Institut für Rechnergestützte Ingenieursysteme

Fakultät Informatik, Elektrotechnik und Informationstechnik

Universität Stuttgart
Universitätsstraße 38
D - 70569 Stuttgart

Master Thesis Nr. 3634

Modeling of an Automatic CAD-Based Feature Recognition and Retrieval System for Group Technology Application

Stanislav Moskalenko

Studiengang:	INFOTECH
Prüfer:	Univ-Prof. Hon-Prof. Dr. Dieter Roller
	Jun.-Prof. Dr.-Ing. Dimka Karastoyanova
Betreuer:	M.Sc. Leila Zehtaban
begonnen am:	11.11.2013
beendet am:	13.05.2014
CR-Klassifikation:	I.3.5, I.3.6, I.3.8, I.5.2, J.6

ABSTRACT

In recent time, many researches have come up with new different approaches and means for Computer-Aided Design (CAD) and Computer-Aided Manufacturing (CAM) integration. Computer-Aided Process Planning (CAPP) is considered to be a bridge that connects these both technologies. CAPP may involve such an important technique as automatic feature extraction – a procedure that is engaged in process plans generation to be used in producing a designed part. Also in terms of CAD, the feature extraction procedure facilitates a cooperative design and process planning within the entire product development process. The main objective of the thesis is to present a new automatic feature extraction and classification system that is able to process mechanical rotational and non-rotational parts from the Opitz Code System point of view. The implemented system takes Standard for Exchange of Product data (STEP) – a neutral product representation format as input and extracts features of parts required for further manufacturing. The STEP format is used to provide geometrical and topological information about machining parts. A methodology to extract shape features was developed based on these geometrical and topological data. As output, the proposed system codes the extracted part features to Opitz Code System. CAD product files were taken from official manufacturers of mechanical parts in order to evaluate the developed system.

TABLE OF CONTENTS

LIST OF FIGURES

INTRODUCTION

Background

The process of manufacturing has become more competitive in recent time in almost all markets. Due to the intensive interest of customers and technological advancement, manufacturers have to rapidly change and develop their capabilities in order to take a major share in the markets competition. This change had been made possible through transforming production system from the mass production to the production of a large product mix. Moreover the rapid advancement in technology leads the products to become outdated more quickly than before. As a result, companies came to realize that developing advanced methodologies for modeling, design, analysis, and performance evaluation, scheduling and control of these systems is vital for increasing the capacity of producing many small volume batches consisting of complex parts in a short production period. The transition to this new approach is not as simple as it may seem, it brings a lot of challenges which make not only the management's task more cumbersome, but also includes unwanted consequences such as an increase in production cost, and a decrease in efficiency of the mass production systems. One approach which has been proved to be most effective in solving these problems is the adoption of manufacturing approach which is known as Group Technology (GT).

GT is generally considered as a manufacturing philosophy or concept on the basis of which certain manufacturing efficiency can easily be improved when part types are identified and collected into groups (known as part families) based on their similarities in design or manufacturing attributes and machines that are required to process the part family into machine-cell. This results in an organization of the production system into self-contained and self-regulated groups of machines such that each group of machines undertakes a maximum production of a family of parts. Such decomposition of the manufacturing operation into subsystems leads to reduced material handling activities, reduction of production time and current amount of required inventory, reduction of setup time, reduction of order time delivery, reduction of unnecessary paper work and better supervisory control.

One of the fundamental requirements for implementing a GT based manufacturing system is having a developed Classification and Coding System (CCS). This coding scheme is used to classify the part or product and assign to it in accordance with the predetermined set of codes that relate to define physical or manufacturing characteristics. The CCS can also be used to organize part description to assist in the retrieval of parts and/or group parts according to the manufacturing process. Although it is a precondition for applying GT, a well-developed CCS on its own right can make a significant contribution to the improvement of manufacturing efficiency (such as effective design data retrieval, effective part family grouping, reduction of duplicated design, etc.). In current thesis as original CSS, Opitz Code System is adopted.

Problem statement

Into serious consideration an integration of different Computer Aided Systems such as CAD, CAM and CAPP has been put recently within the agile manufacturing environment. Thus, various methods have been proposed and investigated for the purpose of integration which includes feature recognition techniques, data processing algorithms, product data representation formats (STEP, IGES, etc.) and many others.

Feature recognition is one of the major challenges to achieve the objectives of CAD/CAM integration. Although the research and development of this methodology has been pursued since 1980s until now, still there are unsolved problems. Within modern CAD/CAM environments the constant growth of system complexity and product design abilities demands for newer feature recognition methods.

Retrieval and archiving of engineering product information by means of feature recognition techniques facilitates part reuse. It eases engineering activities such as new product design based

on archived product data sets. CAD knowledge bases are vital for engineers who search through amounts of corporate data and explore online catalogues to retrieve the appropriate components.
In the current thesis an automated feature extraction system presented, which takes as input STEP – a neutral product representation format; also the system recognizes required shape features to classify some specific detail according to Opitz Code System.

Objectives of the Thesis

The main objective of current thesis is to develop a system to recognize and extract features of rotational and non-rotational mechanical parts; after feature recognition the system should provide generation of a shape signature and part classification. More specific clauses:

- Review product shape representation formats and choose the most suitable for the problem of feature extraction;
- Research modern methodologies of feature extraction, shape signature generation and part classification, especially Group Technology with Opitz Code System;
- Develop a methodology that allows to retrieve all the required product features from the chosen shape representation in accordance with Opitz Code;
- Implement the developed methodology by means of Java Environment, providing process automation;
- Evaluate developed methodology.

Organization of the Thesis

Based on the defined research objectives, this thesis consists of introduction, conclusion and 6 chapters provided below:

- Chapter 1: Investigation and evaluation of the modern product shape representation formats;
- Chapter 2: Literature review of feature extraction techniques;
- Chapter 3: Literature review of STEP-based feature extraction methods;
- Chapter 4: Group technology and Opitz Code investigation;
- Chapter 5: The proposed feature recognition method that is able to extract features from STEP format and to classify input parts according to Opitz Code;
- Chapter 6: Implementation of the proposed method and its evaluation.

CHAPTER 1. SHAPE REPRESENTATION FORMAT CLASSIFICATION

Information and knowledge about an engineering part is mostly stored in its representation. The term "part representation" is defined in this document as the set of shapes, features and dimensions that coexist in a specific balance to meet physical and functional requirements. These requirements may be used for engineering part design, manufacturing, maintenance and even marketing.

The engineering data about particular part can be split by its roles into the following categories [1]: administrative (part identification, part structure), design/analysis (idealized models), basic shape (geometric, topological), augmenting physical characteristics (dimensions and tolerances, intrinsic properties), processing information, presentation information. In current work the main attention directed to the basic shape category, so the usage of "part representation" term implies only this category.

In a traditional design, parts are defined by engineering drawings and related data, but nowadays modern CAD/CAM environments store most drawings in an electronic form. Contemporary computer technology varies from 2D drafting systems to complicated solid editors, therefore the data proved to exist in many different formats. A common data format could enhance a cooperative part and process development between different environment users. The communication in this modern approach to computer systems that manufacture and inspect the part could be increased as well.

In the early years of CAD/CAM technology, software systems were developed with an employment of translators that transform data to support the variety of environments. These translators had some success, but the more vendors appeared in the market the harder was to provide a support for all of them. It leaded to introducing of some neutral data exchange formats with appropriate translators for them. Some of these translators were addressed to the specific industries and others were accepted as standards by general authorized organizations. Such neutral formats as Standard for Exchange of Product data (STEP), Initial Graphics Exchange Specifications (IGES), Data Exchange File (DXF) have gained more popularity among user communities. In the chapter below the overview of the entire area of shape representation formats for engineering parts given with special attention to the neutral formats as they proved to be more efficient.

Shape representation formats can be divided in 3 major groups: Native CAD, Neutral, Lightweight format group [2].

1.1 Native CAD representations

This kind of format is usually characterized with proprietary regulations that company-owner obliges to preserve. As a result a lack of documentation and format specification takes place. Nowadays the escape from native CAD formats can be seen, but there is still a major share of this format in overall engineering branch. Main drawbacks of native CAD representations [3]:

- Software proprietary
- Software subject to obsolescence (CATIA V4-V5-V6, etc.)
- Big file size. This is a domain dependent drawback: for some internet applications that require a high network throughput it can be crucial, but for others not so important.
- Limited abilities to support visualization and manipulation requirements for downstream processes and users (CAD systems might not be affordable for the entire development stream).

In spite of the mentioned drawbacks, one of the principal virtues of native CAD representations is the ability to preserve specific aspects of the engineering data, keeping comprehensive object information for later use. Main representations of this kind are DWG, CATIA and SolidWorks.

1.2 Neutral representations

Accurate, informative communications and collaboration among all of the participants in details manufacturing is a critical success factor. Companies are actively searching for effective approaches to carry, control, distribute and maintain the shape definition throughout the part lifecycle. Neutral format promises to help with solving this challenge. Neutral formats are based on international standards and are capable of expressing robust geometry representations.

Advantages: ability to keep explicit geometry, support downstream compatibility of 3D models.

Drawbacks: lack of security capabilities (passwords, encryption etc.), it takes time to overtake new features of CAD software releases, heavy file size.

The principal neutral representations are Standard for the Exchange of Product model data (STEP) and Initial Graphics Exchange Specification (IGES), Parasolid, ACIS, VDAFS, STL, VRML.

1.2.1 Standard for Exchange of Product model data (STEP)

Standard for the Exchange of Product Model Data or STEP, gives an opportunity to build a part data representation together with the mechanisms that enable the exchange of part information. This exchange takes place among different computer systems and includes data from complete product lifecycle: design, production, utilization, support. The part data generated during these stages is very useful. There are many computer environments can be included in this process, some of them may be located in different geographical regions. In order to support distributed cooperation, parties should be able to represent their part information in a predefined form that should stay consistent and complete in time of the exchange between computer systems.

1.2.1.1 STEP overview

STEP consists of a series of parts, each of them published separately. These parts fall into one of the following series: description methods, integrated resources, application protocols, abstract test suites, implementation methods and conformance testing.

Application protocols (APs) define one of the parts of STEP that belongs to Integrated Resources series. These APs use the low-level information of current series in form of combinations and configurations to represent a particular data model of an engineering or technical application. It is supposed that many APs (more than several hundred) can be developed to support different industrial applications. STEP uses an EXPRESS, a specification language, to specify product information that should be represented. The usage of such language provides accuracy and stability to product representation, facilitating implementations development. There is an addition to the STEP standard that enhances its implementation abilities: abstract test suits and conformance testing are built into this standard.

The main goal of STEP is to realize an ability to describe product information on all stages of product lifecycle in a system independent way. However, there is a time needed to reach this goal. The most tangible advantage of STEP to users today is the ability to exchange design data as solid models and assemblies of solid models.

STEP description methods suitable not only for neutral file exchange, but also for implementation and sharing of product databases and archives. One of the STEP objectives is to build an integrated product information database that is accessible and useful to all the resources that is necessary to support a product over its lifecycle [4].

1.2.1.2 STEP application protocols

Application Protocol (AP) is a domain specific set of rules representing a particular data model of an engineering or technical application. For instance, AP203 addresses 3D mechanical parts, AP210 electronic assemblies. Every AP has a scope that represents the content and the purpose of a particular Application Protocol. Having this information an engineer is able to see the

applied area of different APs and their conformance classes to choose the best solution that meets all user product data exchange requirements.
Nowadays the commercial implementations of STEP are still limited to a few Conformance Classes of AP203 and two conformance classes if AP214 (for the entire definition of Conformance Classes refer to chapter 1.2.1.3). The later one is an extension of AP203 and they both can be roughly treated as the equivalents.
The following list includes STEP Application Protocols that are active in current point of time.

AP	**Publishing date**	**Ballot stage**	**Title**
AP201	1994	International Standard (IS)	Explicit draughting
AP202	1997	IS	Associative draughting
AP203	1994	IS	Configuration controlled 3D designs of mechanical parts and assemblies
	1998	Technical Corrigendum (TC)	
	2000	TC	
	2004	Technical Specification (TS)	
AP204	2002	IS	Mechanical design using boundary representation
AP207	1999	IS	Sheet metal die planning and design
AP209		IS	Composite and metallic structural analysis & related design
AP210	2001	IS	Electronic assembly, interconnection and exchange
AP210 2ND		Draft International Standard (DIS)	
AP212	2001	IS	Electrotechnical design and installation
AP214	2001	IS	Core data for automotive mechanical design processes
AP214 2ND	2004	IS	
AP215	2003	IS	Ship arrangement
AP216	2004	IS	Ship moulded forms
AP218	2004	IS	Ship structures
AP219	2006	DIS	Manage dimensional inspection of solid parts or assemblies
AP221	2006	DIS	Functional data and their schematic representation for process plants
AP223	2006	Committee Draft (CD)	Exchange of design and manufacturing product information for cast parts
AP224	1999	IS	Mechanical product definition for process planning using machining features
AP224 2ND	2001	IS	
AP224 3RD	2006	IS	
AP225	1999	IS	Building elements using explicit shape representation

AP227	2001	IS	Plant spatial configuration
AP227 2ND	2005	IS	
AP229	2006	New Work Item TS Technical Specification (NWI)	Design and manufacturing product information for forged parts
AP232	2002	IS	Technical data packaging core information and exchange
AP233	2005	Approved Work Item (AWI)	Systems engineering data representation
AP235	2005	CD	Materials information for the design and verification of products
AP236	2005	DIS	Furniture product data and project data
AP238	2006	DIS	Application interpreted model for computerized numeric controllers
AP239	2005	IS	Product Life Cycle Support
AP240	2005	IS	Numerical control process plans for machined parts

Table 1: STEP application protocols

STEP continuous development brings more STEP standards annually to the state of finalization and stability. By now there are 22 Application Protocols that have received the status of International Standard (IS).

1.2.1.3 STEP Application Modules and Conformance Classes

In recent time various STEP organizations forwarded an initiative to develop STEP Application Modules (AM's) that are domain, or even complete APs, building blocks. In 2001 the initial set of Application Modules had been published. This attempt was intended to speed up the process of ISO standardization and was worldwide supported, especially by the user community.

Now there’s more of the technical data included in AMs than in the initial APs of ISO 10303. The role of APs now is to provide a business context, when from the side of AMs there are implementations of AP data specifications.

AMs can be divided in 3 module groups: 1 level foundation modules, 2 level implementation modules, AP modules. Foundation modules provide low level reusable blocks that are highly sharable. Implementation modules include information that allows conformance classes to be defined. Each AP references a single root module that is an AP module. The AP module from one AP may be used by another AP. Contents of an AP module are the same as other AP modules, there’s only one difference in their name and title [4].

Each AP has Conformance Classes (CC, cc) associated with it. These are the subsets of APs that can be used in accordance with the provided application domain, having no need to implement the entire stack of the current AP. As an example, implementations of Conformance Classes can be seen in APs that have been already commercially implemented: AP203 and AP214.

It is important to indicate what Conformance Classes have been used when STEP is applied. Providing some AP as a translator or just STEP as a protocol is not sufficient, there is a need to indicate CC as well. An engineer needs to know what Conformance Classes of APs exist and to see their coverage. For example, AP203 has 12 Conformance Classes: from 1a,b to 6a,b. Very few developers who have used AP203 as domain descriptor implemented cc5; the most of them have cc 2a, 4a and 6a implementations, providing minimal, but acceptable Conformance Class 1a – a subset of Configuration Management data (for comprehensive examples of AP203’s CCs refer to chapter 1.2.1.4). Developers claiming the usage of an AP214 domain descriptor have

only implemented cc1 and cc2 that are basically identical to AP203 geometry/topology with the difference in Configuration Management data. AP214 has 20 Conformance Classes and these CCs cover almost the entire area of automotive design.
It is ambiguous for Vendors to claim that they have implemented AP214 without support of the Conformance Classes. Currently there are no commercially available AP214 translators that use Conformance Classes other than the ones of AP203. But it should be said that a few Vendors have already developed the conformance class prototypes of the PDM Schema: AP214 cc6, AP214 cc7. Here also should be noticed that a general effort was initiated to harmonize the PDM Schema with those APs that addressing PDM: AP203, AP209, AP214 and AP232 [4].

1.2.1.4 STEP AP203 closer look

Currently, the most widely used AP is the IS AP203 which is designed for representing 3D geometry and configuration management information [5]. It is named as "Configuration Controlled 3D Designs of Mechanical Parts and Assemblies" (ISO 10303-203). Below there are a scope and conformance classes of AP203 in order to give a better overview of this Application protocol.
AP203 Scope main clauses [4]:

- Five types of shape representations of a part that include wireframe and surface without topology, wireframe geometry with topology, manifold surfaces with topology, faceted boundary representation, and boundary representation;
- Products that are mechanical parts and assemblies;
- Product definition data and configuration control data pertaining to the design phase of a product's development;
- The change of a design and data related to the documentation of the change process;
- Identification of government, industry, company or other specifications for design, process, surface finish, and materials which are specified by a designer as being applicable to the design of the product;
- Data that are necessary for the tracking of a design's release;
- Data that is used in, or results from, the analysis or test of a design which is used as evidence for consideration of a change to a design.

AP203, Edition 1 has 12 Conformance Classes [4]:

- cc 1a, b: Configuration controlled-design information without shape (cc 1a is a specified "product identification" subset of cc 1b)
- cc 2a, b: cc 1a, b and 3D geometrically bounded wireframe and/or surface models
- cc 3a, b: cc 1a, b and 3D wireframe models with topology
- cc 4a, b: cc 1a, b and manifold surface models with topology
- cc 5a, b: cc 1a, b and faceted B-Rep
- cc 6a, b: cc 1a, b and advanced B-Rep

As it can be seen that Conformance Classes of AP203 include such information as 3D shape description (bounded wireframe, surface, B-Rep). By default AP203 stores 3D data as a B-Rep format, structure of which is provided in Fig. 1. Root element 'Solid' contains the complete definition of the 3D model geometry and topology. The outer extent of this solid is defined by a closed shell. Closed shell consists of faces, which are defined by advanced face. Then every face is represented by outer loops and inner loops which are, from their side, defined edge loops. An edge loop consists of oriented edges. Oriented edges in turn consist of edge curves which are represented by vertex points and edge geometry (vector direction, start point, etc.). Current geometric data can be used on later stages, for feature extraction as an example.

