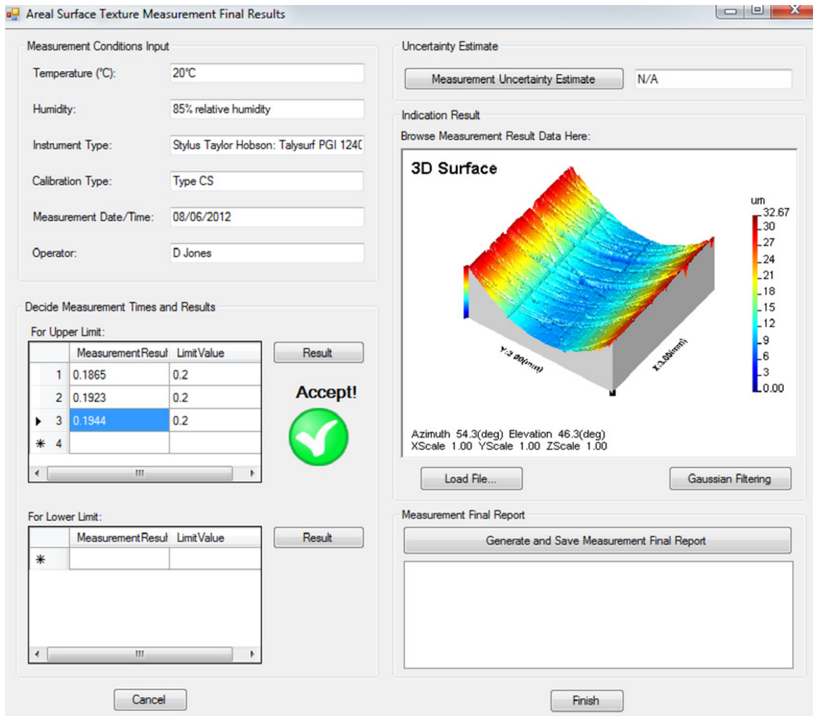


Fig. 9. The interface of *Final report* in *ArealControl*.