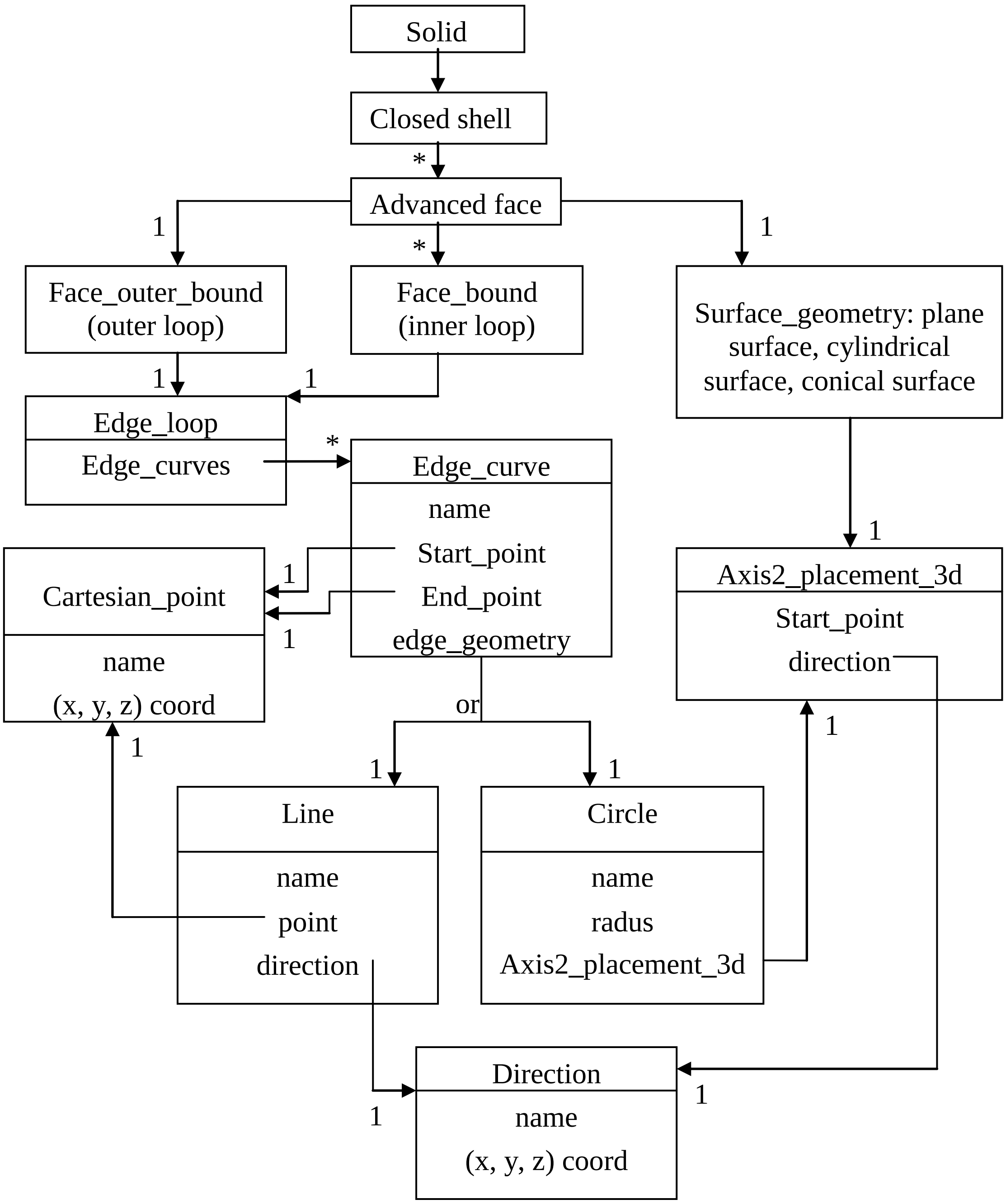


Fig. 1. AP203 entities structure

1.2.1.5 STEP Parts 21 and 28

Part 21 and part 28 are other important pieces of STEP protocol which are widely used form of STEP Implementation methods. They allow saving and keeping the product geometry/topology in a predefined file format inside a desired data storage.

Part 21 that defines a structure of a STEP-File is the most widely used data exchange form of STEP protocol. This part is addressed as “ISO 10303-21 Clear Text Encoding of the Exchange Structure”. Mentioned STEP-file is a full implementation of AP203 entity structure model that depicted in Fig. 1. The file of this type is highly readable due to its ASCII nature and typically read line by line. File extensions *.stp and *.step indicate that the file containing data is compliant to STEP standard. ISO 10303-21 defines the encoding mechanism on how to represent data according to a given EXPRESS schema, but not the EXPRESS schema itself.

To give an example of STEP file format, a test detail (cube) was given as an input to the CAD AP203 system translator software. After this, Part 21 file for the current detail was generated.

This file includes the configuration management information, advanced boundary representation structures, geometric validation properties and so on. There is a fragment of the generated file given below, which reflects a few important geometric entities in correspondence with Fig. 1. STEP-File consists of ordered by its number lines, but in current fragment the lines are provided in a mixed order for better clearness.

#100 = EDGE_LOOP ('NONE', (#73, #75, #74, #76)) ;
#73 = ORIENTED_EDGE ('NONE', *, *, #38, .T.) ;
#38 = EDGE_CURVE ('NONE', #98, #92, #122, .T.) ;

#98 = VERTEX_POINT ('NONE', #172) ;
#172 = CARTESIAN_POINT ('NONE', (-21.17487599707601500, 0.0000000000000000000, 34.37326996612455300)) ;

#122 = LINE ('NONE', #123, #230) ;
#230 = VECTOR ('NONE', #124, 1000.000000000000000) ;
#124 = DIRECTION ('NONE', (1.0000000000000000000, 0.0000000000000000000, 0.0000000000000000000)) ;

As can be seen EDGE_LOOP consist of EDGE_CURVEs that are formed by VERTEX_POINTs and VECTORs. This is a sufficient 3D geometry description model.
Inside the STEP-file CLOSED_SHELL is a root entity that should be first identified in order to start feature extraction process. It has a set of links to ADVANCED_FACE entities. ADVANCED_FACE has a topological sense; it describes a set of inner loops placed within one outer loop on the same surface Fig. 2.

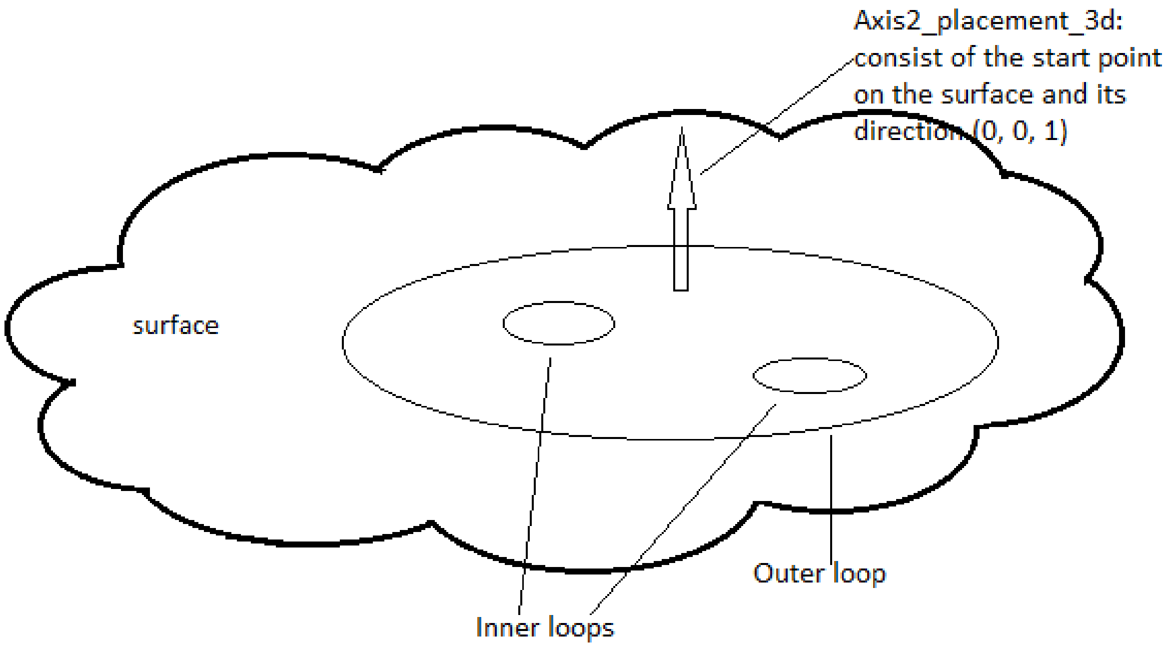


Fig. 2. Advanced face example

Also ADVANCED_FACE must contain one of the surface geometry entities: CYLINDRICAL_SURFACE, CONICAL_SURFACE or PLANE, each of them include AXIS2_PLACEMENT_3D entity that represents one normal (start point plus direction) to surface as shown in Fig. 2.

It should be mentioned that there is a three-dimensional XYZ space, that means: CARTESIAN_POINT ('NONE', (0.0000000000000000000, 0.0000000000000000000, 0.0000000000000000000)) is defined at the begining (0, 0, 0) of the coordinate system, DIRECTION ('NONE', (0.0000000000000000000, 1.000000000000000000, 0.0000000000000000000)) is pointed strait to top from XZ-surface as depicted in Fig. 3.

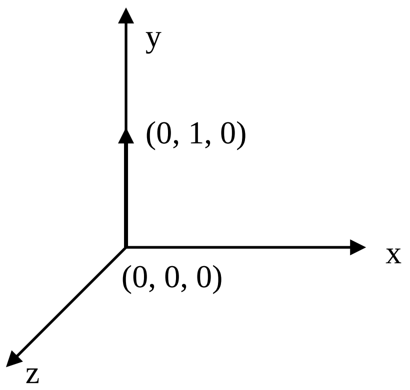


Fig. 3. Direction (normal to surface XZ) example

Every FACE_OUTER_BOUND and FACE_BOUND has a one-to-one relation with an EDGE_LOOP entity. The latter one represents a set of adjacent EDGE_CURVES entities forming shape boundaries on a surface, Fig. 4. An EDGE_CURVE entity always includes start point, end point and its edge geometry. When the edge geometry is a LINE entity, a direction and a start point of a vector is inscribed within this entity. And when the edge geometry is a CIRCLE entity, an AXIS2_PLACEMENT_3D (with a vector starting from the point on the top middle of an arc) entity and a radius of the arc is inscribed within this entity. The directions in both last cases help to decide an orientation of a particular edge that is useful for edge curves relationship calculations (e.g. angles between lines).

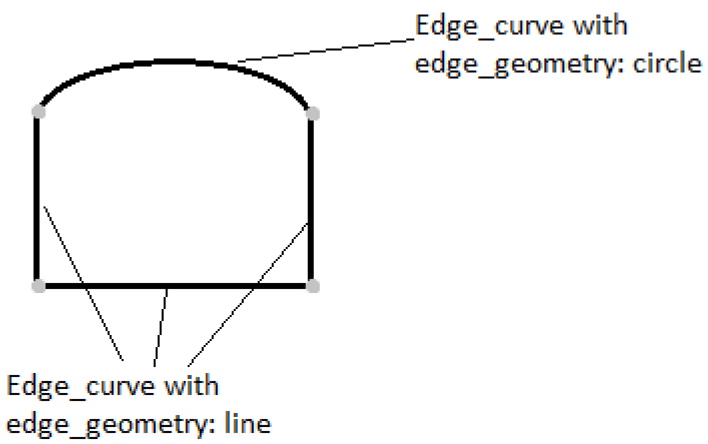


Fig. 4. Four edge curves forming one edge loop – a reflection of a shape

All entities inside the STEP-file shape description model can have only one parent, e.g. some ADVANCED_FACE entity can be referred only by a single CLOSED_SHELL entity (not

others). There are two exceptions from this rule: EDGE_CURVE and CARTESIAN_POINT entities may have multiple parents allowing elements sharing, when two intersecting surfaces have one common edge and common points along this edge.
Part 28 needs a special attention. It provides a representation of data according to the syntax of Extensible Markup Language (XML) defined using ISO 10303-11 (the EXPRESS language) and/or for EXPRESS schemas. Nowadays XML language is extremely popular technology; it is used worldwide in almost every IT branch by various vendors. That is why STEP developers also have supported an implementation of STEP in XML format. The mappings in the Part 28 are specified by the EXPRESS language. Any EXPRESS schema and the data it describes can be represented by its format. The original Part 28 was subsequently split into two parts: a revised Part 28 and a Part 25. They are both being developed as Technical Specifications.

1.2.1.6 STEP application domains

Within STEP specification there are Application Suites are provided which address to general application domains. The following Suites, in contrast to a single Application Protocol, employ a series of Application Protocols. As an example can serve: the Shipbuilding Suite, the Electromechanical Suite, the Process Plant Suite, the System Engineering Suite, the Engineering Analysis Core Model, Product Life Cycle Support and the Manufacturing Suite [4].
The share of STEP awareness continues to grow that leads to the gain of its industrial acceptance in such spheres as automotive industry, defense industry, aerospace and ship building industry. Now companies and vendors have started to treat STEP as an instrument of defining product information together with storing. Also STEP increases popularity recently by means of active support from the aircraft and the automobile industries. The overall amount of STEP-based applications keeps on rising for the last years [6].
Production implementations of STEP.

- CSTAR, C-17 STEP Transfer and Retrieval. Went through production in 1995 at McDonnell Douglas (Boeing) having AP203 cc1.
- AEROSTEP/PowerSTEP (Boeing). Went through production in 1995 with Rolls Royce (Catia/CADDS5 - AP203 cc6) - Went through production in 1996 with General Electric and Pratt & Whitney (Catia/UG - AP203 cc6) - In 1997 entered into agreement with Rolls Royce, General Electric, and Pratt & Whitney to exchange data using STEP AP203 to support digital preassembly verification for the 777 and 767-400 aircrafts.
- General Motors STEP Translation Center. Went through production in 1996 to test and validate surface and solid model data exchange. Extensive STEP/IGES comparison analysis. CATI/UG translation services with GM Powertrain, Delphi/Delco Electronics, and Delphi Automotive divisions.
- Lockheed Martin - Tactical Aircraft Systems. Went through production in 1998 with the use of CATIA STEP AP203 translators for data exchange on the F-16, JSF, F-22, KTX-2, and F-2 aircraft Programs. In 1999, Lockheed Martin-Tactical Aircraft Systems (LM-TAS), undertook the Virtual Product Development Initiative for Finite Element Analysis (VPDI-FEA) using AP209 DIS.
- NASA. The policy that STEP Translators are required to be available at all NASA Sites stated.

1.2.2 Initial Graphics Exchange Specification (IGES)

IGES is a neutral format aiming to define the product representation data by means of solid modeling with B-Rep structure. It addresses a wide range of application domains including electrical, plant design, as well as mechanical applications. IGES includes a format by which the user transfers the data among different CAD systems. To perform such data exchange, IGES requires two levels of processing: a pre-processor that takes some CAD data as described in the system specific format and converts it into IGES format; a post-processor that converts data from IGES back to some CAD-system format [7].

IGES includes data entities that are used to describe and represent the product definition data. These entities are stored in a domain-independent manner that enhances the product representation exchange capability across the different CAD-systems. As can be concluded the fundamental information unit in an IGES is the entity. The description of the product goes in terms of geometry, non-geometry and topology entities. The geometrical entities provide a definition of the object physical shape; they include points, curves, solids, surfaces and relations that form a bunch of similar entity sets. Non-geometry entities define a view perspective of the object by means of annotations and dimensions. These entities may also include view, drawings, text, notation, witness lines and leaders descriptions. Topology entities define the relationships between the geometrical object primitives (edges, vertices, loops, etc.).
The IGES-file consists of 80 column lines that are grouped into 5 or 6 sections. Each section has its own code as shown in Table 2.

Section name	Col.73 Letter Code
Start	S
Global	G
Directory Entry	D
Parameter Data	P
Terminate	T

Table 2: IGES-file sections [1]

IGES data can persist in either an ASCII or a binary format. The code identifying a section is placed in 73 column of each line. In columns 74-80 the sequence number of every line within a section is stored. This sequence number is a seven digit number starting from 1, sequentially increasing by 1 with leading space or leading zero fill in accordance to the line number. Columns 1-72 store the data specific to the entity. General file structure is depicted in Table 3.

<table>
<tr><td>1 8</td><td>9 16</td><td>17 24</td><td>25 32</td><td>33 40</td><td>41 48</td><td>49 56</td><td>57 64</td><td>65 72</td><td>73 80</td></tr>
<tr><td colspan="9">Start Section – a human readable prologue to the file
It contains one or more lines
:
:
using ASCII characters in columns 1–72.</td><td>S0000001
S0000002
S0000003
…
S000000N</td></tr>
<tr><td colspan="9">Global Section – sending system and file information.
It contains the number of lines needed to hold the parameter fields, separated by
:
:
parameter delimiters, and terminated by one record delimiter, in columns 1–72.</td><td>G0000001
G0000002
G0000003
…
G000000N</td></tr>
<tr><td colspan="9">Directory Entry Section – contains one pair of lines for each entity.
Directory entry fields 1-9 in nine 8-column-wide fields
Directory entry fields 10-18 in nine 8-column-wide fields</td><td>D0000001
D0000002</td></tr>
<tr><td colspan="7">Parameter Data Section – values and parameter delimiters terminated by one record delimiter, in columns 1-64; column 65 is unused</td><td colspan="2">DE back pointer</td><td>P0000001
P0000002</td></tr>
<tr><td colspan="5">S0000020 G0000003 D0000500 P0000261</td><td colspan="4">Terminate Section – record counts for preceding sections; columns 33–72 unused</td><td>T0000001</td></tr>
</table>

Table 3: IGES-file structure [1]

1.2.2.1 IGES file sections. A closer look.

Flag section

This section is optional and is used to indicate whether the file is of binary format or of the compressed ASCII format.

Start section

The start section is supposed to provide a human-readable prologue to include some engineer information. There should be at least one record in this section. Every record within this section is preceded with an S letter that is stored in column 73 and a sequence number in the range of 74-80 columns. There are no any special format rules within 1-73 columns to simplify the user data input. This section includes such information as the ids of the sending and receiving CAD-systems, as well as short description of exchanged product.

Global section

Current section stores a data that is required to support pre-processor and post-processor operations. Each record within this section identified with letter S in column 73 and is sequenced in columns 74-80. The parameters for the Global Section are written as delimited, variable-length field values and describe delimiter characters, record delimiters, product ids, file names, native system ids, preprocessor versions, units, date and time of file creation [1].

Directory entry section

The directory entry section aims to provide the index for the file and to store the attribute information for every entity. The order of the records within current section is arbitrary. Some of the fields in the Directory Entry may contain either an attribute value or a pointer to an entity containing one or more such values. In these fields, a positive value corresponds to an attribute; a negated value indicates that its absolute value is a pointer to the Directory Entry of an entity containing one or more attribute values. As depicted in Table 4, the attribute 1 and the attribute 11 contain the record number; attribute 2 is a pointer to the record that is presented in the Parameter Data Section (see below). The legend of Table 4: (n) – field number; # - an integer; → - a pointer; #,→ - an integer or a pointer; 0,→ - zero or pointer.

1 8	9 16	17 24	25 32	33 40	41 48	49 56	57 64	65 72	73 80
(1) Entity Type Number #	(2) Para- meter Data →	(3) Struc- ture #, →	(4) Line Font Pattern #, →	(5) Level #, →	(6) View 0, →	(7) Trans- form. Matrix 0, →	(8) Label Display Assoc. 0, →	(9) Status Num #	(10) Seq- uence Num D #
(11) Entity Type Number #	(12) Line Weight Number #	(13) Color Num #, →	(14) Para- meter Line Count #	(15) Form Num #	(16) (17) Reserved		(18) Entity Label	(19) Entity Subscr . Num #	(20) Seq- uence Num D# +1

Table 4: Attributes of an entity of Directory entry section

In Table 5 the main IGES entities are shown.