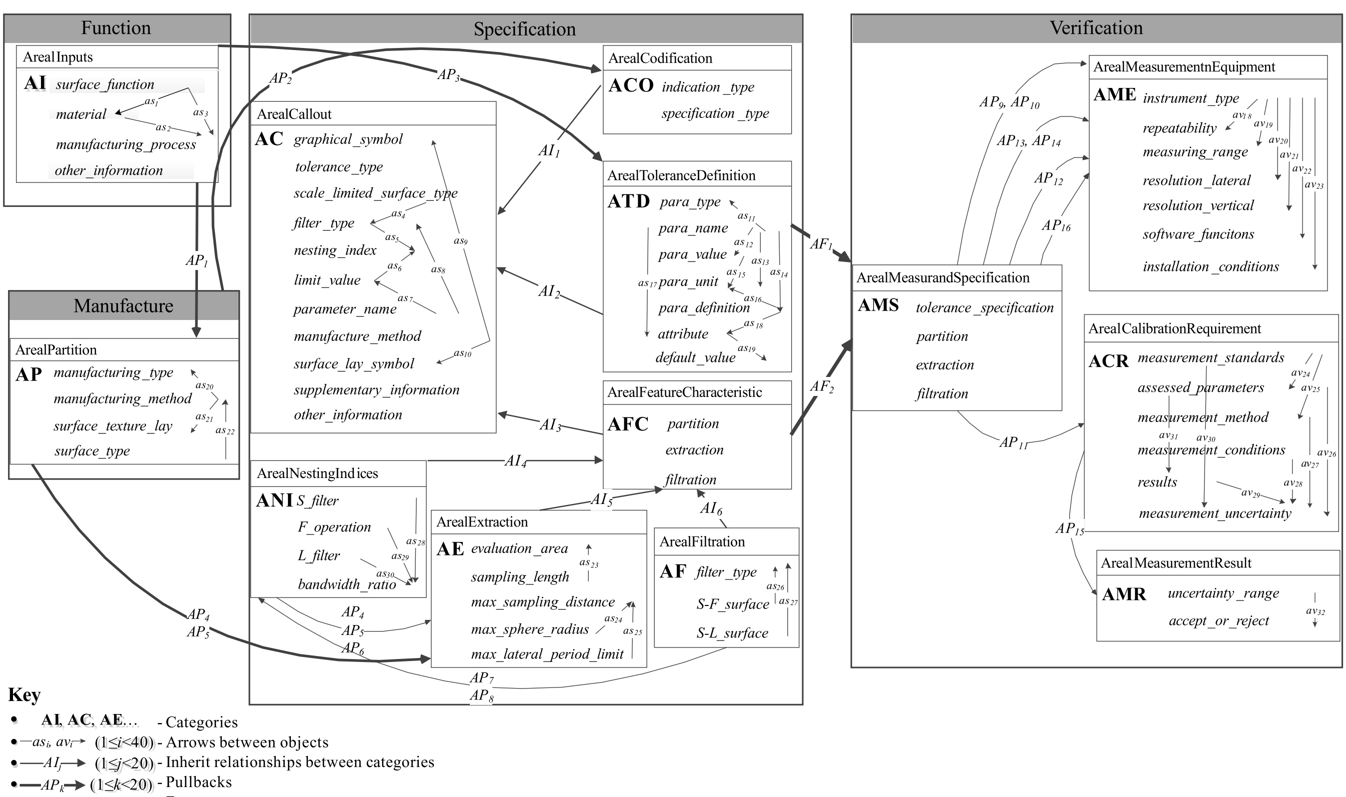


Fig. 10. The interaction between *Function*, *Manufacture*, *Specification* and *Verification* components using categorical modelling.

AP_2 —the interaction between *Manufacture* (categories **AP**) and *Specification* (category **ACO**):

AP-object: *manufacturing_method* $\rightarrow$ **ACO**-object: *indication_type*;

AP_3 —the interaction between *Function* (categories **AI**) and *Specification* (category **ATD**):

AI-objects: *functional_surface* $\times$ *material* $\times$ *other_information* $\rightarrow$ **ATD**-objects: *para_name* $\times$ *para_value*;

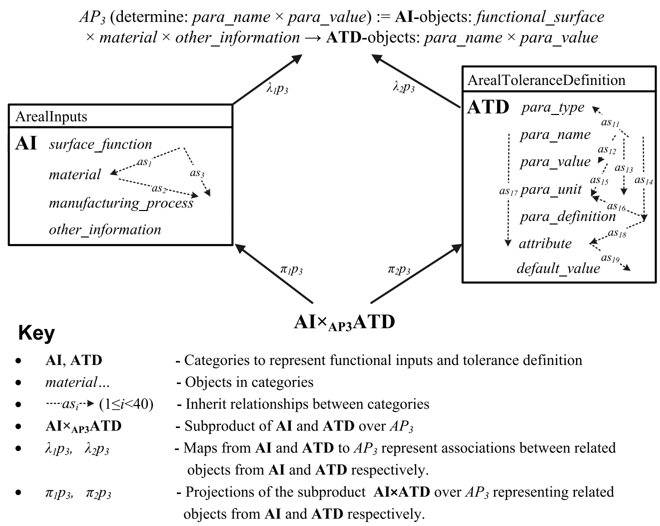


Fig. 11. The relationship AP_3 in the AST categorical model [30] for determining the parameter name and limit value.

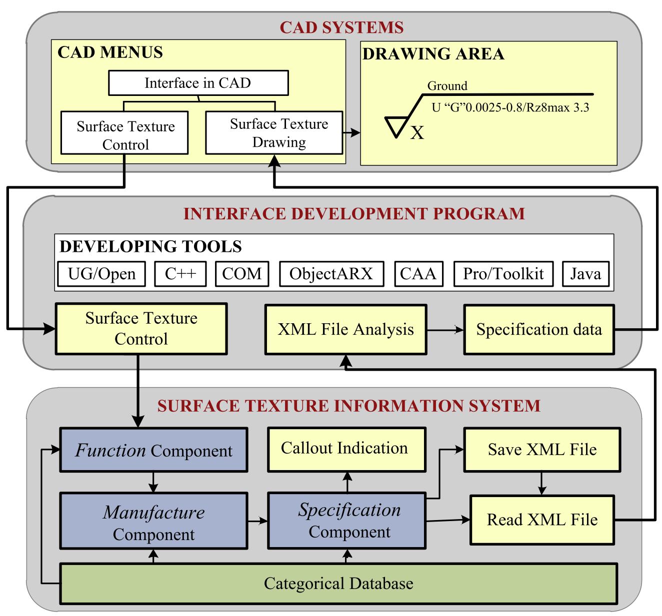


Fig. 12. The architecture of the integrated computer-aided surface texture platform.

AP_4 and AP_5 —the interaction between *Manufacture* (categories **AP**) and *Specification* (categories **AE** and **ANI**):
AP-object: *surface_type* × **ANI**-object: *S_filter* → **AE**-objects: *max_sampling_distance* × *max_sphere_radius*;
AP-object: *surface_type* × **ANI**-object: *S_filter* → **AE**-objects: *max_sampling_distance* × *max_lateral_period_limit*;

Fig. 11 showed a pullback example AP_3 of the interaction between the *Function* and *Specification* components, where an AST parameter and limit value have been derived. Pullbacks such as AP_1 and AP_2 show a more straight forward interaction between components. One of the interactions between the *Manufacture* and *Specification* components is to determine the indication symbol from the selection in the *Manufacture* component; it will employ Pullback AP_2 to determine the symbol type in accordance with the selected manufacturing process.

AP_4 and AP_5 are two pullbacks with complex structures. It requires a composition of categories **AP** (Areal Partition) and **ANI** (Areal Nesting Indices) to produce a new subcategory called **SPNI** with subobjects *surface_type* and *S_filter*, then the pullbacks represent relationships between category **SPNI** *and* **AE** (Areal Extraction), to determine the **AE**-objects *max_sampling_distance*, *max_sphere_radius* and *max_lateral_period_limit*.

The interaction between *Specification* and *Verification* components is implemented by the functors AF_1 and AF_2. A functor in category theory is a structure-preserving morphism (arrow) from a source category to a target category. An obvious case is when the shape of the target category is determined by the functor. It accommodates all assignments from the source category and has no other structure of its own. This property of functor ensures a rigorous mapping from specification to verification, such that the results derived from *Verification* components are traceable and unambiguous. The functor AF_1 is the morphism between category **ATD** (Areal Tolerance Definition) and subcategory **ATS** (Areal Tolerance Specification) inherited from **AMS** (Areal Measurement Specification), and functor AF_2 is the morphism between category **AFC** (Areal Feature Characteristic) and subcategory **AVFC** (Areal Verification Tolerance Specification) inherited from **AMS**.

3. The integrated platform

3.1. The platform framework

The architecture of the platform is represented in Fig. 12, where in the middle of the diagram an interface development program package is developed to carry out the interaction between CAD systems and the CatSurf. The interface program, which is programmed by specific development tools for different CAD systems, will execute the CAD commands, direct to the information system for performing the specification design process, and derive the specification data to generate indications in the drawing area.

A universal XML based approach is proposed to integrate the surface texture information system and CAD systems. Specific interface development programs are embedded with the functions of reading and analysing the XML data. The designed specifications are saved to XML files which are determined by a specified format. An interface application program with two embedded function menus is developed to read the XML files, to transfer the specification data to a CAD database, and to execute the command from the interface in the CAD. The menu "Surface Texture Control" is used to open the surface texture information system for specifications design. The menu "Surface Texture Drawing" is used to read and analyse the saved XML file, translate the specification data to CAD systems, then generate the surface texture indications in the CAD drawing space. Sharing the same address space and making direct function calls, the interface application is programmed by specialized software development tools provided by different CAD systems.

3.2. XML schema for surface texture specification

This section shows how to represent surface texture information using XML schema in multiple layered conformance levels to meet different application domains' requirements (see Table 2). As a markup language, XML provides the standard format for structured document/data exchange, and leaves potential possibility to solve interoperabilities problems [31]. XML schema is also utilized to store 3D surface texture data by the openGPS Consortium [32], named X3p, which is a file format capable of storing a wide range of 3D data types such as Line Profiles, areal surface profiles, unordered point clouds and Multi-layer line- and Surface profile data. It requires a fully compatible reading platform for different instrument software. Comparing with the OpenGPS X3p's format, the XML schema in this paper is to define the file format to store and transfer the specification data (specification elements and indication elements) between CatSurf and CAD systems.

Table 2
Four levels of the XML schema for AST.

First level	Second level	Third level	Fourth level
Areal surface texture	Specification	Symbol	UnsignedByte
		ToleranceType	String
		SurfaceType	String
		SFilter	Decimal
		FOperator	Decimal
		LFilter	Decimal
		Parameter	String
		LimitValue	String
		OtherNonDefault	String
		ManufacturingProcess	String
		Lay	String
		OtherInfo	String
	Indication	ManufacturingProcessElements	String
		SpecificationElments	String
		LayElements	String
		OtherInformaiton	String
		FontSize	UnsignedByte
		LabelVisible	String
		LayOrientation	UnsignedByte
		Mode	UnsignedByte
		Zoom	UnsignedByte
		AutoFontSize	String
		Position	UnsignedByte
		CalloutNumber	UnsignedByte

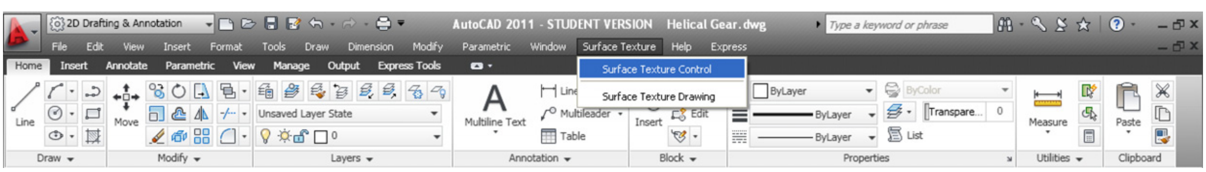


Fig. 13. The embedded menu interface in AutoCAD 2011.

3.3. *Programming achievement in AutoCAD and SolidWorks*

Most current commercial CAD systems such as AutoCAD, SolidWorks, Pro/Engineer employ a surface texture model only as an indication tool. This paper mainly focuses on the integration of the information system with AutoCAD and SolidWorks. The integration with other CAx systems will be implemented in future work.