Entity number	Entity Type
110	Line
100	Circular arc
124	Transformation matrix
120	Surface revolution
116	Point
123	Direction
190	Plane surface
502	Vertex list
404	Edge list
508	Loop
510	Face
514	Shell
192	Right circular cylindrical surface

Table 5: A list of basic IGES-file entities

Parameter data section

Current file section contains the parameter data associated with each entity. Parameter data is formatted in a free way with the first field that always contains the entity type number. The entity type number and a parameter delimiter precede Index 1 of each entity in the exchange file. The formatted in a free way part of a parameter line ends in Column 64. Column 65 contains a space character. Columns 66 through 72 on all parameter lines contain the sequence number of the first line in the Directory Entry of this entity. Column 73 of all lines in the Parameter Data Section contains the letter P and Columns 74 through 80 contain the sequence number as shown in Table 6.

1 64	66 72	73 80
Entity type number followed by parameter delimiter followed by parameters separated by parameter delimiters	DE pointer	P0000001
Parameters separated by parameter delimiters followed by record delimiter	DE pointer	P0000002
:	:	:

Table 6: Parameter Data section [1]

Terminate section

Current section is defined by one line only that signals the end of file. It includes some summary information such as number of lines within each section.

1.2.2.2 IGES drawbacks

IGES is essentially a neutral format that enables the product data representation exchange; therefore it has a clear advantage over native translators. Although it is really overall used and employed in various industries, IGES has several drawbacks. It does not have a formal data model that causes ambiguities in some cases. As it was shown its 80-column files are very verbose; it is difficult to understand them because of their complicated structure. For the same reason if there is a syntax error in IGES-file, it is hard to find and correct it. Errors may appear due to the changes made to the file.

The lack of any conformance requirements leaded to IGES file deviations among different vendors. In this way from time to time inconsistent state takes place which implies incomplete translation and information losses.

IGES provides only drawing or 3D modeling data. In current work the main interest is focused exactly on these kinds of data, but for example in process planning domain the manufacturing view (form features and the manufacturing information) of the part would be of higher demand. Each application domain would share the same IGES data and it would be left to the individual interpretation to extract the relevant information. Consequently IGES does not provide life-cycle related data and thus it is not much extensible because of the lack of any integrated resources [6].

1.3 Lightweight representations

Lightweight representations are details model formats that are missing some of the richness of a traditional CAD model. The major characteristic of lightweight representation is a reduced file size via compression techniques, platform/application independence, open source and support for the protection of sensitive information. In addition, they can read and display 3D annotations.

Advantages: small file size that minimizes storage and network requirements, recent capabilities (solid geometry interpretation, 3D annotations and mark-up notes support, etc.), they offer data encryption mechanisms, CAD independence.

Drawbacks: conversion is required, shape representation data losses.

1.4. Conclusion

Native CAD representation appeared as bulky and proprietary shape formats; their specifications are closed that contradicts with the ideas of effective information exchange. On the other hand lightweight representations imply a data loss that is not suitable within current work; only precise geometrical data can be used for feature extraction. So the choice fell on the neutral format group.

Further the neutral format group was analyzed with figuring out 2 market leaders: IGES and STEP format. After a closer look IGES proved a few crucial drawbacks for the problem in this master thesis. Thus, STEP was selected for reasons of evident strengths and overcoming mentioned issues.

As a summary STEP ISO 10303-AP203 cc5a (or cc6a) appeared as the most suitable protocol for shape data description and exchange.

CHAPTER 2. SHAPE FEATURE EXTRACTION TECHNIQUES

Nowadays the high importance of feature technology within CAD/CAM sphere is generally accepted. This acceptance is approved by the existence of facilities for feature modeling and feature recognition in all major commercial CAD/CAM systems. Features are obtained within these parts by either designing with features, either by feature extraction (i.e. recognition) or by means of interactive form of feature definition [8]. Design-by-feature demands the existence of a form feature library, accommodated to part manufacturing requirements. Shape model is formed by using only form features from the library. Automated feature recognition comprises browsing some type of part representation aiming to find information, which characterizes singular form feature types. All approaches in this field have a unique goal: to form an algorithm capable of recognizing any possible type of form feature, without any interfering of manufacturing engineer. Interactive form feature definition is a system in which user selects a form feature set, determines recognition parameters to those features and then the system, using those instructions, performs an automated recognizing, whether directly in CAD model of the part, or in some structure developed from it. This research focuses on the use of automatic feature recognition for obtaining features from three-dimensional computer aided design (CAD) models.
The scientific world engaged in feature recognition has proposed many feature detection methods [8, 9]. Here is a general classification:

1. Syntactic pattern recognition methods
2. Rule-based methods
3. Graph-based methods
4. Convex-hull volumetric decomposition (volume decomposition)
5. Cell-based volumetric decomposition
6. Hint-based methods
7. Hybrid methods

In following sections the more detailed overview of proposed methods is presented.

2.1 Methods of Syntactic pattern recognition

In current group of methods, a model of the part is formed using semantic primitives written in some description language. A set of grammar, which consists of some rules, defines a particular pattern. The parser for input sentence analysis has been then used to apply a grammar to the part description (entities connected to form a part). If the syntax agrees with the grammar, then the description can be classified in a corresponding class of forms (pattern). There are three components of pattern recognition, Fig. 5. Input string represents semantically unknown grammar. Form semantics will be recognized if it can be classified in a group of predefined forms (pattern). Classification is made through form syntax lookup. Pattern syntax is also defined using grammar.

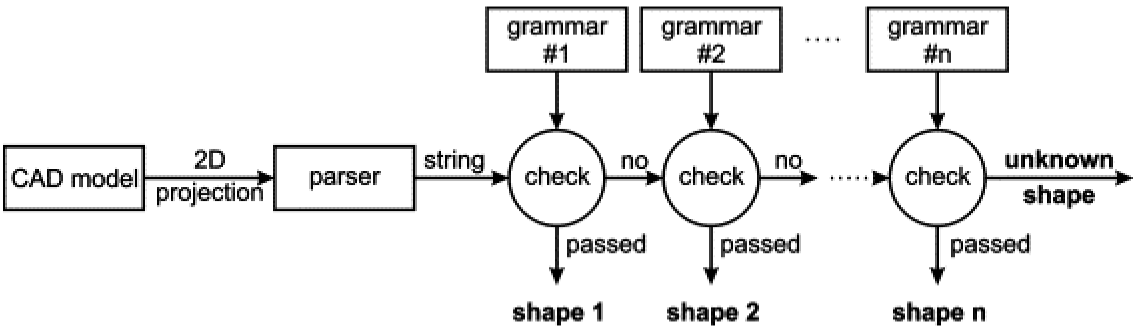


Fig. 5. Syntactic pattern recognition convey

Syntactic pattern recognition requires form primitives to be defined, and an automated translation of design model suitable for syntax analysis (string) to be provided. This is the earliest concept of AFR, introduced by Kyprianou [10]. This method has been more often and more successfully implemented for 2D form feature recognition. When used for 3D feature recognition it had to be previously translated in 2D part model.

Jane and Kumar presented one of these systems in [11]. The system takes a wire frame part representation model, imported from AutoCAD *.dxf file. It is developed for prismatic parts - a case not so often met in practice. The wire frame model (3D) is translated in a vertices-edges graph (2D), for each one of six boundary planes of a parallelepiped. This system provides the recognition of several form features: hole, step, slot and protrusions with orthogonal boundary faces. Hole is a basic feature, and all other are derived from it: steps are holes without two faces, slots are holes without one face, and protrusions are treated like a combination of slot and step manufacturing. In extension, graphs are translated in strings of shape primitives, using methodology illustrated in Fig. 6, and then strings are matched with the patterns in a knowledge base.

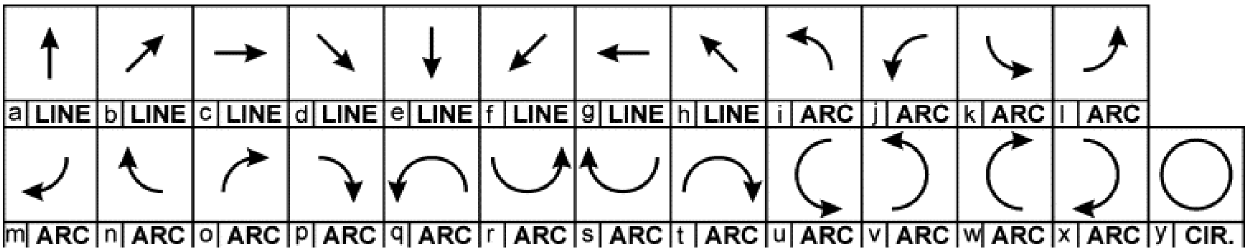


Fig. 6 Shape primitives of PRIZKAPP approach [11]

A more recent technique that may be included in syntactic pattern recognition group is so-called Edge Boundary Classification (EBC), presented by Ismail et al., in a series of papers (2002–2005). It considers the use of spatial addressability information of solid models that identifies the solid and void sides of a boundary entity. For each edge loop identified in B-rep model of the part, an EBC pattern can be formed from the result of classifying a set of test points (located at a close proximity to the edges that form the loop) with reference to the solid model. Depending on whether these test points qualify to solid or outer space, they are coded, and the string of these codes for each edge in the loop forms the pattern that can be used for form feature recognition. This approach can be applied for recognition of some features in parts: pockets, slots and steps consisting of planar and/or semi-cylindrical faces and, also, for cylindrical and truncated conical features, both in prismatic and rotational parts. The advantage of this technique is that it has the ability not to be affected by geometric and topological changes, except by those affecting primary faces. This implies that no post-processing is required, not even in case of interacting features. Its shortcoming is in complicated pre-processing for form feature identification: extraction of all relevant geometric and topological data, presentation in a format suitable for use by the EBC algorithms, establishing spatial addressability information and EBC patterns creation.

The main shortcoming of the syntactic pattern recognition is its area of application, limited to 2D prismatic parts, rotational parts with turning features and axis symmetric volumes. The implementation of this technique in the systems for non-axis symmetric 3D part or rotational parts with non-turning features has not been very successful, mostly due to restrictiveness of pure syntactic representation.

2.2 Rule-based methods

This approach was firstly introduced by Henderson and Anderson [12] . A set of production rules, IF C1, C2, C3 … Cn THEN A, which define form features, provide the patterns for automated feature recognition. If the conditions (C1, C2, C3) predefined by some pattern are satisfied, then the structure in the part representation that agrees with them is recognized as a corresponding form feature A.

The common approach for systems based on current group of methods is the following: a model of the part is translated from 3D solid modeler into IGES format (Fig. 7a). Then, using special utility program, IGES data are converted into Prolog facts. The first stage of the recognition process is face extraction and base faces (base face is a feature face which has a concave adjacency with at least one feature face) determination. Then, the boundary faces are being determined (Fig. 7b). The main criterion for a form feature matching (except for the holes) is the number and type of boundary faces. For faces satisfying the same basic conditions, additional conditions are introduced. The features that system may recognize are pocket, slot, blind slot, step, corner step, hole, blind hole, countersink hole. A very similar system, also applying logic rules to a set of data obtained from neutral IGES file, aimed for CAPP of prismatic parts was developed by Bouzakis [13].

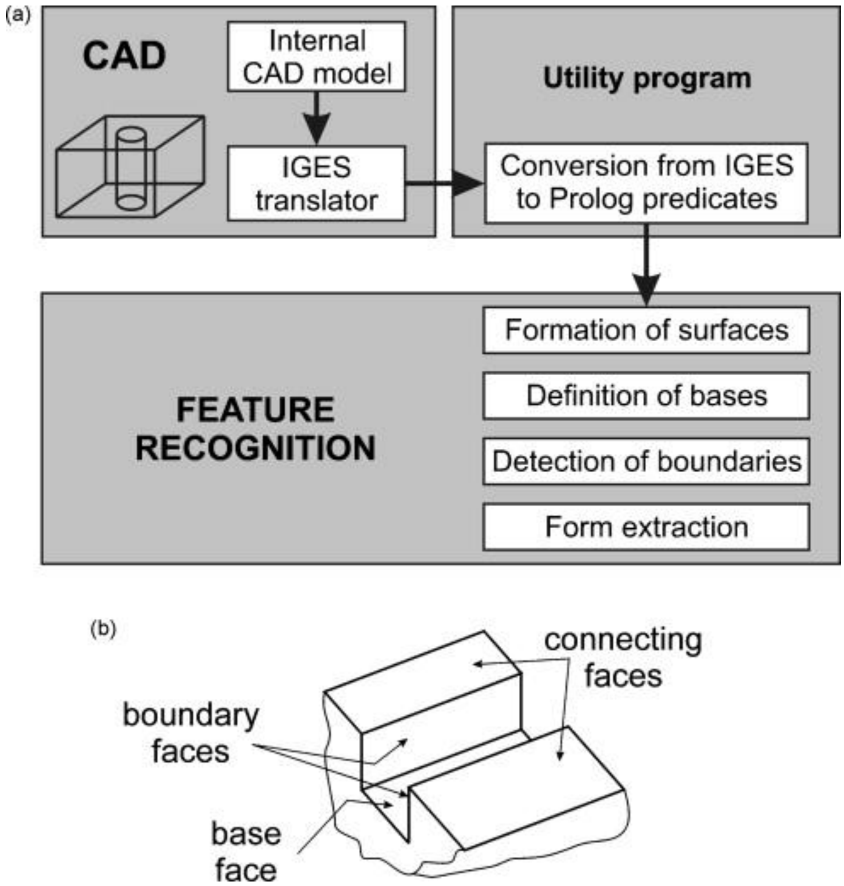


Fig. 7 (a) A CAD/CAPP interface model; (b) definition of the faces a form consist of

One more example of logic approach implementation is given in [14]. A CSG part representation (less common in research practice) is used for geometric feature extraction, which is performed by browsing the *.txt file in SolidWorks application protocol interface, using a program written in Visual Basic. Recognition is performed by matching extracted forms with the patterns in Oracle database. The system is designed for the recognition of numerous types of forms that occur in prismatic parts, but it is not capable to recognize intersecting form features. Recognized forms are then used for automated process plan generation.

Rule-based method, as a kind of generalization of syntactic method, has proved to be more robust and handling more kinds of parts than syntactic method. Ambiguous representation and predefined rules needed for every conceivable feature, make rule-based systems overloaded and inflexible.

2.3 Graph-based approach

Current approach was firstly developed by Joshi in 1987, aiming to form such part representation in which a topological information and some geometric information of the part will be included. The authors proposed an attributed adjacency graph (AAG) in which B-rep model of the part (designed in some solid modeler) is transformed. AAG is a graph in which every arc takes attribute “0”, if its nodes have a concave adjacency relation, or “1”, if they have convex adjacency relation that depicted in Fig. 8a, Fig. 8b. Form features represent subgraphs of part AAG and form feature recognition becomes a process of finding such subgraphs that can be matched with the patterns from the database. Subgraphs are analyzed by using logic rules, called recognizers. Such an approach is called subgraph isomorphism and represents a long-term and computationally demanding process of AAG structure browsing. The alternative method is graph partitioning/graph isomorphism [8]. Extraction is performed by parsing the AAG in nodes which have all adjacent faces convex (all arcs converging have attribute “1”), Fig. 8c. AAG concept, in its original form, suffered from two major shortcomings: possibility of application only for negative, polyhedral objects (polyhedral shaped intrusions, without curved faces) and impossibility of extraction of boundary faces, but only basic faces. The problem of extraction of the faces which are connected with one more face only (like top surface of cylindrical protrusion), or with two more surfaces (cylindrical protrusion envelope) has been discussed by researchers and solved by using special approaches.

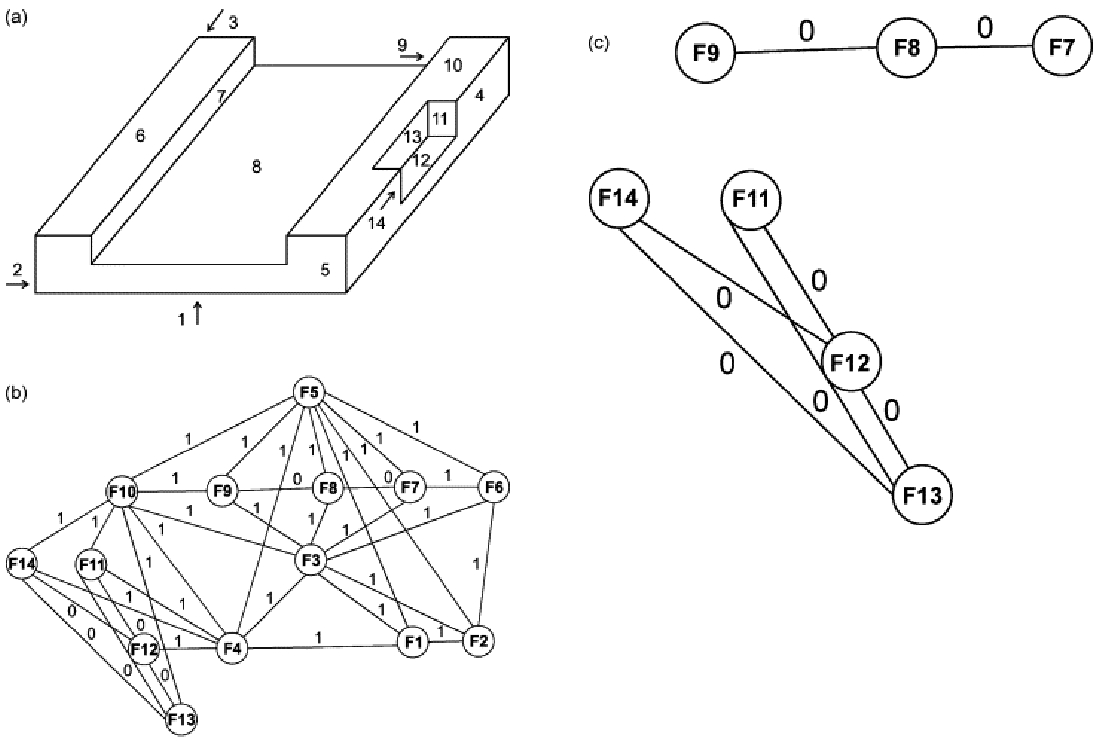


Fig. 8. (a) Sample part; (b) Attributed graph of the part; (c) its subgraphs [8]

The drawbacks of the AAG can be significantly lowered through the concept of multi-attributed adjacency graph (MAAG), assigning the attributes which more precisely descript adjacency relations (e.g. if plane and curved face make a convex angle of 270, then the attribute is “2“). If MAAG is represented with a matrix, then such system is called multi-attributed adjacency matrix — MAAM. The recognition process is performed over adjacency matrix schemes, which are predefined for each elementary form.

One of the numerous systems designed upon MAAG implementation was introduced by Venuvinod and Wong [15]. This AFR system is particular in following: B-rep of the part, whose elements were extracted from CAD model, is formed using so-called EWEDS data structure. EWEDS (enhanced winged-edge data structure) presents an enhanced version of the ”winged-edge data structure”, emerged by Baumgart in 1974, Fig. 9. Winged-edge data structure (WEDS) is an edge-based data structure, which provides information of the object's faces, edges and vertices in an explicit way. A pointer connects each marked face to each of its boundary faces. Likewise, there is a pointer to each of its bounding vertices (start and end). Every edge occurs in two faces exactly, once in the clockwise and once in the counter-clockwise orientation from object's outside aspect. In EWEDS a new level of data is added, relating additional information about edges, faces and vertices. In these data, that facilitate recognition process, the most important information relate to the type of each face, and convexity/concavity of the angle it makes with the adjacent faces.

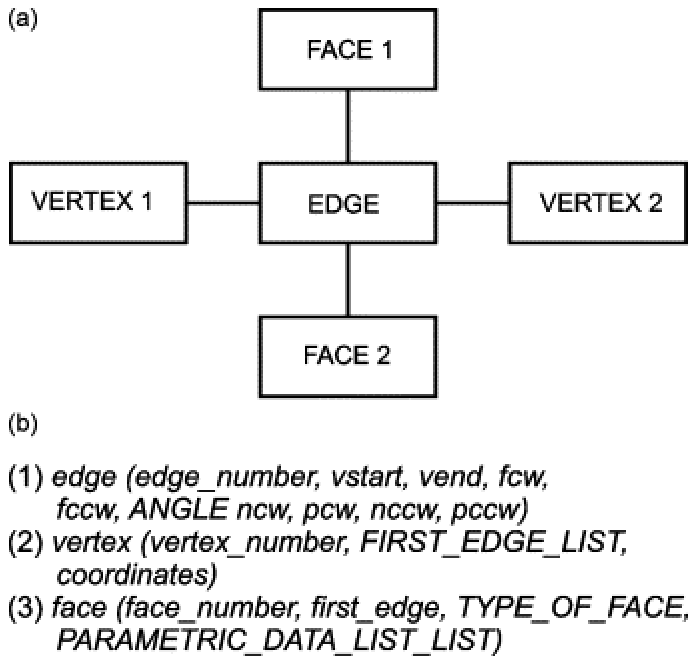


Fig. 9. (a) WEDS; (b) EWEDS (Prolog facts) with additional information (upper case letters) [8]

More recently, Venuvinod, Wong and Yuen make an experiment with MAAM concept [16] and seek for “less expert-system and more algorithmic” way for form pattern recognition. They invented a detailed coding system for description of so-called “primitive template feature” identified in EWEDS B-rep. These features, which by definition cannot be further decomposed in a reasonable way, are used for identification of feature relationships. The process results with a multi-layered representation of a part containing feature relationships on the first level, primitive template features and their variations on the second, face-edge MAAG on the third, EWEDS B-rep structure on the second and CAD file on the fifth level. This structure gives an opportunity for higher levels (the first and the second) to be CAPP domain-specific, based on the same geometric reasoning in lower levels, thus avoiding unnecessary repetition of geometric analysis for each of the CAPP domains.

The most of graph-based AFR systems had the problem of interacting features. Marefat and Kashyap [17] were the first to try and solve this problem by restoring the missing arcs between the nodes in part's graph representation (which stand for the edges between the faces which disappear when two or more features interact, Fig. 10). However, this approach often led to wrong set of missing arcs, due to the uncertainty reasons [18]. While features interact, uncertainty develops as a result of non-uniqueness of the patterns associated with the topology and geometry of features in these interactions. Various techniques have been developed to investigate available amount of geometric and topologic evidences, which if used correctly, can lead to the resolution of the uncertainty and therefore eventual recognition of the features. Two universal techniques for handling uncertainties and finding the exact set of missing arcs have often been applied in graph-based AFR systems: the Dempster-Shafer theory [17, 18] and Bayesian probabilistic rules [19]. In the Dempster-Shafer theory, the missing links are identified from a set of possible missing links by accumulating both geometric and topological evidences using Dumpster's rule of combination. This method suffered from computational inefficiency, because of its incapability to recover more than one missing link at a time. Ji and Marefat [19] expanded the original algorithm [17] for missing arcs restoration by means of Bayesian nets. The parts of evidence, which consisted of topological and geometric relationships at different abstraction levels, were combined to form a set of correct virtual links. These links were to be emerged to the cavity graph representing a depression of the object so that the resulting supergraph can be partitioned to obtain the features of the object. The hierarchical belief network was constructed on the basis of the hypotheses for the potential virtual links, which impacted the belief network through the amount of support for different hypotheses. This method was able to recover simultaneously all missing links and recognize more complex interacting features with improved recognition accuracy. But anyway these approaches could not completely solve the problem.

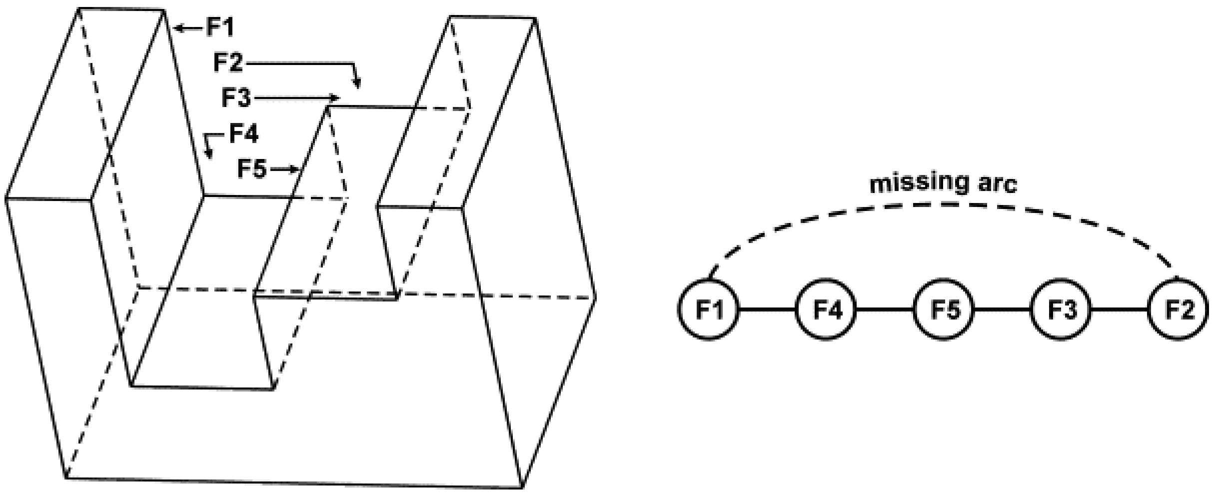


Fig. 10. Missing arc in intersecting features [17]

Not every graph-based method was built on face-edge graph. Qamhiyah [20] proposed a feature extraction technique which was based on loop-adjacency hypergraph and was focused to obtaining generalized properties of the classes of features with planar faces only. This limited area of application is the major drawback of this technique.

The system presented by Huang and Yip-Hoi [21] is focused on so-called “high-level” features such as “stepped”, “compound” and “array” features (Fig. 11). A feature relationship graph is used to organize primitive features in user-specific high-level feature patterns. In such way, at least these categories of interacting features may be recognized, but it requires extensive user intervention, reducing the level of automatization of this feature recognition system.

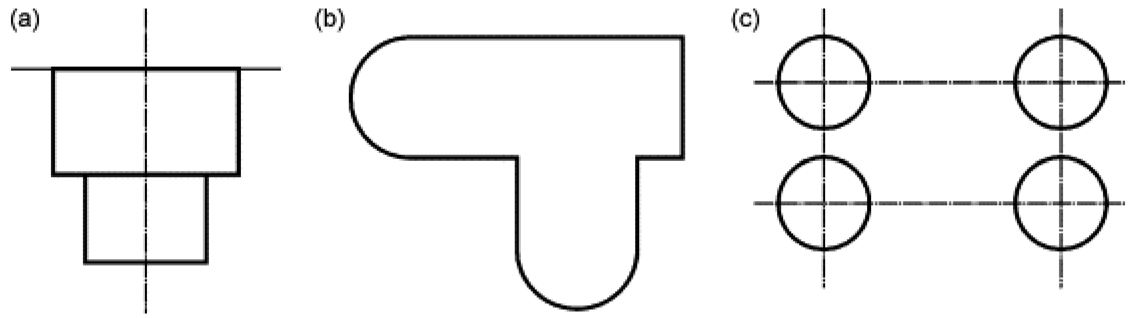


Fig. 11. High-level features: (a) stepped features (profile combination); (b) compound features (path combination); (c) array features (instance combination)

Di Stefano [22] proposed a system which is based on face adjacency multi-graph, precisely attributed to capture all properties that are important for the part's manufacturability. For example, the system can manage properties such as mutual relations between faces that are not necessarily adjacent: parallelism, coaxiallity and perpendicularity. The attributes of the nodes and of the arcs of the graph are arranged in order to obtain a unique representation, so-called “intermediate model” with a wide range of data needed for engineering oriented semantic recognition. The semantic construction is based on the concept of "semanteme”: the minimal element of meaning that the system can manage and recognize in a geometric model which is related to some kind of machining context. This system may be capable of capturing a large quantity of procedural knowledge, directly from geometric model, but it depends of human intervention for its validation. Integrating the geometric modeling system with the feature-recognition system, as proposed by the authors themselves, may be an improvement of the proposed semantic recognition method, especially in the domain of interacting features, but it will bring this system even further from AFR to design-by-feature methods.

All methods of this graph-based group require extensive pre-processing in order to construct representation for each part and each primitive and in most examples of their application only polyhedral parts are treated. Even when they are capable of successful recognition, there is no guarantee that the recognized feature will prove applicable in the sense of manufacturability, i.e. that they will be suitable for use in further other modules of a CAPP system. However, the main problem which graph-based methods could not effectively solve is the problem of interacting features. This drawback can be partially lowered by using various techniques of geometric reasoning. The other way is to enrich the feature library with as many interacting features as it is possible, treating them as singular features. This approach requires a lot of computational time for searching and pattern matching and does not give an universal and complete solution of the problem, because on and on a feature will occur which is not included in an existing feature library. From all these reasons graph-based approaches caused larger investigation of alternative methods, such as volumetric and hint-based, to deal with interacting features.

2.4 Convex hull volumetric decomposition

Convex hull decomposition is the approach based on decomposing the input model into a set of intermediate volumes and manipulating the volumes in order to produce features. Kyprianou gave the original idea for convex hull approach [10]. A polyhedron convex hull is determined, circumscribed around a part. The difference in volume between the part and its convex hull is defined as an alternating sum of volumes (ASV). Kim in [23] provided the method for convergence, initiating remedial partitioning procedure - ASV with partitioning (ASVP) and, since then, he and his associates (Wang, Waco and others) worked on this task to provide an effective algorithm for this method implementation.

At the beginning this approach was successfully applied for polyhedral parts because of the complexity of the convex hull computation for curved objects. Other problems that emerged during the development of the method raised: how to convert the set of alternating volumes into meaningful constituents of shape of the part and, further, into machining volumes. So Kim's approach, after a few modifications gave a solution to these problems, by the means of the following steps [23]:

Step 1. Extraction of cylindrical hole features - an algorithm has been provided for extraction of holes (cylindrical surface closed at least at one position along its axis), illustrated in Fig. 12a, which is not part of the original work.

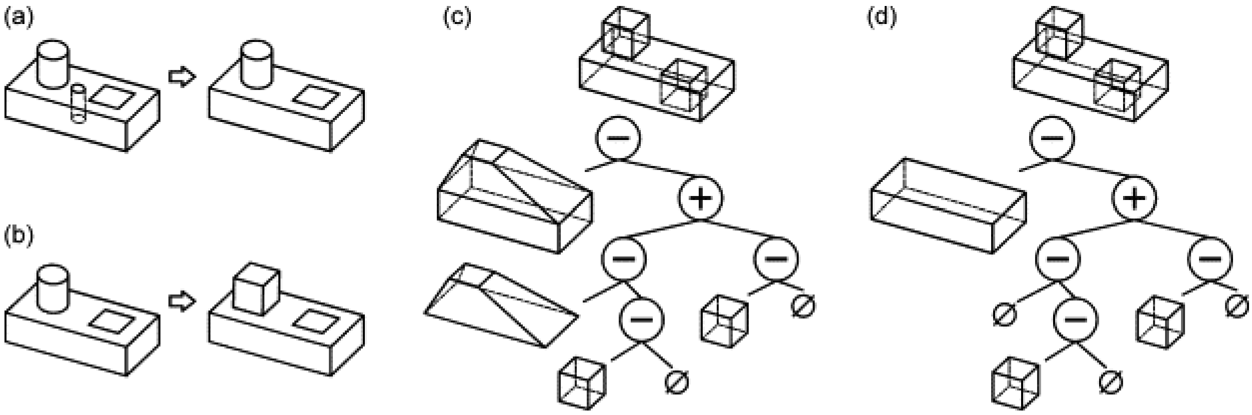


Fig. 12 Kim's approach: (a) extraction of cylindrical holes; (b) polyhedral abstraction; (c) ASVP decomposition; (d) form feature decomposition

Step 2. Polyhedral abstraction of blending and cylindrical faces – another algorithm has been provided for identification of blending and cylindrical surfaces (other than holes and with a constant radius only), Fig. 12b.

Step 3. ASVP decomposition, Fig. 12c. The convex hull of a polyhedron is the smallest convex point set containing that polyhedron. The convex hull difference is the regularized set difference between convex hull and the polyhedron. The decomposition is recursively applied to the convex hull difference until it becomes convex, when the decomposition terminates.

Step 4. Form feature decomposition, Fig. 12d. This step aims to give meaning to decomposed component combining them into high-level constituents of the shape of the part.

Step 5. Conversion to machining features - a “positive-to-negative” conversion is applied to convert positive components of decomposed volume to negative features which represent removal volumes providing information about machining surfaces, tooling and sequencing. Geometric reasoning in this step is different for specific manufacturing processes. In case of getting an unsatisfying result the alternative machining volumes can be obtained through the process of aggregation of negative features. Combining the positive components from previous step with the stock component a machining feature extraction for cast-then-machined parts can be obtained;

Step 6. Re-attachment of cylindrical and blending information—blend is restored to a feature if it exists between faces which are all part of that feature; other blends between multiple features are stored as implicit relations between corresponding faces of those features.

The test of current approach, performed on a group of various parts in several manufacturing domains (such as mill-turn process or machining of casting parts), produced good results. However, this system can deal only with polyhedral features and cylindrical features which interact with them in principal directions, with constant-radius variations. There are some other drawbacks which may produce unsatisfying and not stable results. The ASVP decomposition is completely separated from the feature recognition, and is not guided by the goal of recognizing specific types of features. Combining methods described in step 4 of the method are often incapable to produce recognizable features. Negative components (machining forms) obtained in step 5 should always be convex but the algorithm often terminates with a set of unmeaningfully shaped negative features. Proposed conditions for aggregating primitive components do not provide universal solution of the problem.

One of the examples of ASVP implementation has been shown in [24]. Proposed system has some limitations concerning forms production that can be managed only in 2.5- and 3-axis machining centers. Extraction of geometric information (primitives) is performed using external approach - neutral STEP or IGES data file is exported from CAD model of the part, with B-rep. Faces in ASVP derive different attributes whether they are part of the stock (SS), finished part (MS) or they emerge in some intermediate stage of manufacturing (IS). A general set of forms, issued as a unique combination of SS, MS and IS is then defined as a generation attribute of the feature. Independent forms are directly recognized through pattern matching, while interacting forms have first been parsed along the concave edge loops. Process illustration is given in Fig. 13.

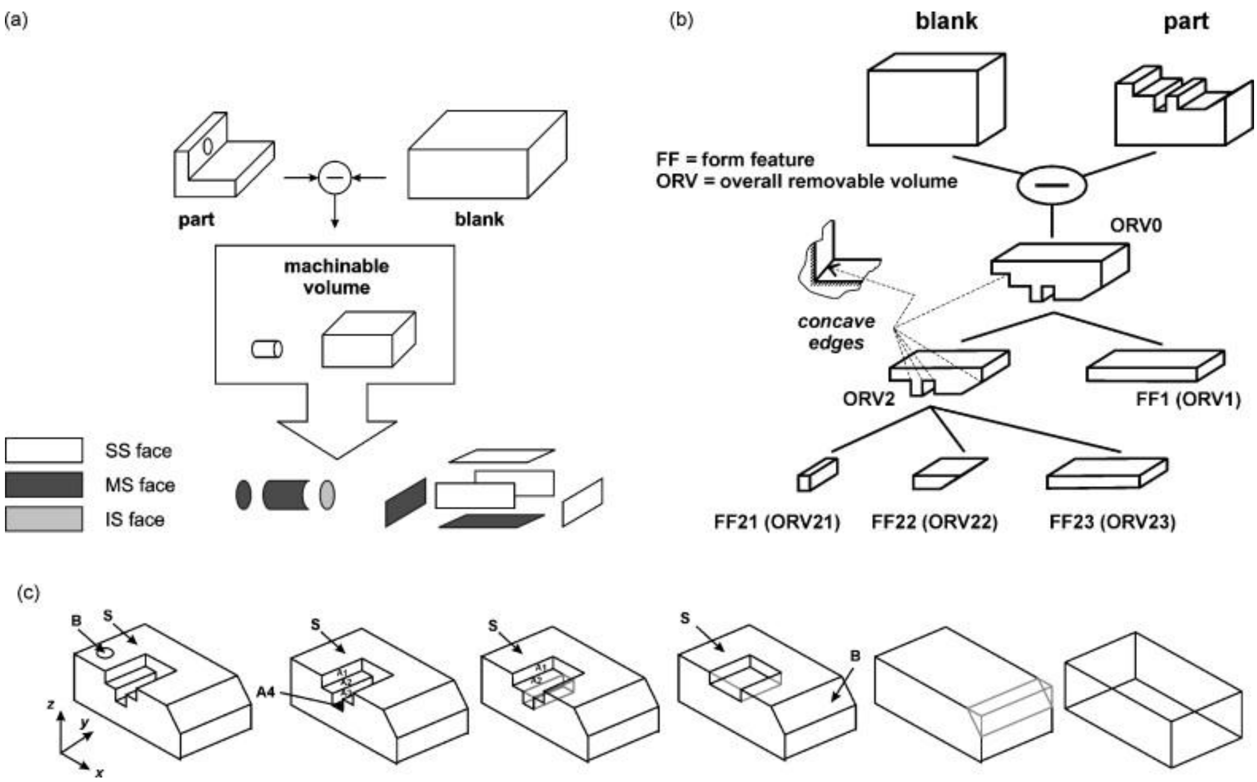


Fig. 13. AFR systems (a) Miao; (b) Dong and Vijayan; (c) Nagaraj and Gurumoorthy [24]

A similar approach has been developed by Dong and Vijayan [25]. The system extracts features from a CAD model using an original technique called "blank surface - concave edge". There is the overall removable volume (the total material that has to be removed from the blank to produce a finished part) is represented as a set of elementary manufacturing features (parts of volume that are removed in a single tool path). There are numerous alternatives for segmenting the overall removable volume into machine volumes - the optimal one is selected using the mathematical optimization algorithms. The extraction of geometric features and formation of the part representation for AFR is performed through graphical comparison of the part and its blank. The pattern matching process is based on if-then rules. The illustration of the current system is given in Fig. 13b.