Two sections have been developed in the interface programme for AutoCAD 2011. The menu of the two sections is shown in Fig. 13, and is developed using COM. The first section is an interface which connects with the CatSurf system. AutoCAD users using this section have access to the system to assign the surface texture specification. When users finish the specification design, the saved XML file will be sent back to the interface program. The second section is built into AutoCAD 2011 using ObjectARX 2011 as shown in Fig. 14. This program firstly reads the XML file, and transforms the data format to the AutoCAD program. It then generates a surface texture indication block, and inserts it onto the engineering drawing with a certain angle, position and scale according to the users' selection. The indication block is saved in the database of AutoCAD.

A "Surface Texture Addin" with two sections has been developed in SolidWorks 2009 using Visual C#. The menu and interface are shown in Fig. 15.

4. Validation of the CatSurf and interface programs

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

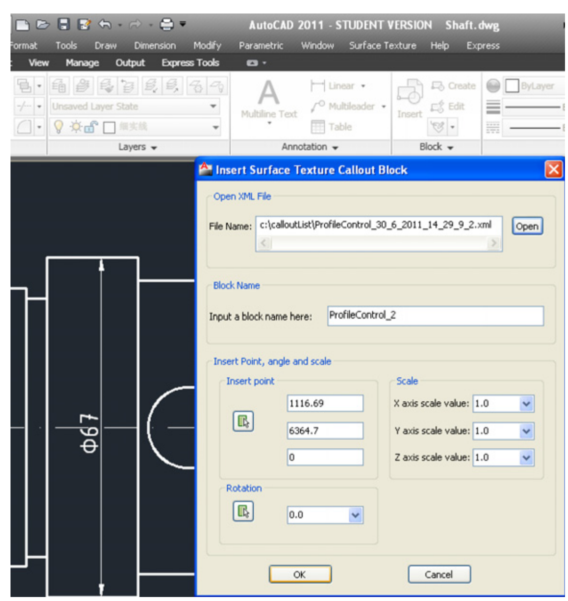


Fig. 14. The interface in AutoCAD for inserting surface texture specification blocks.

4.1. *Profile specifications design for a helical gear in AutoCAD*

The test case aims to assign a profile specification for a helical gear which is shown in Fig. 16. It is held in the *ProfileControl* module in the CatSurf and AutoCAD 2011. There are three steps in the system to assign a complete specification for a helical gear tooth.

Step 1: In the *Function* component, select the correct functional surface type and material. The selected functional surfaces

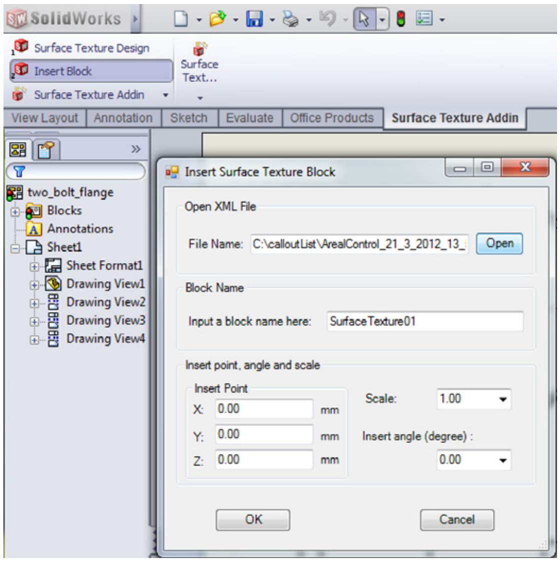


Fig. 15. "Surface Texture Addin" in SolidWorks.

is "Spur and helical" for "Gear teeth"; and the selected material is "Steel Titanium and Heat Resisting Materials".

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

Step 3: In the *Specification* component, the details of specification are generated automatically. The indication and XML file are saved and the XML file is named "ProfileControl_2_5_2012_11_50_59_41.xml", as shown in Fig. 17.

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

Step 4: Click "Surface Texture Drawing" menu, open the "Insert Surface Texture Callout Block" interface. In the interface, open the saved XML file "ProfileControl_2_5_2012_11_50_59_41.xml".

Step 5: Change the name of the block; select the insertion point, scale and rotation. Insert the block in the drawing, see Fig. 18.

Step 6: Repeat Steps 1–5 to design more specifications for a different surface in the helical gear. Alternatively it is possible to insert the saved blocks for the surfaces with the same requirements.

4.2. Areal specifications design for a stepped shaft in SolidWorks

The second test case aims to assign areal specifications for a stepped shaft which is shown in Fig. 19. Determined by the functional requirements, the shaft is divided into six segments. The shaft segment 1 of 55 mm diameter is manufactured by fine turning and is an interference fit with a rolling bearing. The segment 2 of 58 mm diameter with IT grade 7 is interference fitted with a helical gear. The segment 3 of 55 mm diameter is manufactured by fine turning and is an interference fit with a sleeve. The segment 4 shares the same shaft with segment 3, and is an interference fit with a rolling bearing. The segment 5 of 52 mm is manufactured by turning and is a sealing fit with an end plate. The segment 6 with IT grade 7 is an interference fit with a flat key.