One more similar approach was developed in 1990 by Wang called "backward-growing approach" [26]. It had a purpose of more effective treatment of intersecting form features, which can be recognized as particular complex forms or sets of trivial forms. This approach was implemented in the system developed by Nagaraj and Gurumoorthy [26], also based on predefined manufacturing features. The cavity volumes, regarding the most distant outer surface of the part, are defined and, in an iterant process, filled with predefined manufacturing primitives (cuboid, wedge, cylinder, etc.). These primitives are then used for CSG tree formation, in whose structure they can further be reorganized to better suit the selected blank's dimensions. This approach is also specific in that it envisages preformed blanks. The illustration of such system is given in Fig. 13c.

2.5 Cell-based volumetric decomposition

Every example of cell-based decomposition approach, the basic algorithm is the same and consists of three steps: (1) the overall removable volume is identified as a difference set between the blank and the finished part; (2) this volume is then decomposed to unit volumes by using the extended boundary faces as cutting planes (cell decomposition); (3) all unit volumes that have common faces or coplanar faces are merged to get maximum cells that can be removed in a single tool path (cell composition).

The basic problem that characterizes this method: even in the case of a simple part a number of cells created in the step (2) may be very huge, leading to large number of possible feature interpretations in step (3). This problem, addressed as "the global effect of local geometry", is generated in the cell decomposition step which extends surfaces or half-spaces associated with the faces of delta volume through the whole part, reaching the areas where machining features would not extend in a reasonable machining sequence. As a consequence, a large number of unnecessary cells are created and the most attention in cell-based methods is paid to dealing with them because they generate multiple interpretations of possible machining features.

Although it was not originally the idea of Sakurai, he was the major contributor to this approach. In earlier version of his system [27], he proposed all multiple interpretations to be generated and then recognized through graph-pattern matching. This system was designed for parts with planar faces and only a limited number of cases of convex curved faces. A large number of possible combinations of cells (up to n!) led to an enormous time complexity and, also, not all the interpretations were reasonable from a machining point of view that depicted in Fig. 14.

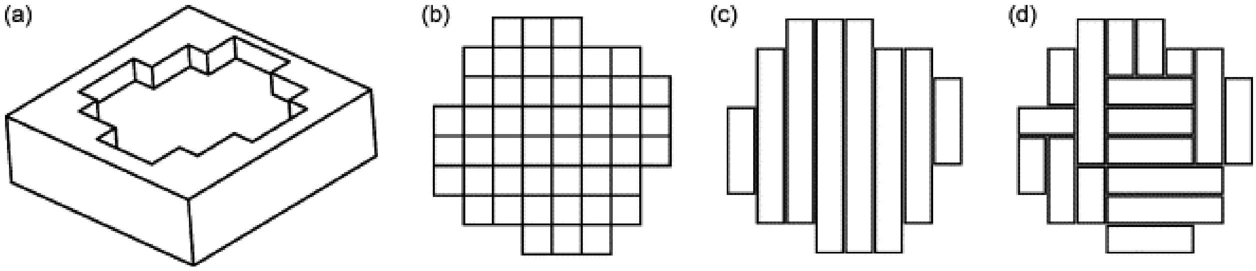


Fig. 14. Multiple interpretations of features in cell decomposition/composition process: (a) part; (b) cell decomposition; (c) a reasonable composition result; (d) an awkward composition result.

In later work he gave detailed and very efficient algorithm for decomposition process and also proposed several heuristic rules to solve the problem of unreasonable compositions, based on convex intermediate volumes and empirical concept of machining the volumes of “simple shapes with large cutters”.

Sakurai and Dave [28] proposed another algorithm which, unlike previous, allows concave intermediate and final volumes, extending the application of cellular method to objects with all types of curved faces and reducing, but not totally eliminating the problem of “complex and awkward” features. In the most recent research, Woo and Sakurai present the development of an algorithm for scalability of complex parts in order to reduce computational exhaustion and improve applicability of cell-based approach. The delta-volume is recursively bisected into smaller volumes until each one has less than 16 faces. Bisecting planes are chosen to divide volumes in two with similar number of faces. Each of the volumes is then recomposed into maximal volumes which do not have concave edges and are not contained one in another. Then, a search for minimal non-redundant set of maximal volumes is performed. Each volume is examined whether it can be produced in a single operation on a 3-axis machining center. Then if so, it is recognized as a maximal feature, if not, it is further decomposed. This method is very useful for the most real-world parts (for 2,5- and 3-axis machining centers are the most numerous in contemporary manufacturing industry), but has no application beyond this limit.

Another attempt to further improve this method was made by Woo [29]. He concentrated on the problem of “the global effect of local geometry” developing an algorithm for “localized face extension” which enables faces to be extended only over the concave edges, reducing computational complexity in more than 10 times. Further simplification is performed in cell composition stage, through cell collection using “seed cells” (which always exist in maximal volumes), significantly reducing the number of possible interpretations. The drawbacks of this approach are inherited from the original method and it also suffers from limitation to objects with features that possess concave edges.

One more example of cell-based decomposition method application is a system developed by Tseng and Joshi [30], which can be explained using example of AFR for so-called mill-turn parts. Input geometrical information (B-rep) is a postprocessed text file in ACIS solid-modeler application protocol interface. First two steps in this method are similar to Sakurai's. In the composition stage the feature volumes are generated by sweeping a boundary face in direction of an adjacent boundary face. The direction depends on predefined type of machining operation: rotational (turning) or prismatic (milling). The methodology of the approach has one additional step, differing for prismatic and rotational structures. For prismatic structures an AAG representation of the cells is made and pattern matching performed, whereas for rotational structures syntactic pattern recognition is performed that is shown in Fig. 15. This approach extends application of cell-based methods to more than 2,5-axis machined parts, but is not applicable to eccentric, asymmetric and complex-curved profiles. Another major drawback of this method is redundancy in pattern recognition-some features may be recognized both as cylindrical and prismatic, depending of sweeping direction.

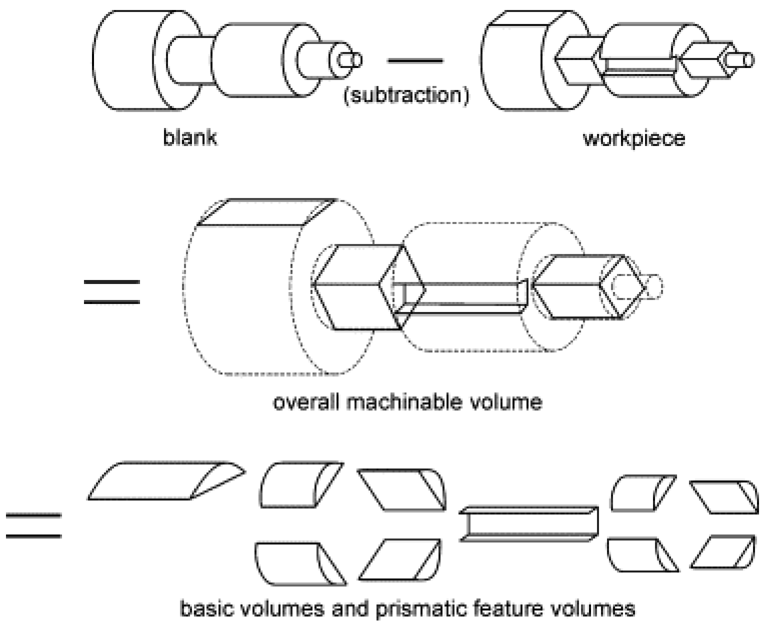


Fig. 15. Cell decomposition method illustration

Recent investigations, with broader field of application than AFR, have opened new perspectives for cell-based decomposition. A multiple-view feature modeling system for integral product development [31], with a practical solution (SPIFF modeling system) has been a result of a research team of Delft University of Technology, leaded by Bronsvoort. In its core is a semantic cellular model (non-manifold geometry) that integrates the contributions from all features in a feature model. It represents geometry as a set of volumetric cells of arbitrary shape that do not geometrically overlap, and lie completely inside or completely outside the shape of a feature. Each cell contains information on every feature that overlap with it, and each cell face contains information on the feature faces that overlap with it. A cell also contains information whether it represents material or void and a cell face whether it has material on both sides, or just one side (feature boundary cell). All this information is stored in “owner list” for each cell and cell face; it indicates to which features and feature faces it belongs in each view. The nature of a cell in a view is determined by the features in that view that overlap with the cell and the dependencies between them. So the semantics of feature is well defined and maintained during all modeling operations. The term “view” is used to denote different models of product in its development phases.

Cellular representation is designed to be an alternative to classic B-rep models (sometimes even more valuable) because it is history-independent (independent of order of feature creation) and it may store some additional information on features, including some faces which are not on a boundary of the object. Such defined model, using geometric reasoning based on constraint solving, can be converted in specific feature models to be used in different product development phases: conceptual, assembly, part detail and manufacturing planning, and, within the latter, feature recognition. The feature recognition algorithm consists of four phases [31]: shape recognition (the candidates for the shape of a new instance of a feature class are recognized in the cellular model using the shape type and the attach constraints specified in a feature class), parameter determination (the dimensions of the candidate shape are determined from the location of their cell faces, then, an instance of the candidate shape is created and remaining constraints of the feature class are checked. The largest candidate shape satisfying all constraints is the feature shape; if no such shape exists, another feature class is examined), extraction (the feature shape is added to the feature model of the manufacturing view, reducing the inconsistency between the feature models of the part detail and manufacturing), and organization (selection of faces relative to which the feature is positioned, based on specific information from the feature class of the new feature). This approach, thanks to its consistency through different stages of product development and singularity of feature definition, may solve the problems of previous examples of cell-based AFR method application (computational complexity and multiple interpretations), but, due to parametric B-rep based character of the most of contemporary modelers, it will wait for greater popularity until commercial cellular modelers emerge.

2.6 Hint-based approach

Initial purpose of current approach was to solve the problem of arbitrary interactions of form features and presents a combination of logic approach and delta-volume approach or graph-based approach. Topological, geometrical and heuristic information about the envisaged part are used as the hints of presence of a certain form feature. The largest volumetric feature possible from a hint is then constructed and tested for validity. The method was initiated by Vandenbrande and Requicha in a system called Object Oriented Feature Finder (OOFF), which was designed to deal with intersecting features that, due to immense variety of types of their appearance, had made approaches based on searching the exact patterns of faces, edges and vertices unsuitable for the most of practical problems. The authors defined the “presence rule” which stated that a machining operation which had produced some feature should have left a trace in the part boundary even when that feature intersected with another. This provoked some other relevant researchers in the field to have given an alternative name to this approach: trace-based. For any feature, a minimal indispensable portion of a feature's boundary may be defined, which, when found in the nominal part geometry, might provide a hint for the potential existence of that feature.

Later this system has been improved with the ability to reason about hints generated from various sources, such as direct user input, tolerances and attributes, and design features. The system has been renamed to IF2 (Integrated Incremental Feature Finder) to reflect its ability to combine design-by-features and automatic feature recognition approach into one. Opposite to OOFF, which produced all possible interpretations of feature intersections, in a time-consuming and computationally expensive process, IF2 uses heuristics to generate an interpretation and considers alternatives only on user's demand. The latest version of IF2 system [32] has included the principle to recognize only manufacturing features, consulting the tool database, in order to facilitate sequencing process in an overall CAPP system.

Many other researchers have contributed to enhance the method with completeness of class of features recognized, efficiency of algorithms, use of additional information as hints, and independence from a modeler applied for the part's design. One more example of such a system is given in [33]. The system focuses on form feature recognition in orthographic and isometric projections of a part, without the use of hidden lines, which cannot be obtained as an input from automated visual inspection systems that this AFR system is aimed for. The input into the system is a graph representation of engineering drawing projections. This representation is first to pass the profile searching stage, which identifies 2D contours in the orthographic views and then it goes through the feature completion stage which establishes the cavity volumes associated with these 2D contours. If the features cannot be established using orthographic views only, then the isometric views are examined. The method uses the so-called “divide and interpret” strategy: instead of exhaustive analysis of the whole drawing, the system divides the drawing into several parts, using information obtained from isometric view analysis as hints of presence of a certain form feature.

McCormack and Ibrahim [34] implemented MAAM concept in their researches, adopting external approach for geometric information extraction (neutral data formats - IGES and STEP, or *.dxf) and using hints instead of simple pattern matching to extract the multiple interpretations of the features detected in the part.

Clark and Corney introduced a new algorithm to extract hints from solid models, based on a concept of “light rays” originating from the eyes of a human observer. From this viewpoint, rays are fired and the intersections of rays with different faces of the solid model determine various types of features. This technique, also known as “a viewer-centered approach” can successfully recognize different types of orthogonal features, using the same principle applied in computerized tomographic scanners in medicine. The algorithm suffers from several shortcomings, as pointed by the authors themselves: a large number of hints containing many duplicates, small features may be missed by the hint generation process if an inadequate set of viewpoints is chosen, and the process of finding the bounding faces of the feature may give awkward result in cases of features like depression in torus.

2.7 Hybrid approach

Gao and Shah [35] proposed an approach that combines the conventional graph-based recognition with hint-based recognition. The authors have accepted the concept of virtual links (the edges that, as a result of feature interaction, are not contained in the B-rep of a part) and used them to attribute the extended attributed adjacency graph (EAAG). Their algorithm uses a library of predefined features (steps, blind steps, slots, chamfers, etc.) and a library for compound features, general, through and open pockets and other features that can be generalized through a set of heuristic rules. Each feature is defined in terms of its (EAAG) and other data (feature parameters, access directions, obstacle faces, etc.). The sub-graph components, called minimal condition sub-graphs (MCSG), are generated from the part EAAG, and used as feature hints. After being further processed using extensive geometric reasoning, MCSGs are completed to a recognizable form by restoring their missed links. This system, when applied to parts with planar and cylindrical faces, has proved capable to recognize both non-intersecting (isolated) and interacting features and provide alternative interpretations for each set of interacting features. Nonetheless, its limitation to this class of features only presents a significant shortcoming.

The group of researchers (Corney, Clark, Little, Tuttle) from Heriot-Watt University, Scotland, UK, developed a feature recognition system, known as FeatureFinder [36], with an algorithm based on a graph search. At first, the algorithm had been able to interpret the geometry of polyhedral single-sided components but, it was later extended to handle multi-sided components requiring more than one machining direction, depressions and protrusions bounded by cylindrical faces (as well as planar faces) and “open” features as through slots. The latest version of FeatureFinder has been capable of identifying a variety of features on a wide range of machined components. The system has been designed to be used within a solids machining package and identifies features from a specified tool approach direction.

In four distinct steps, the algorithm produces a set of manufacturing feature volumes, each of which represents the material to be removed by a manufacturing operation. The first step represents the selection of the tool approach (aspect) direction by a process plan engineer. Only one tool approach direction is considered at a time. Secondly, a closed chain of paths which travel across each vertical face of the component is generated (“vertical” face is the one that lies in a plane parallel to approach direction). Then, for each machining direction, the paths are linked to form closed cycles representing the 2D profile of 2,5D feature volumes. In the last step, a user interaction is needed again to select which cycle is to be used to create the feature volume. Once a valid cycle has been generated, the 2D profile is swept to the height of the cycle, manually defined by a user. Only planar entities can be considered for sweeping in this technique for volume construction and even the presence of drafts may prevent its successful application. For multi-sided components, the user has to observe which paths are approachable (“visible”) in the given direction.

Human intervention, demanded to provide the tool approach direction, cycle selection and visibility information, should be considered as a major drawback, because it reduces the level of automation of feature recognition process; but it also may be regarded as an advantage, for the volumes identified in such way share a common set-up which would be useful in subsequent stages of process planning, such as sequencing the manufacturing operations. Although primarily graph-based, this approach is in this review classified as a hybrid one, because of its multi-step reasoning character and volume construction driven by tools accessibility, which is the attribute of volumetric approaches, and human intervention, which is widely promoted in hint-based approaches.

X.Ye in [37] presented the AFR system based on face-edge EAAG, aimed to recognize isolated and interacting undercut features from moulded parts planar, quadric and free-form surfaces. It takes face properties and parting lines as hints for recognition of undercut sets. To deal with the face properties of free-form surfaces a convex-hull algorithm has been developed. This system is supported with an extensive set of heuristic rules for hint generation and has proved to be successful in this limited area of application.

An attempt to develop more general system has recently been presented in [38]. This method uses an AAG which is decomposed to limit and organize the search space for feature hints. Hints, in sub-graph forms, are extracted from the decomposed graph components. They indicate whether the feature is 2.5D, floorless or 3D. To reduce the product model complexity while extracting features, a method to remove fillets existing in the boundary of a 2.5D feature is also proposed. The hints are extracted in a graph form, but the feature is completed geometrically, using three geometric completion algorithms (base and profile completion for 2.5D features and 3D-volume generation algorithm). The base completion and profile completion algorithms generate maximal volumes for features whereas the 3D volume generation algorithm extracts 3D portions of the part. This hybrid system, beside graph and hint based combination, adds the volumetric component, completing feature's geometry instead of graph components in the process of restoring of missing links. The authors have described examples of successful application on several test-parts from NIST design repository, but as any other system based on hints, it also requires user intervention and seems to be verbose.

There are several examples of hybrid AFR systems which combine characteristics of graph-based approach and convex-hull volumetric decomposition. One of them is the system described in [39], which uses a modified attributed adjacency graph (AAG) to facilitate the representation and recognition of isolated or interacting depression features. The modification of AAG is performed by adding the “reference face” (determined on the basis of convex hull concept) into the AAG, providing more clues in the process of feature detection and recognition. Two general feature types, namely depression and protrusion features, are identified by the reference face. The basic features such as slots, pockets and bosses are represented by the modified AAG (called RAAG by the authors) and the other features that remain unrecognized are regarded as interacting features. The recognition of features, also based on the concept of virtual links, is

enabled by reconstructing the necessary boundary faces that might have been lost due to feature interaction. The extraction of features is simplified by adding the cavity volumes of the recognized features back to the original volume of the object. This system has shown a significant capability of dealing with interacting features, but only with those predefined in the feature library. That's why it needs redesign of its pattern recognition function (the third task of AFR) and transfer towards the use of AI techniques.
The other example of a hybrid between graph-based and convex-hull approach is a system developed by Sundararajan and Wright [40]. It introduces so-called “open edge” concept to promote the information about feature adjacency. The system is able to treat prismatic parts and even a limited class of free-form features, but only those machinable from six basic directions along coordinate axes. Free-form features are defined similar to the 2.5D features as comprising a planar contour, but substituting a bottom free-form surface for the depth. Covering faces, defined as projection of the free-form surface on the faces of the bounding box of the surface, are used as equivalent planar faces for performing the recursive descent. The relationship between the free-form feature and other neighboring features is determined through common open-edge identification. The drawbacks of the system, beside the restriction regarding machining directions, include the problem of recognition of fillet as a free-form feature.
Subrahmanyam in his more recent work also introduces hints into the cell-based volume decomposition method. Improved, heuristic-based volume decomposition method, by removing possible isolated machining features and slicing edges in an early stage of recognition process, reduces the search space and the complexity of the combinatorial problem. This system is strongly oriented to manufacturability of recognized features and, being able to generate alternative solutions, proves to be flexible and adaptable. However, the algorithm is not able to recognize chamfers and fillets as high-level features. The other drawback is that this system can work with parts with single set-ups only.
As a summarization: if a system with rule-based pattern recognition has one or more future, then it belongs to the hybrid approach resolution. They combine the advantages of constituting conventional methods and there are many examples of their successful applications, but only for limited classes of features they are primarily designed for. The lack of successful hybrid algorithms generalization for broader range of feature classes presents the major drawback of this set of approaches for automated feature recognition.

CHAPTER 3. STEP-BASED FEATURE EXTRACTION METHODS

As mentioned in the previous section the effort to develop an international standard for product data exchange has led to the development of the STEP standard. Gilman [41] has demonstrated the use of Product Data Exchange using STEP (PDES) in a feature-based designing environment. The work focuses on the implementation of the PDES form feature information model (FFIM) as a conceptual schema for an object-oriented database. Shah and Mathew in [42] developed a translator to and from FFIM.

One of the other work in this field by Ssemakula and Satsangi [43] describes interfaces between process planning systems and CAD systems using the PDES.

In the rest part of this chapter there are different methods presented which are of some interest from the point of STEP-based feature extraction optimization.

3.1 Automatic Feature Recognition framework

Van der Velden developed a framework for automatically extracting engineering features from neutral STEP models for use in downstream processes including, but not limited to, analysis (CAE systems) and manufacturing process planning (CAM systems) [44]. The Automatic Feature Recognition framework (AFR) introduced and implemented within a prototype system. This prototype system was based on a particular application: identification of analysis features in integrally stiffened frames. The system is expected to automate the extraction of different features basically focused on panels and detail ribs to facilitate automation in downstream stress analysis processes.

The feature recognition technique developed for this system differs from traditional graph techniques in several key areas [44]:

- Analysis of relationships between faces that are not immediately adjacent provides more complete view of the model allowing the interaction of detailed features to be determined with a greater level of accuracy.
- Identification and suppression of detailed features such as fillets, rounds, holes and thickness simplifies the model. This allows generic rules to be specified that can identify features regardless of the complexity of surrounding detail.
- The system employs a computational geometry engine to determine the relationships between faces and edges with multipart underlying surface and curve geometry. This allows complex 3D design models to be evaluated.
- High computational requirements associated with pattern matching in graph structures are reduced with the integration of an inference engine that efficiently searches model data for entities that satisfy feature rules.

The identification, capturing, organization and implementation of feature recognition rules within the AFR system is an important design consideration that affects system performance and accuracy of feature searches. Below a five steps process to retrieve these rules presented [44]:

Step 1. Define Feature Taxonomy. Feature taxonomy refers to the fundamental characteristics of a given feature type, i.e. how is a region of a geometric model recognizable as an instance of a feature type? The definition should be generic, including only the minimum characteristics required to identify an instance of the feature type, i.e. ignore variations caused by sub features. The main task of current framework is to extract panels from B-Rep model, so the top level description of a panel could be “thin planar section of material bounded on all sides by ribs”. This definition is true for all panels regardless of the level of complexity.

Step 2. Identify Feature Attributes. The second step identifies the minimum set of information that is required to represent the feature that can be identified in a B-Rep model. Because the

AFR system applies inference procedures to the set of processed model data, consequently only this data can be used to define feature attributes. This means that rules can only use topological and geometric entities contained within the model itself, and not extrapolation/interpolation of these entities (e.g. mid-surfaces, intersection curves/points, etc.) unless previously calculated. For this reason, current step identifies model elements (surfaces, curves, etc.) that provide the necessary information to extract the desired feature. As a result of this step the following attributes specifying the minimum panel information are retrieved: panel face, opposing panel face, collection of rib faces and collection of opposing rib faces.

Step 3. Parametric Design of Feature Attributes. The parametric design phase identifies parameters of feature attributes that uniquely separate them from other entities in the model data. When identified, the following attribute parameters take place: adjacent faces (either faces immediately adjacent, or faces separated by one or more intermediate faces), face surface type, edge attribute (concave/convex/tangent), angle between faces, distance between entities, face area and etc. After the unique parameters of feature attributes are identified, a rule for extracting the feature attribute is extracted. Continuing with the example introduced earlier, instances of Panel Face attributes can be identified from most others using the following criteria: “a planar face surrounded on all sides of the outer boundary by faces that are concave and normal to the face surface”, i.e.: 1) surfaceType = plane; 2) edgeAttribute = concave for all adjacentFaces on main wire; 3) faceAngle = 90 degrees (± tolerance) for all AdjacentFaces on main wire; 4) faceArea > minPanelArea.

Step 4. Form Logical Expressions to Extract Feature Attributes. On this step parameterized rules are translated to the form that can be interpreted by the inference engine. This is accomplished by forming a set of logical expressions from the previous step. For instance:

```
get list of all faces with surfaceType = plane (List1);
for each face in List1
    if faceArea > minPanelFaceArea then
    get list of all adjacent faces on outer wire (List2);
    get list of all adjacent faces on outer wire that are
    normal and concave (List3);
        if lengths of List2 and List3 are NOT equal then
        remove current face from List1;
        end if
    end if
end for
return List1;
```

Step 5. Execute Rules within Rule Engine Framework. Outputs from the inference engine will be a list of entities that satisfy criteria contained with in the rule for a particular feature attribute. This will usually be a list of unique identifiers of model entities.

Presented five steps process for structuring feature recognition rules is also depicted in Fig. 16.

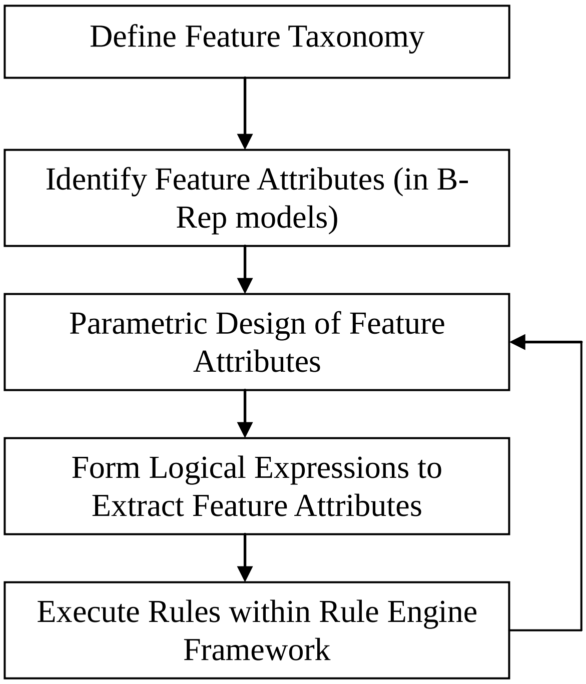


Fig. 16. Rule design process [44]

Referring to the test results in [44], the AFR showed good results in recognizing and extracting of engineering features from the geometrical models. But the limitation of current framework to the area of the plane panels only, encourages to further investigations.

3.2 Attributed graph matching

El-Mehalawi and Miller [45] used the attributed graph matching approach to compare CAD models of engineering parts in STEP format. The models are converted from the STEP format to attributed graphs whose nodes contain geometric attributes that represent the surfaces of the STEP model. The graph matching process and the experimental results are also described there.

3.2.1. Attributed graphs overview

Attributed graphs are appropriate representation scheme for building a data structure that captures the geometrical and topological similarity between mechanical components. It facilitates the retrieval and comparison of parts. The CAD model itself is not a suitable representation scheme for such a data structure. Yes, it can provide some information for the retrieval process such as the number of surfaces and the number of planar surfaces. However, it is very difficult to compare two CAD models in order to assess the similarity between them and to pick the closest part among the retrieved parts. That is because the process needs a huge amount of complex reasoning to compare two B-Rep structures. Attributed graphs represent part topology, geometry and size in a compact data structure. Although graph comparison is not an easy task, it is much easier than comparing B-Rep structure.

Benefits of Attributed graphs usage [45]:

- attributed graph is a compactified data structure that saves required storage size;
- accelerating data search and extraction, it facilitates the comparison of objects;
- graph comparison is known to have an NP-complete problem that is effectively solved within attributed graph approach;

- attributed graphs facilitate representation, indexing, retrieval, and object comparison for the case when objects are stored within database;
- this representation also provides a means for data abstraction that might be needed for tasks.

3.2.2 Attributed graphs representation

Within attributed graphs approach, a part is represented using its surfaces and the relations between them. Part surfaces are given explicitly in the STEP file. A part is represented by an attributed graph with the nodes representing the surfaces and the links representing the edges that connect two surfaces. Surface attributes such as type of surface and direction of the normal are associated with nodes. Edge attributes such as type of the edge, the two connected nodes, length of the edge, and the relative direction between the two nodes are associated with links. So the graph itself represents the topology while the attributes represent the geometry of the component. Authors claim that this representation is adopted from the literature of computer vision of 3D objects and from the literature of CAPP with modifications. Both of these areas deal with part representation that starts with close information provided by STEP.

3.2.3 Construction of attributed graphs

Below presented an algorithm that describes the mapping from the STEP space to the attributed graph space:

1. Find the line containing CLOSED_SHELL(face1,face2,…,facek) and/or OPEN_SHELL (facek+1, facek+2, …, facen).

2. For every face from 1 to n, do:

  ◦Find the corresponding ADVANCED_FACE((bound1,…,boundm),surface)

  ◦For every bound from 1 to m, do:

    -Find the corresponding FACE_OUTER_BOUND(edge_loop) and/or FACE_BOUND (edge_loop)
    -For edge_loop, find EDGE_LOOP(edge1,…,edgeu)
    -For each edgei, find ORIENTED_EDGE(E)
    -Assign E as an edge of the current face.

  ◦For surface, find the corresponding one of the following and assign it as SURFACE_TYPE:

    -PLANE(direction)
    -CYLINDRICAL_SURFACE(direction, radius)
    -CONICAL_SURFACE(direction, radius, semi_angle)
    -SPHERICAL_SURFACE(direction, radius)
    -TOROIDAL_SURFACE(direction, major_radius, minor_radius)
    -BOUNDED_SURFACE( )

  ◦For direction, find the corresponding AXIS2_PLACEMENT_3D(origin, z, x)

  ◦Assign the DIRECTION corresponding to z as the DIRECTION_Z of the face.

  ◦Assign the DIRECTION corresponding to x as the DIRECTION_X of the face.

3. For every face from 1 to n, do:

  ◦For every edge that belongs to that face, do:

    -Initiate an instance for that edge.

-Assign the current face as NODE_1 of that edge.
-Find the other face that contains the edge and assign it as NODE_2.
-Delete the edge from both faces.

4. For each initiated edge, do:

◦ Find the corresponding one of the following and assign it as the CURVE_TYPE of the edge:
-LINE(point, vector)
-CIRCLE (direction, radius)
-B_SPLINE_CURVE_WITH_KNOTS( )

◦ If the curve is LINE, assign it to CURVE_TYPE and
-For vector, find the corresponding VECTOR(direction, length)
-Assign length to be the LENGTH of the edge.

◦ If the curve is CIRCLE, assign it to CURVE_TYPE and assign radius to LENGTH of the edge.

◦ If the curve is B_SPLINE_CURVE_WITH_KNOTS, assign it to CURVE_TYPE of the edge.

◦ x1←DIRECTION_X of NODE_1

◦ x2←DIRECTION_X of NODE_2

◦ Assign arcos (x1·x2) to the RELATIVE_DIR_X of the edge.

◦ z1←DIRECTION_X of NODE_1

◦ z2←DIRECTION_X of NODE_2

◦ Assign arccos (z1·z2) to the RELATIVE_DIR_Z of the edge.

3.2.4 Conclusion

Attributed graph approach takes an inexact graph matching approach that avoids the combinatorial problems associated with exact matching. Similarity measures are generated using an inexact graph matching algorithm based on integer programming. However tests in [45] were accomplished with the usage of models of small sizes and moderate complexity. Two parts with surfaces of the order 200 compared successfully, although the times taken for comparisons have not been presented. While topological graph matching based on the original detailed representation is a good approach for similarity determination, it becomes unworkable for large and complicated models.

3.3 Multiple-level feature taxonomy method

Multiple-level feature taxonomy method proposed by Fu et al. [46] that sufficiently retrieves feature set from the viewpoint of part design and manufacturing. To facilitate feature identification and extraction, a multiple-level feature taxonomy and hierarchy are proposed based on the characteristics of part geometry and topology.

Within current method relationships between features and their geometric entities are established. Also a bunch of algorithms for the identification of design and machining features are proposed. Besides, the ways to recognize the intersecting features or compound features based on the featureless chunks of geometry entities is the issue that is also addressed here.

From the design viewpoint, features can be defined as a set of the geometric entities which represent certain shape patterns that have some significance or certain functions. From the machining viewpoint, features can be defined as portions of a work that can be generated with metal removal processes. For different domains, the same geometric portion of a part can be defined as different features, which may specify different meanings. Design features have specific functions and could be generated by certain modeling approaches, while machining features refer to the removal volumes in the bounding box of a part, which can be machined by certain machining operations that is presented in Fig. 17.

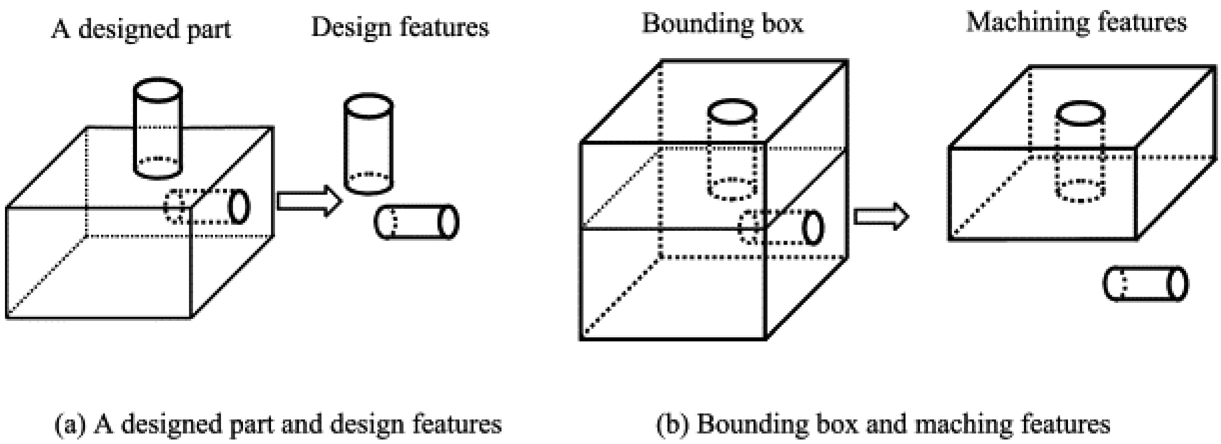


Fig. 17. The relationship of designed part, bounding box, design features and machining features [46]

Design features are classified into form features, which refer to the convex and concave portions in a machining part, and transitional features, which refer to the features generated by trimming edges, blending edges or releasing vertices in the part. If the design feature is a super-feature class, the form feature is a sub-feature class. According to the characteristics of the geometric entities of form features, they could be classified into sub-classes, Inside Form Feature (*IFF*), which is located inside the target surface, and Outside Form Feature (*OFF*), which is formed by the entire target surface with its adjacent surfaces. The target surface refers to the surface in which there are features attached to the surface. For the *IFFs*, they can be further classified into low-level classes, viz. Convex IFF (*CvIFF*) and Concave IFF (*CoIFF*). *CvIFF* is the convex portion in a target surface, while *CoIFF* is the concave geometric portion in the surface. Fig. 18 gives the instances of these form feature classes, in which Fig. 18(a) is a convex portion of the top surface investigated (target surface) and hence a *CvIFF*, while Fig. 18(b) is a *CoIFF* in the target surface. Fig. 18(c) is an *OFF* since it is constituted by the entire target surface and its three adjacent surfaces. The *CoIFF* in Fig. 18(b) is a through cylindrical hole. A blind cylindrical hole or a blind profile hole in a target surface is also a *CoIFF*. The through or blind cylindrical hole or profile hole features in a target surface belong to the detailed-level feature class.

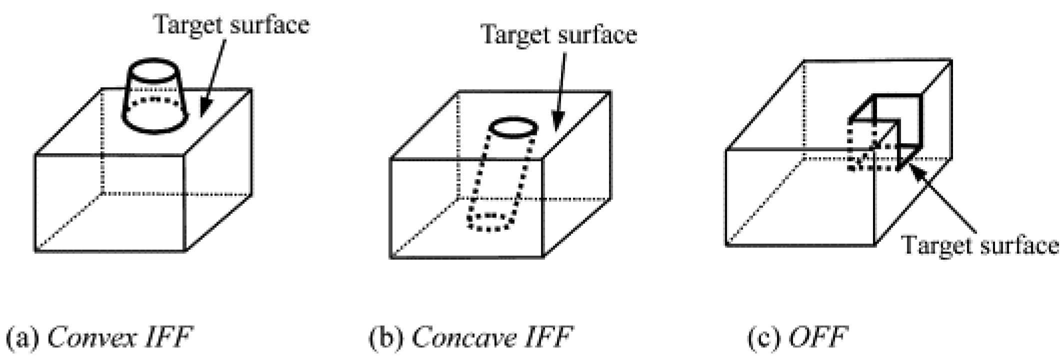


Fig. 18. Inside and Outside form features [46]

Machining features are the volumes to be removed from the bounding box of a part and can be classified into three categories, viz. surface, transitional and form features. The definitions of transitional and form machining features are the same as those of design features. For the surface machining features, they refer to surfaces in a part that have removal volumes from the bounding box, but do not belong to the former two features. Transitional features refer to those features in a part boundary generated by the trimming and blending edges or releasing vertices in the part. The trimming and blending edges or releasing vertex operations convert the edges or vertices in a part into the corresponding surfaces, and these surfaces or their combinations constitute the transitional features. For the surface machining features, they refer to surfaces in a part that have removal volumes from the bounding box, but do not belong to the former two features.

In Fig. 19 shown the relationship diagram of machining feature classes (taxons) and the geometry and topology entity classes. The Geometry Entity and Topology Entity classes are derived from the *B-Rep* model of a part. The Geometry Entity class handles the basic geometric entities such as vertices, edges and surfaces in a part model, while the Topology Entity class deals with the compound geometry entities, viz. shell and loop, and the topological relationships of the basic geometric entities. Since a compound geometric entity has only topological but not geometric information associated with it, it is designated as a Topology Entity [46].