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

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

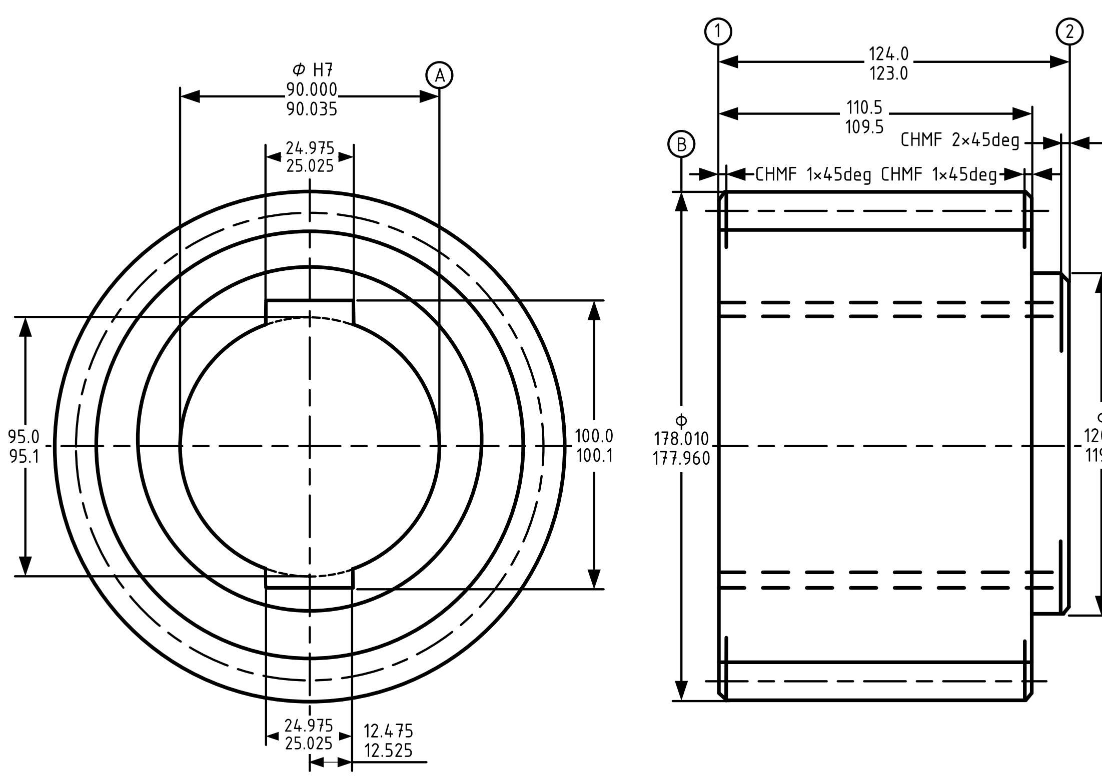


Fig. 16. The design of a helical gear.

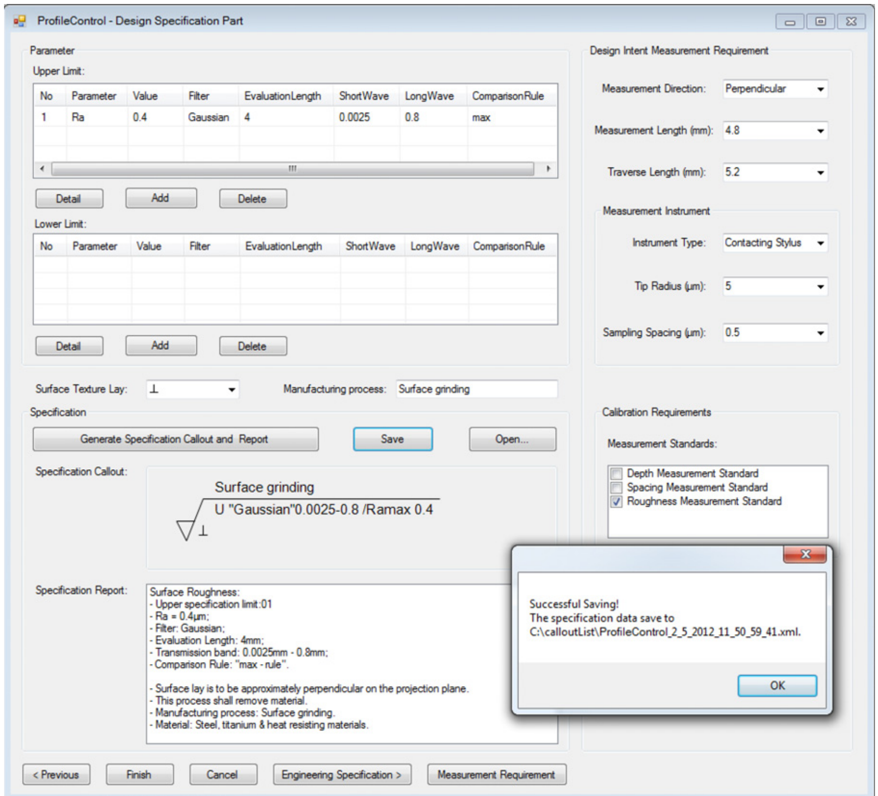


Fig. 17. The generation of the specification in the *Specification* component of *ProfileControl*.