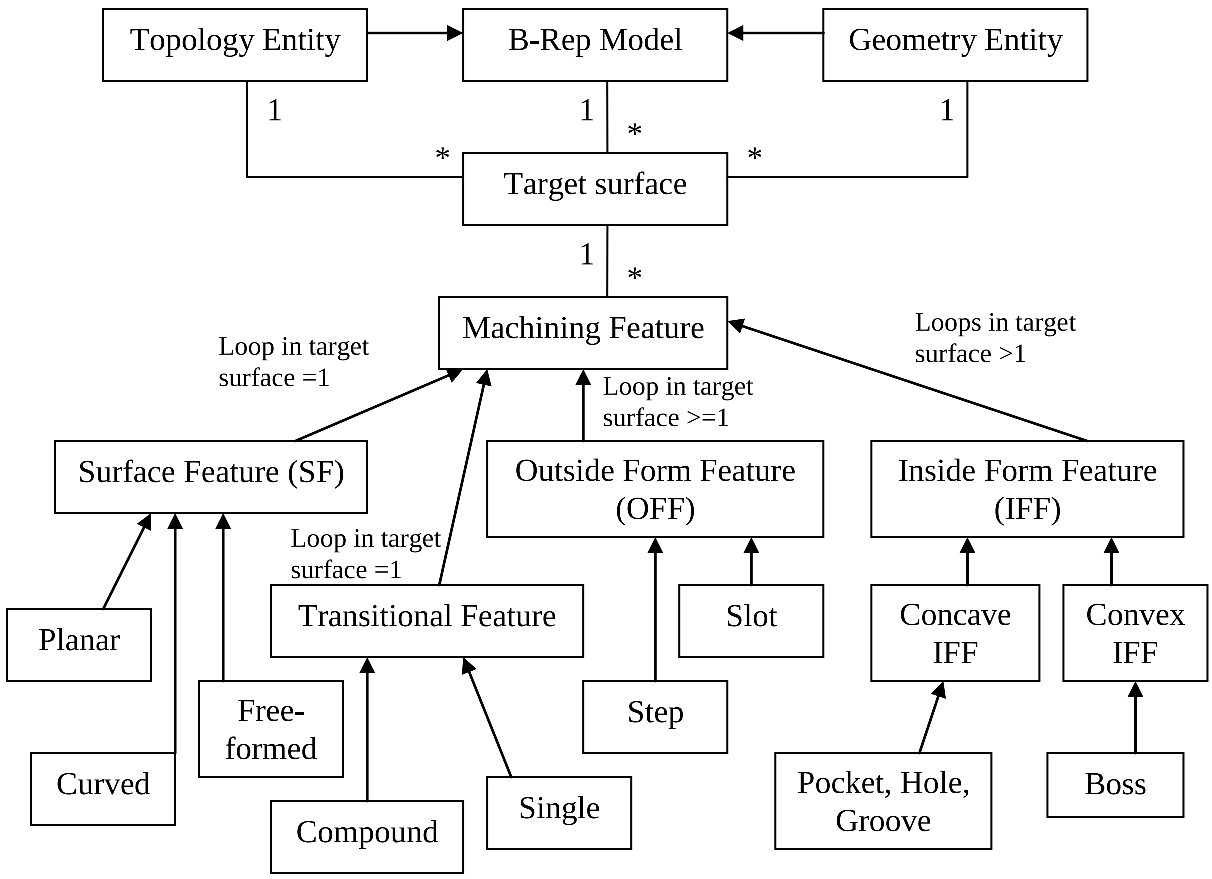


Fig. 19. Class relationship diagram [46]

In this way if there is more than one edge-loop (Loop) in a target surface, *IFFs* exist and these *IFFs* could be *CvIFFs* or *CoIFFs*. Any types of pocket, groove, and hole features in the target surface are *CoIFFs*. A boss on the target surface is a *CvIFF*. On the other hand, if there is only one loop in a target surface, the target surface could then be a surface feature, transitional feature or an *OFF* with the adjacent surfaces of the target surface. For surface features, the feature could be a planar face, curved or free-formed surfaces. For a transitional feature, it could be a single or compound transitional feature.

CHAPTER 4. GROUP TECHNOLOGY AND OPITZ CODE

Group Technology (GT) was developed as a method to index parts and part families in manufacturing process [47]. It facilitated process planning and cell-based manufacturing by imposing a classification scheme (a human-assigned alphanumeric string) to individual machined parts. While there have been a number of attempts to automate the generation of GT codes transition to commercial practice has been limited.

The term “Group technology” signifies a method that aims to analyze and to arrange the parts spectrum and the relevant manufacturing process according to the design and machining similarities so that the basis of groups and families can be established for rationalizing the production processes in the area of small and medium batch sizes [48].

4.1 Historical Background

Firstly the concept of GT was introduced by Mitrofanov [49]. Later his contribution to this concept has been summarized in a two-volume book (Mitrofanov 1983). Burbidge and Ham belong to the group of pioneers advocating the GT concept in the English technical literature (Burbridge 1975 and Ham el al. 1985). Clustering analysis is one of the most frequently applied mathematical tools in GT. There are two basic formulations of the clustering models: first matrix formulation, and second integer programming formulation [47].

In the matrix formulation, judgment regarding the number of clusters and the numbers of elements in each cluster is performed by a human, while in the integer programming formulation both of them are determined by the clustering algorithm.

Some of the most efficient algorithms for matrix formulation of the clustering model have been developed by McCormick et al. (1972), Bhat and Haupt (1979), King (1980), Kusiak (1983, 1985a, 1985c), Kusiak et al. (1985a, 1985b) and Faber and Carter (1986).

4.2 Group Technology, closer look

Group Technology as a manufacturing philosophy nowadays plays a major role in development standardization, manufacturing design, process planning and even material purchasing.

Design for manufacturability, also known as Design For Manufacturing (DFM), is the general engineering concept of product design that lightens the manufacturing process. Its basic idea lies in almost every engineering discipline with different details depending on the manufacturing technology. DFM practice focuses not only on the design aspect of a part but also on its manufacturing productivity, which means relative ease to manufacture a product, part or assembly.

So product design should lean not only on a good implementation but it should also consider manufacturing aspects. It can happen that some design is not producible within some plant. Typically a design engineer creates a model and sends it to production review to get a feedback ‘design review’. This process should be driven accurately in order the product not fail at production stage. If DFM guidelines are not followed, it may result in an iterative design, loss of production time and overall deadline failures. Hence many organizations have adopted concept of Design for Manufacturing.

Depending on types of manufacturing processes there are different DFM practices exist. Such practices help to define various tolerances, rules and common manufacturing checks. On this step Group Technology comes as a great solution that provides product classification considering production rules and constraints. Some GT data includes more specific manufacturing information such as tolerance, material, general dimensions, etc. and design information referred to main shape of the part. To summarize, GT Classification can help to get a good design without penalizing manufacturing practices.

To facilitate and reduce costs of material flow, people, and information between areas, a Layout concept has been introduced [50]. Layout decisions are one of the key facts determining the long-run efficiency of production operations. Layouts have numerous strategic implications

because they establish an organization´s competitive priorities in regard to capacity, processes, flexibility, and cost. They are associated with the tactical decision horizon and are dedicated to the concretion of strategic decisions like, for example, facility location. Configured production systems are input for the operational level, where the goal is to run the given system as efficiently a possible.
To achieve these objectives, a variety of configuration designs have been developed. The most relevant ones are [50]:

- Fixed-position layout: addresses the layout requirements of large, bulky projects;
- Job shop production (Process-oriented layout): deals with low-volume, high-variety production;
- Cellular manufacturing systems (GT layout): arranges machinery and equipment to focus on production of a single product or group of related products;
- Flow shop production (Product-oriented layout): seeks the best personnel and machine utilization in repetitive or continuous production.

As a matter of fact first and second layouts are often described as centralized, and third and fourth as decentralized manufacturing systems. Previously there were only three layout types, but the emergence of group technology as a manufacturing concept has added a new type to layout classification. This new layout type is the GT or cellular layout. It is suitable for both automated and non-automated manufacturing, and can be implemented in new or existing facilities. The concept of GT layout was developed to exploit the advantages of other types of layouts and currently is a great contribution to the worldwide production optimization facilities.

4.3 Group Technology codes structure

A Group Technology code is an alphanumeric string which represents the important information about the products (features, volume, size, characteristic, etc.). Comparing the GT codes of two products is a fast and efficient process for estimating product similarity. GT codes can be used to search a database of products and retrieve the designs and process plans of those products which are similar to a given design, to generate new process plans automatically using a knowledge-based system, and to assess manufacturability of a product design.
Group Technology codes can be made with different structures. Three main structures for creating GT codes are discussed [51].

4.3.1 Hierarchical structures (monocodes)

In this method each digit (or position) in the code represents a feature/sub-group, see Fig. 20.

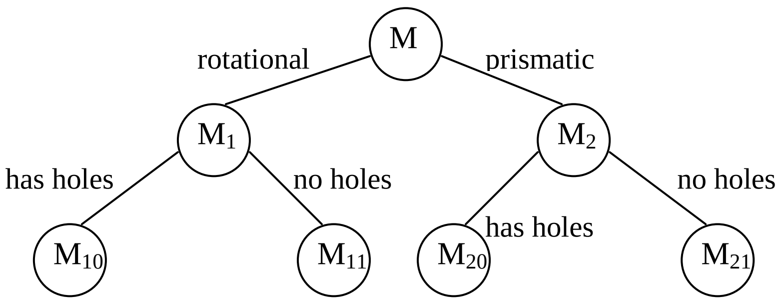


Fig. 20 Monocode structure [52]

In Fig. 20 one digit divide the parts in two groups, rotational and prismatic shapes and other digit divide the parts in different features, with or without holes [52]. In this sense, each subsequent digit is qualified by the preceding digits (or, in an object-oriented sense, each subsequent digit inherits the properties of the previous digits).

Advantages of monocodes:

With just a few digits, a very large amount of information can be stored.

The hierarchical structure allows parts of the code to be used for information at different levels of abstraction.

Disadvantages:

Impossible to get a good hierarchical structure for most features/groups.

Different sub-groups may have different levels of sub-sub-groups, thereby leading to blank codes in some positions.

4.3.2 Attribute codes (polycodes)

A polycode is a chain-type structure where each digit is of fixed significance and a certain digit value in a specific position always represents the same feature. A polycode can be broken down further to either a fixed field or variable field polycode.

A fixed field polycode is a code structure that keeps the order of descriptions the same for all parts. For example, some coding system may have digit 8, 9, 10 representing the length, width and height respectively for any part.

A variable field polycode is a code structure that is the most efficient in terms of code length. There is no restriction on what a digit should present; as a result a code may be of different length for different parts. This type of structure is applicable in industries which have a wide range of parts.

Advantages:

Easy to formulate

Disadvantages:

Less information is stored per digit; therefore to get a meaningful comparison of some shape may cause very long codes.

Comparison of coded parts (to check for similarity) requires much work.

To overcome lengthy codes, this data structure can be implemented via a relational database scheme. Such a scheme will result designing efficient retrieval mechanisms and will lead to a better memory management.

4.3.3 Hybrid codes

This polycode is a combination of the Hierarchical monocode structure and the Attribute polycode codes as shown in Fig. 21.

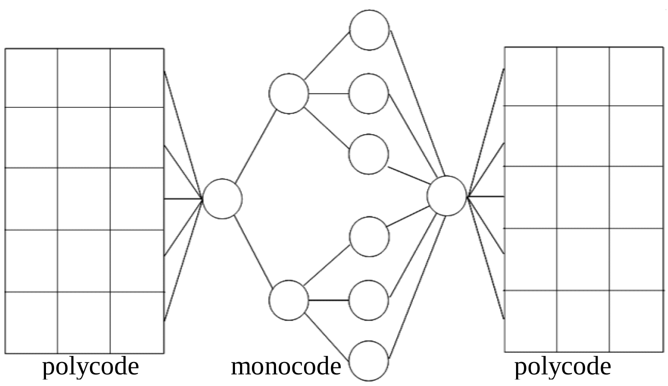


Fig. 21 Hybrid structure [52]

By combining both structures, a code can be developed that will result in efficient storage and retrieval. There will be no false used space due to the variable field polycode and can be relatively short in length due to the incorporation of the monocode structure.

4.4 Different GT codes

The most famous GT codes are explained in the following sections.

4.4.1 Opitz code

One of the most known Group Technology methods is Opitz code. This code system was initially proposed by Henvart Opitz in 1970 at Aachen Technology University in Germany [53]. The code has a maximum of 14 digits and each digit may contain 10 different values (attributes) as indicated in Fig. 22.

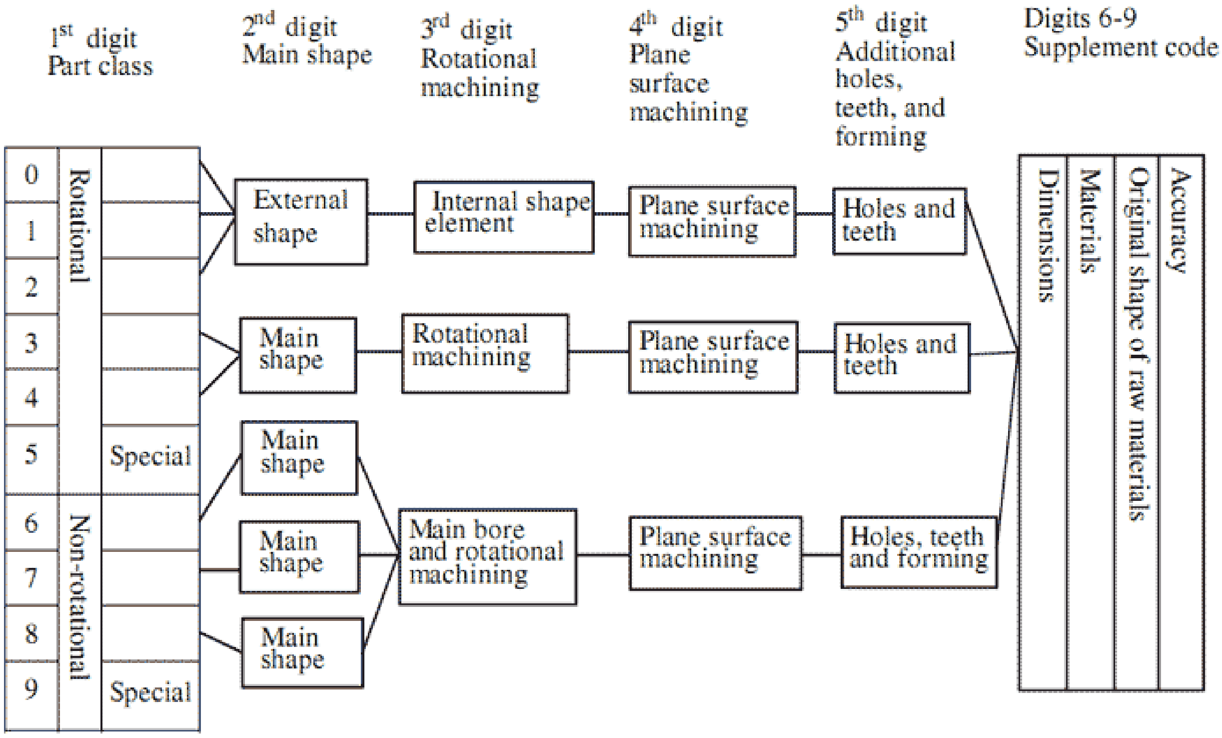


Fig. 22. Basic structure of the Opitz system of parts classification and coding

The first five digits are called the form code and indicate the design or the general appearance of the part and hence assist in design retrieval:

Form code	Supplementary code	Secondary code
1 2 3 4 5	6 7 8 9 10	A B C D

Later, 5 more digits (6-10) were added to the coding scheme, in order to increase the manufacturing information of the specific work part. These five digits are called supplementary code. All five digits are integers, and respectively represent: Dimensions, Material, Original shape of raw stock, and Accuracy of the work part. The extra four digits, A, B, C, and D, called the secondary code, are used by the specific organization to include those characters that are specific to the organization.

Some of the advantages of this code are that it is not proprietary, it is widely used, provides a basic framework for understanding the classification and coding process, it can be applied to machined parts, non-machined parts and purchased parts, it considers both design and manufacturing information.

The applications of Opitz coding system can be pointed as:

- Design: Variety reduction, Recognition of repeat or similar parts;
- Standards: Standard components easily identified, Uniformity of characteristics;
- Production planning: Use of repeat, Grouping parts requiring same machines, Use of standard times;
- Production Control: Suitability for Data processing;
- Production: Parts family manufacture;
- Equipment: Adapting the machine tool to the workpieces required.

In the form code, the first digit is the one that makes the difference between rotational and non-rotational parts and in this digit it is used a dimensional ratio to evaluate the geometry of the shape. For rotational parts the code uses the length (L) and the diameter the components in decreasing order of magnitude (A, B and C). Then the second digit is for external shapes and relevant forms, these features are recognized as stepped, conical or straight contours. Threads and grooves are also important. The third digit is for internal shapes, features are solid, bored, straight or bored in a stepped diameter, threads and grooves are integral part. The fourth digit is for the surface plane machining, such as internal or external curved surfaces, slots and splines. And finally the fifth digit is for auxiliary holes and gear teeth.
In the supplementary code there are four digits, the first one is for diameter or length of the workpiece, the second one is for material used, the third one is for raw materials like round bar, sheet metal, casting or tubing and the fourth digit is for the accuracy of the workpiece.
Table 1 in Appendix shows all possible attributes for the flat parts (first digit of Opitz code = 6). Table 2 in Appendix represents all attributes that classify long parts (first digit of Opitz code = 7). Table 3 in Appendix represents Opitz Code classification attributes for cubic parts (first digit of Opitz code = 8).

4.4.2 MICLASS System

The name MICLASS stands for Metal Institute Classification System, and was developed by the Netherlands Organization for Applied Scientific Research (TNO) of Holland in 1969. After being implemented and applied in different manufacturing industries of Europe, it had been introduced to North America around 1974. The MICLASS was developed to help to automate and standardize a number of design, production, and management functions, which includes:

Standardization of engineering drawings
Retrieval of drawings according to classification number
Standardization of process routing
Automated process planning
Selection of parts for processing on particular groups of machine tools
Machine tool investment analysis

A total number of digits used in MICLASS classification system may vary from 12 to 30 digits. The digits can be divided into two categories. The first twelve digits are claimed as universal and can be applied to any work part. The other 18 digits called supplemental codes and can be used for some specific data of a particular company. Those supplemental digits provide a flexibility to accommodate a broad range of applications. Such as lot size, cost data, and the operation sequence. Design attributes used in the first twelve digits of MICLASS classification are as follows [54]:

1^{st} digit	Main Shape
2^{nd} and 3^{rd} digit	Shape Element
4^{th} digit	Position of shape element
5^{th} and 6^{th} digit	Main Dimension
7^{th} digit	Dimension Ratio
8^{th} digit	Auxiliary Dimension

9th and 10th digit   Tolerance Code
11th and 12th digit   Material Code

One of the features of this coding system is an ability to be coded interactively with a computer. To classify a given work part the user should answer a series of questions which depend on a complexity of current task. The questions in most cases require a simple answer: a numeric value or yes/no. After the end of an analysis the computer assigns code to the part. Then based on this number it is possible to retrieve a similar design or manufacturing procedure.

4.4.3 CODE System

The CODE system is a parts classification and coding system developed and marketed by Manufacturing Data Systems (MDSI) company. The main application of this code is not only to retrieve part design data, but also to support manufacturing process planning, purchasing, tool design and inventory control.

The CODE number has eight digits. For each digit there are 16 possible values (zero through 9 and A through F) which are used to describe the part's design and manufacturing characteristics. The initial digit position indicates the basic geometry of the part and is called the Major Division of the CODE system. This digit would be used to specify whether the shape was a cylinder, flat piece, block, or other. The interpretation of the remaining seven digits depends on the value of the first digit, but these remaining digits form a chain-type structure. Hence the CODE system possesses a hybrid structure [55].