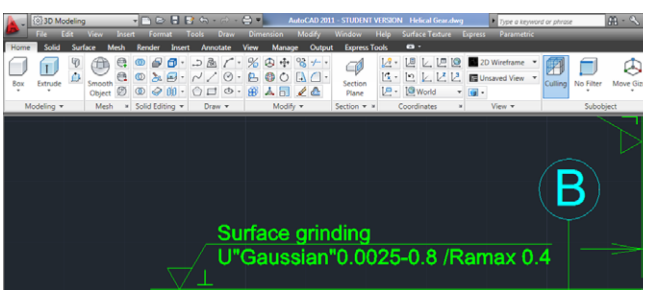


Fig. 18. Insert the saved specification in AutoCAD drawing.

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

Step 3: In the *Specification* component, the details of areal specification are generated automatically. The indication and XML file is saved and named "ArealControl_3_5_2012_12_15_2_8.xml" as shown in Fig. 20.

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

Step 4: Click "Insert Block" menu, open the "Insert Surface Texture Callout Block" interface, then open the saved XML file "ArealControl_3_5_2012_12_15_2_8.xml".

Step 5: Change the name of the block; select insert point, scale and rotation. Insert the block in the drawing (as shown in Fig. 21).

Step 6: Repeat Steps 1–5 to design specifications for segments 2–6. The suggested parameter for segment 2 is *Sa* of 0.8 μm, for segment 3 is *Sa* of 0.8 μm, for segment 4 is *Sa* of 0.4 μm, for segment 5 is *Sa* of 0.6 μm and for segment 6 is *Sa* of 1.6 μm.

5. Conclusions and future work

The design and development of the CatSurf have been described in this paper. The target concept for the CatSurf system is that a designer, a product engineer, a metrologist or a manufacturing

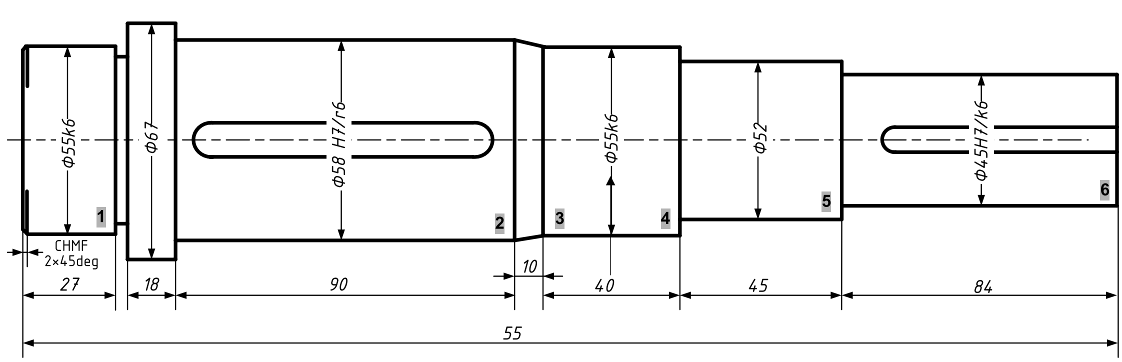


Fig. 19. The design of a stepped shaft.

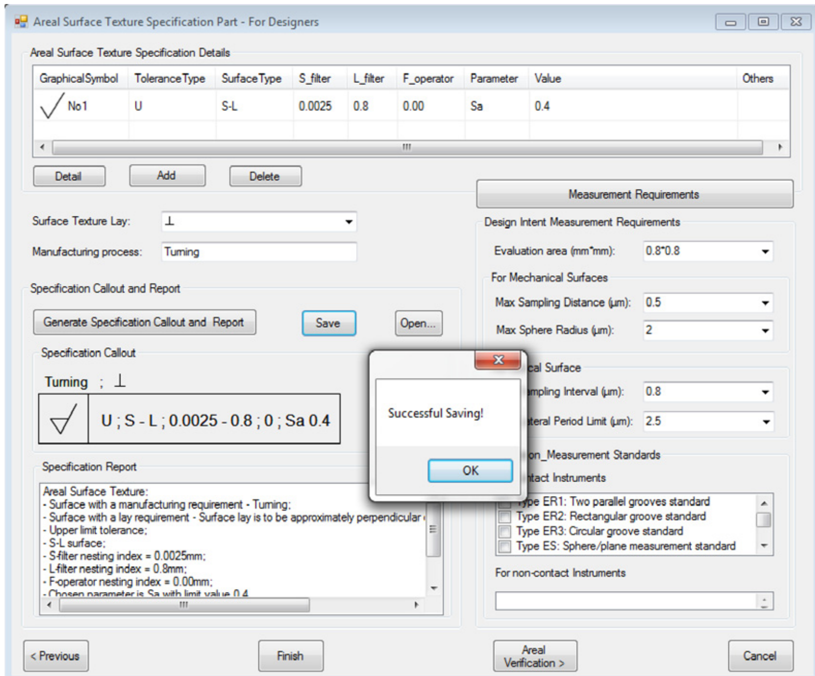


Fig. 20. The generation of specification in *Specification* component of *ArealControl*.

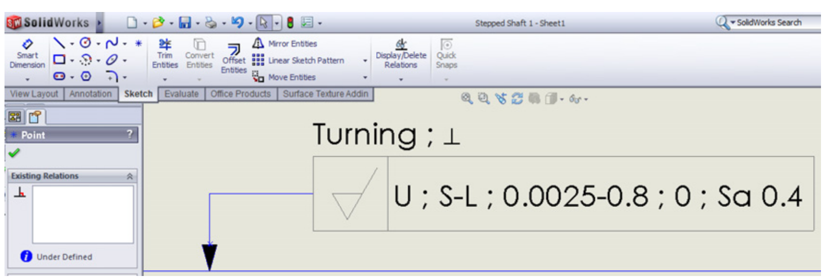


Fig. 21. Insert the saved specification in SolidWorks.

manager does not need to be an expert in surface texture GPS standards, having in mind all the complex standardized rules. Should the product function change, the product surface texture specifications will remain consistent to the GPS standards. It is also designed to ensure seamless synergy between design, manufacture and measurement stages. The system has some distinctive advantages over the other conventional systems for surface texture:

1. A comprehensive and standalone system framework with one categorical database and two modules for PST and AST (each composed of *Function*, *Manufacture*, *Specification*, *Verification* and *Help* components) to provide unambiguous surface texture information for designers and metrologists, and enables metrology assisted design and manufacture to become reality.
2. A unified integration methodology between CatSurf and CAD systems, based on XML and COM technologies, has been developed and implemented. Currently it is successfully integrated with CAD platforms such as AutoCAD and SolidWorks, and has been tested by a world-leading manufacturing company.
3. The categorical model proposed is comprehensively updated compared with the model proposed by Wang and Xu. It redefines the families of categories, the relationships such as pullbacks, categories pullbacks and functors, providing a more flexible, clear and easy way to update models for surface texture. One of the important functions of the updated categorical model is to implement seamless and rigorous interactions between different components involved in design, manufacture and measurement stages.
4. A *Function* component has been designed in the CatSurf system to provide a universal structure to encapsulate the relationships within the correlation between functional requirements and assigned specifications. The information embedded in the component is firstly modelled by a categorical model; this structure is then populated by information selected from open-source information such as handbooks, research papers, published case studies and internal research outputs.
5. Rigorous mapping model from *Specification* to *Verification* components has been modelled and implemented in the CatSurf system using functors in category theory. The property of functor ensures a rigorous mapping from specification to verification, such that the results derived from *Verification* component are traceable and unambiguous.

In terms of involvement in a supply chain, a manufacturing company can design a complete specification using the CatSurf. A detailed specification report and a related verification report can be generated and then be sent out together with the specification to suppliers. Whether they are located in different countries, using different standards and speaking different languages, the specification report can help them to understand every element of the specification; the verification report can guide them to measure the surface precisely. Their measurement results can also be verified by the manufacturing company using the *Final Report* function in the *Verification* component.

However, our work cannot be considered without limitations. Detailed work in the development of the system revealed more interesting issues each of which needs to be further investigated or updating. An interesting issue that arises out in the *Function* component is the difficulty of discovering the correlation between functional requirements and surface texture specifications. It would be desirable to incorporate more examples into the categorical database, for instance, industrial correlation results. Advanced industrial users need to be assigned administrational privileges for the functional database in the future, such that they could update and modify the correlations by themselves.

The knowledge model of AST developed for *ArealControl* module requires continuous updating with the development of standards, especially the AST symbol needs to be updated with the progress of draft ISO 25178-1. The functions for the *Verification* component in both profile and areal modules require further implementation. For instance, the uncertainty estimation function is currently not implemented and requires further exploration. The indication of areal measurement data and filtration requires further updating as well.

A simplified surface texture specification module for beginner users who only require some simple functions of the CatSurf system may be required.

Acknowledgements

The authors gratefully acknowledge the UK's Engineering and Physical Sciences Research Council (EPSRC) funding of the EPSRC Centre for Innovative Manufacturing in Advanced Metrology (Ref: EP/I033424/1), and the funding of the EPSRC Fellowship in Manufacturing (Ref: EP/K037374/1), and the European Research Council under its programme ERC-2008-AdG 228117-Surfund.

References

[1] Srinivasan V. A geometrical product specification language based on a classification of symmetry groups. Comput-Aided Des 1999;31:659–68.
[2] Dantan J, Ballu A, Mathieu L. Geometrical product specifications—model for product life cycle. Comput-Aided Des 2008;40:493–501.
[3] National Physical Laboratory, Metrology for the 2020s, 2012.
[4] Jiang X, Scott PJ, Whitehouse DJ, Bllunt L. Paradigm shifts in surface metrology. Part I. Historical philosophy. Proc R Soc A 2007;2049–70.
[5] Jiang X, Scott PJ, Whitehouse DJ, Bllunt L. Paradigm shifts in surface metrology. Part II. The current shift. Proc R Soc A 2007;2071–99.
[6] Rosén BG, Ohlsson R, Westberg J. Interactive surface modelling, an implementation of an expert system for specification of surface roughness and topography. Int J Mach Tools Manuf 1995;35:317–24.
[7] Qi Q, Liu X, Jiang X. A surface texture information integrated with AutoCAD for next generation GPS. Key Eng Mater 2008;381–382:237–40.
[8] Bui SH, Vorburger TV. Surface metrology algorithm texting system. Precis Eng 2007;31:218–25.
[9] Bui SH, Muralikrishnan B, Raja J. A framework for Internet-based surface texture analysis and information system. Precis Eng 2005;29:218–25.
[10] Bui SH, Gopalan V, Raja J. An Internet based surface texture information system. Int J Mach Tools Manuf 2001;41:2171–7.
[11] Sacerdotti F, Porrino A, Butler C, Brinkmann S, Vermeulen M. SCOUT—surface characterization open-source universal toolbox. Meas Sci Technol 2003;13:N21–6.
[12] Jung L, Kruger-Sehm R, Sparanger B, Koenders L. Reference software for roughness analysis, In: Proceedings of the EUSPEN, 2001, pp. S500–503.
[13] Li T. Softgauges for surface texture, In: Department of Computing & Engineering 2011, The University of Huddersfield, Huddersfield. p. 199.
[14] Wang Y. A knowledge-based intelligent system for surface texture (VIRTUAL-SURF), In: Department of Computing & Engineering 2008, The University of Huddersfield, Huddersfield. p. 232.
[15] Xu Y. The exploration of a category theory-based VirtualGPS system for design and manufacturing, In: Department of Computing & Engineering 2008, The University of Huddersfield, Huddersfield.
[16] Lu W. Modelling the integration between the design and inspection process of geometrical specifications for digital manufacturing, In: Department of Computing & Engineering 2012, The University of Huddersfield, Huddersfield. p. 146.
[17] Leach RK, Harris PM. Ambiguities in the definition of spacing parameters for surface-texture characterization. Meas Sci Technol 2002;13:1924–30.
[18] Scott PJ. The case of surface texture parameter RSm. Meas Sci Technol 2006;17(3):559–64.
[19] International Organization for Standardization, ISO 17450-1:2011 Geometrical Product Specifications (GPS)—General concepts—Part 1: Model for geometrical specification and verification. 2011.
[20] Srinivasan V. An integration framework for product lifecycle management. Comput-Aided Des 2011;43:464–78.
[21] International Organization for Standardization, ISO 1302:2002 Geometrical Product Specifications (GPS)—Indication of surface texture in technical product documentation. 2002.
[22] International Organization for Standardization, ISO 25178–2 Geometrical product specifications (GPS)—Surface texture: Areal—Part 2: Terms, definitions and surface texture parameters. 2012.
[23] Barr M, Wells C. Category theory for computing science. Prentice Hall International; 1995.
[24] Walters RFC. Categories and computer science. Carslaw Publications; 1991.
[25] Rossiter BN, Nelson DA, Heather MA. The categorical product data model as a formalism for object-relational databases: www.cs.ncl.ac.uk/publications/trs/papers/505.pdf, 1994.
[26] Lane SM. Categories for the working mathematician. New York: Springer-Verlag; 1971.
[27] Pierce BC. Basic category theory for computer scientists. Cambridge: MIT Press; 1991.
[28] Awodey S. Category theory. Oxford: Clarendon Press; 2006.
[29] Qi Q, Jiang X, Liu X, Scott PJ. An unambiguous expression method of the surface texture. Measurement 2010;43:1398–403.
[30] Qi Q, Jiang X, Scott PJ. Knowledge modeling for specifications and verification in areal surface texture. Precis Eng 2012;36:322–33.
[31] Rezayat M. Knowledge-based product development using XML and KCs. Comput-Aided Des 2000;32:299–309.
[32] www.open-gps.sourceforge.net.