4.5 Evaluation of Group Technology

4.5.1 Advantages

There are several advantages that can be obtained by adopting Group Technology, and some of them are typically realized in the following areas [56]:

- Design;
- Setting time, and Batch quantities;
- Materials handling;
- Production and inventory control;
- Process planning, and
- Effective Supervision and job satisfactions

a. Product design benefits

In the area of product design, the principal benefit will obtained by using from the use of a developed part classification and coding system. When a new part design is required, a designer or an engineer can devote a few minutes to figure out the code of the required work part. Then the existing part designs that match the code can be retrieved to see if one of them will serve the desired function. The few minutes spent searching the design file with the aid of the coding system may save several hours of the designer's time. If the exact part design cannot be found, perhaps a small alteration of the existing design will satisfy the required function. Since a simple or minor change in an existing part would be much less time-consuming than starting from scratch, it would save the designer's precious time which might have otherwise been unnecessarily wasted. Another advantage of Group Technology is that it promotes design standardization. Design features such as inside, corner radii, chamfers, and tolerance are more likely to be standardized with GT.

b. Reduced lead time, Setting time, and Batch quantities

Selection of components to form a family invariably means bringing together components which have similarities, although these similarities may not be obvious at the outset. This usually reduces the time required to reset the machines between batches. This reduction may be taken entirely as reduced set up cost. However it may be more beneficial to sacrifice o this advantage for the sake of smaller batch quantity. If for instance, the average setting time per batch were halved, batch quantities could be reduced equally without increasing total setting cost per year. This could also reduce the working stock of the components in the family by half and also further reduce the throughput time of each batch hence, manufacturing lead time is reduced.

c. Materials handling

Another advantage in manufacturing is a reduction in the work part move and waiting time. The group technology machine layouts lend themselves to an efficient flow of materials through the shop. The contrast can be visible when the flow line cell design is compared to the conventional process-type layout. This will optimize the use of material handling and reduce the cost as well.

d. Production and Inventory control

Several benefits accrue to a company's production and inventory control function as a consequence of group technology. Production scheduling is simplified with group technology. In effect, grouping of machines into cells reduces the number of production centers that must be scheduled. Grouping of parts into families reduces the complexity and size of parts scheduling problem. And for those work parts that cannot be processed through any of the machine cells, more attention can be devoted to the control of these parts. Because of reducing setups and more efficient materials handling within machine cells, manufacturing lead times and work-in-process are reduced and makes the inventory control much simpler.

e. Process Planning

Proper part classification and coding can lead to an automated process planning system. even without an automated process planning system, reductions in the time and cost of process planning can still be accomplished. This is done through standardization. New part designs are identified by their code as belonging to a certain parts family, for which the general process routing is already known.

f. More Effective Supervision and Job Satisfaction

In traditional batch production, with the machines laid out by type, the supervisor inspects a group of machines. Thus he supervises some of the operations on many components but perhaps may not supervise the complete production of any component. In Group Technology, a supervisor can supervise a group of machines and an operator which manufactures from raw material to a finished state of all the components in a family. This means that the supervisors must acquire knowledge of all type of machines with which they have not previously acquainted with. Another fundamental change which often has to be accepted is that some of the operators in a group have to operate more than one machine. But once those changes have been over come, the following important benefits will occur. Supervision of quality will be more effective. Work part quality is more easily traced to a particular machine cell. A quality assistant should in any case be responsible for seeing that the quality of the work done conform to the standard set forth. This is simpler if one supervisor is responsible for the whole production from start to finish. Tractability of part defects is sometimes very difficult in a conventional process-type layout, and quality control suffers as a result. In traditional process layout the operators see only part of the operation as being performed in their specific department, so they don't get the chance to see through the whole operation. But in the case of Group Technology the complete process is handled in a single cell and so workers are able to realize their contributions to the firm more clearly. This tends to cultivate an improved worker attitude and a higher level of job satisfaction.

4.5.2 Disadvantages

The most quoted disadvantage of Group Technology is that machine utilization is likely to be lower than with the traditional functional layout. As it was shortly presented, this is more realistically regarded as offsetting of some of the benefit of lower investment in work-in-progress. Probably the biggest disadvantage is the effort required to changeover to a Group Technology method of working, perhaps combined with some risk, if one has not done it before, of not obtaining sufficient benefit to justify the effort.

CHAPTER 5. THE PROPOSED FEATURE RECOGNITION METHOD

5.1 Introduction

In previous chapters STEP was chosen as an input CAD format that stores geometry and topology of a particular engineering part. During the process of analysis, the incoming format should be parsed for the purpose of Opitz Code feature extraction with further assignment to Opitz Classification System. There is a method presented in this chapter that enables product classification and product feature recognition from standardized STEP representation in conformity with Opitz Classification.

5.2 Opitz feature description of non rotational components

5.2.1 Component class (1st digit of Opitz code)

To relate a part to a predefined component class there are 3 measures should be considered: part length, part width and part height as shown in Fig. 23.

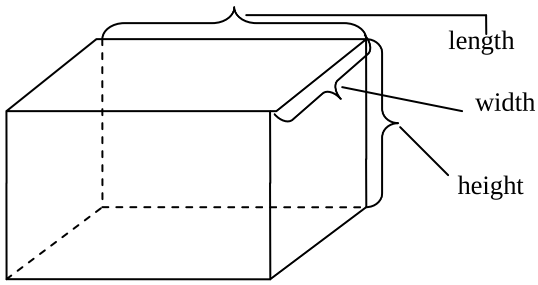


Fig. 23 Shape measures

According to Opitz code specification:

- Non rotational flat part (1st digit of Opitz code = 6): length / width must be <= 3 and length / height >= 4
- Non rotational cubic part (1st digit of Opitz code = 7): length / width must be <= 3 and length / height < 4
- Non rotational long part (1st digit of Opitz code = 8): length / width must be > 3

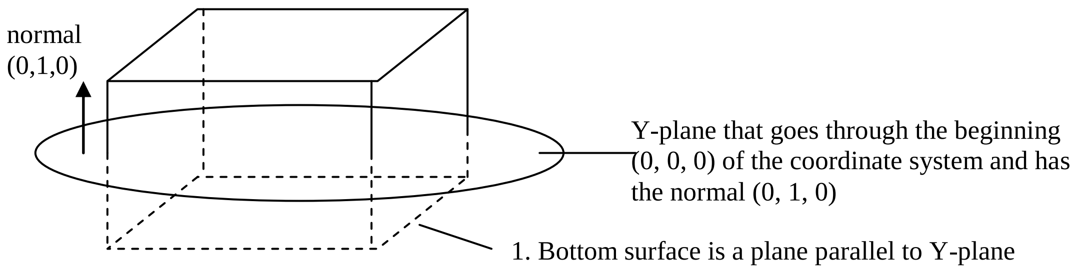
5.2.2 Overall shape of non rotational components

Terms ‘surface’, ‘outer loop’, ‘plane’ are referred from 1.2.1.5 section.

5.2.2.1 Non rotational flat components (1st digit = 6)

Rectangular flat component (2nd digit = 0) can be identified after consideration of the following aspects as depicted in Fig. 24:

1. Bottom surface should be found which is plane (not cylindrical or conical surface), parallel to Y-plane and has the minimal y coordinate
2. Area of the bottom surface should be bigger than the Y-plane cross-section of each cylinder orthogonal to this plane.
3. All adjacent surfaces to an outer loop of the bottom plane must be orthogonal to this plane as well
4. Outer loop of the bottom plane must be a rectangle.



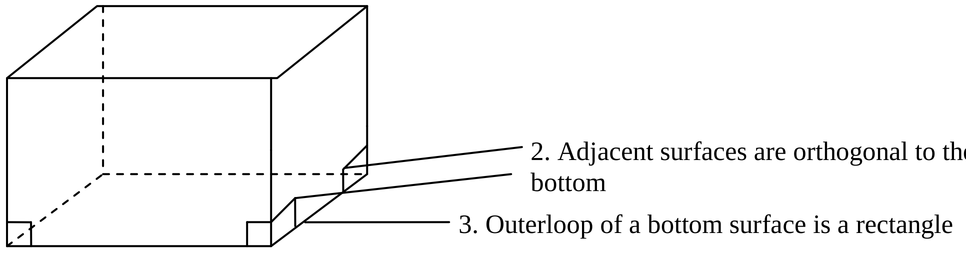


Fig. 24 Main aspects of feature extraction of the rectangular flat component

Right Angle or Triangular flat component (2nd digit = 1) overall shape can be identified after consideration of the following aspects (Fig. 25):

1– 3. Same rules as in previous part

4. Outer loop of the bottom plane must be a triangle.

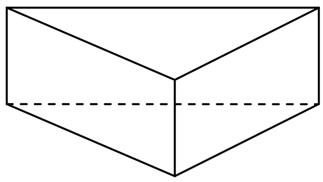


Fig. 25 Example of triangular flat component

Angular flat component (2nd digit = 2) overall shape can be identified after consideration of the following aspects (Fig. 26):

1– 3. Same rules as in previous part

4. Outer loop of the bottom plane should have more than 4 edges, with the same angle between them.

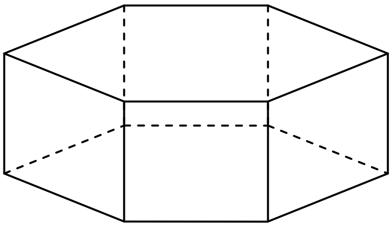


Fig. 26 Example of triangular flat component

Rectangular with circular deviations flat component (2nd digit = 3) overall shape can be identified after consideration of the following aspects (Fig. 27):

1– 3. Same rules as in previous part

4. Outer loop of the bottom plane should consist of linear edges and one circular edge.

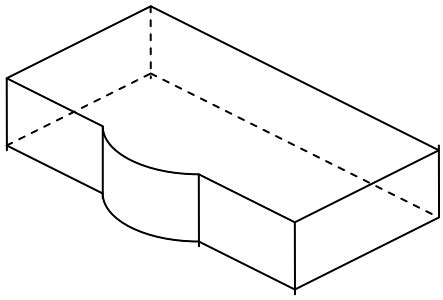


Fig. 27 Example of rectangular with circular deviation flat component

Rectangular or right angled with small deviations flat component (2nd digit = 5) overall shape can be identified after consideration of the following aspects (Fig. 28):

1– 3. Same rules as in previous part

4. Outer loop of the bottom plane should consist of linear edges that form a shape with small deviations.

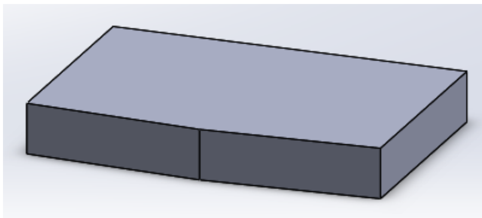


Fig. 28 Example of rectangular component with small frontal deviation

5.2.2.2 Non rotational long components (1st digit = 7)

Rectangular long component with uniform cross-section (2nd digit = 0) can be identified after consideration of the following aspects:

1. Front and back surfaces should be found which are plane (not cylindrical or conical surface) and parallel to Z-plane having the maximal and minimal z-coordinates within current shape respectively (Fig. 29)
2. Outer loop of the front plane equals to the one of the back plane: for each vertex of the front plane's outer loop there is should be a vertex on the outer loop of the back plane having the same x, y coordinates (Fig. 29)
3. Shape axis should be straight and have a direction (0, 0, +/-1). This condition is satisfied when all adjacent surfaces to front or back plane's outer loop are planes.
4. Outer loop of the front and back plane must be rectangular

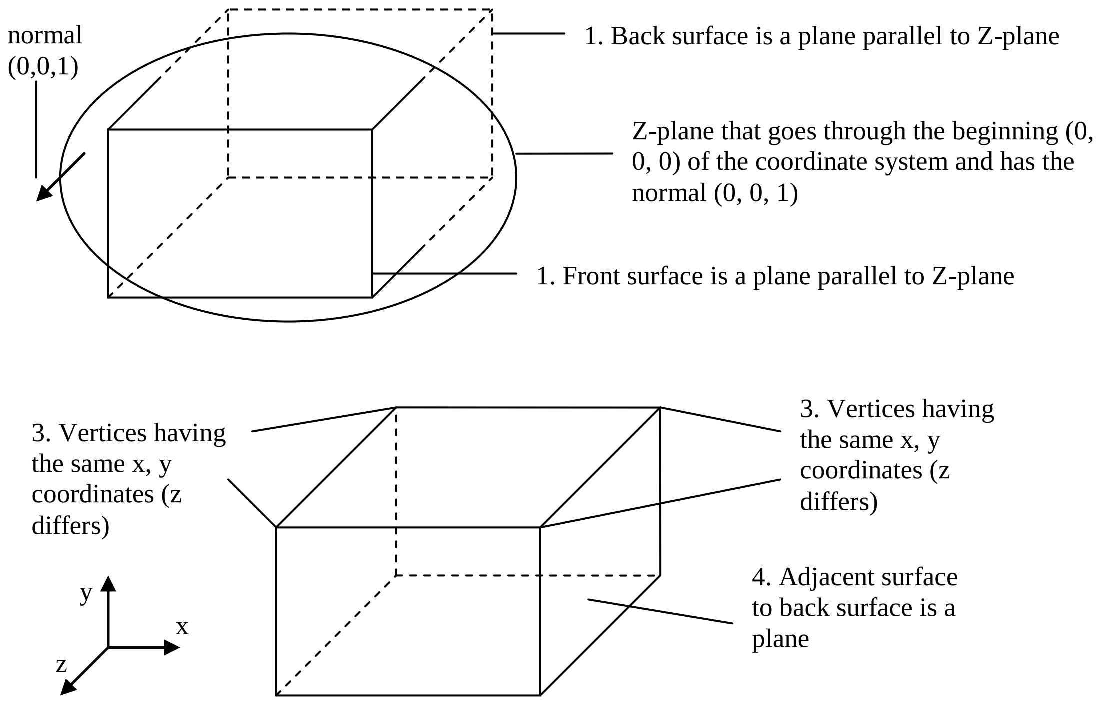


Fig. 29 Example of the rectangular long component with uniform cross-section

Right Angle or Triangular long component (2^{nd} digit = 1, Fig. 30) with uniform cross-section can be identified after consideration of the following aspects:

1– 3. Same rules as in previous part

4. Outer loop of the front and back plane must be triangular

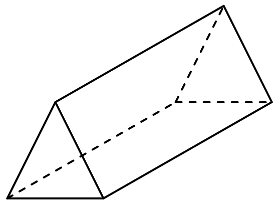


Fig. 30 Example of the triangular long component with uniform cross-section

Any uniform cross-section other than 0 and 1(not rectangular and not triangular) long component (2^{nd} digit = 2, Fig. 31) can be identified after consideration of the following aspects:

1– 3. Same rules as in previous part

4. Outer loop of the front and back plane must not be triangular and rectangular

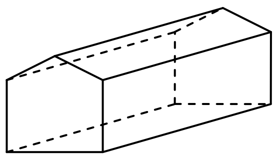


Fig. 31 Example of the long component with uniform cross-section other than 0 and 1 (not rectangular and not triangular)

Rectangular long component with varying cross-section (2nd digit = 3, Fig. 32) can be identified after consideration of the following aspects:

1. Front and back surfaces should be found which are plane (not cylindrical or conical surface) and parallel to Z-plane having the maximal and minimal z-coordinates within current shape respectively
2. Shape axis should be straight. This condition is satisfied when all adjacent surfaces to front or back plane's outer loop are planes.
3. Outer loop of the front plane not equals to the one of the back plane: not for each vertex of the front plane's outer loop there is should be a vertex on the outer loop of the back plane having the same x, y coordinates
4. Outer loop of the front and back plane must be rectangular

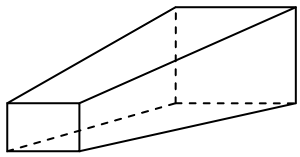


Fig. 32 Example of the rectangular long component with varying cross-section

Right Angle or Triangular long component (2nd digit = 4, Fig. 33) with varying cross-section can be identified after consideration of the following aspects:

1– 3. Same rules as in previous part

4. Outer loop of the front and back plane must be triangular

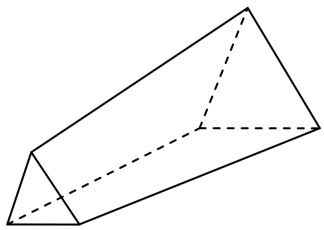


Fig. 33 Example of the triangular long component with varying cross-section

Any varying cross-section other than 3 and 4 (not rectangular and not triangular) long component (2nd digit = 5, Fig. 34) can be identified after consideration of the following aspects:

1– 3. Same rules as in previous part

4. Outer loop of the front and back plane must not be triangular and rectangular

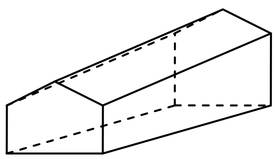


Fig. 34 Example of the long component with varying cross-section other than 3 and 4 (not rectangular and not triangular)

Rectangular, angular or other cross-section, shape axis curved long component (2nd digit = 6, Fig. 35) can be identified after consideration of the following aspects:

1. Front and back surfaces should be found which are plane (not cylindrical or conical surface) and parallel to Z-plane having the maximal and minimal z-coordinates within current shape respectively
2. Shape axis should be curved.
3. Outer loop of the front and back planes should be rectangular, angular or have other cross-sections.

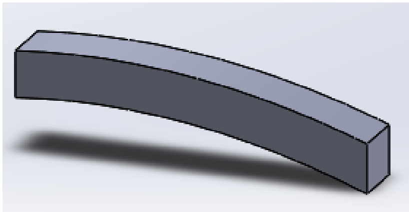


Fig. 35 Example of the long component with curved shape axis

5.2.2.3 Non rotational cubic components (1st digit = 8)

Rectangular prism cubic component (2nd digit = 0) can be identified using the same aspects as for Rectangular flat component from 5.2.2.1 section.

Right angled or triangular cubic component (2nd digit = 1) can be identified using the same aspects as for Right angled or triangular flat component from 5.2.2.1 section.

Compounded of Rectangular Prisms cubic component (2nd digit = 2, Fig. 36) can be identified using the following aspects:

1. Bottom surface should be found which is plane (not cylindrical or conical surface) and parallel to Y-plane
2. All adjacent surfaces to an outer loop of the bottom plane must be orthogonal to this plane as well
3. Outer loop of the bottom plane should have more than 5 linear edges, with an angle of 90 degrees between any adjacent pair of edges as depicted in Fig. 36 (right).

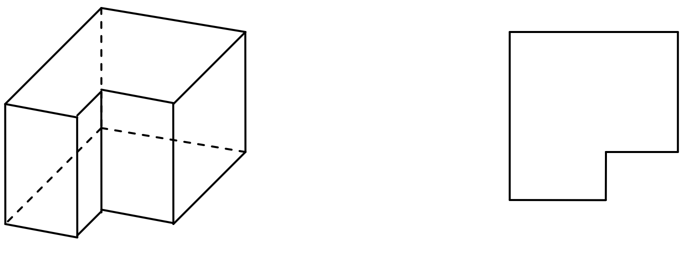


Fig. 36 Cubic component compounded of rectangular prisms (left), its bottom plane (right)

Box-like compounded of Rectangular Prisms cubic component (2nd digit = 6, Fig. 37) can be identified using the following aspects:

1. Bottom surface should be found which is plane (not cylindrical or conical surface) and parallel to Y-plane

2. All adjacent surfaces to an outer loop of the bottom plane must be orthogonal to this plane as well
3. Outer loop of the bottom plane should have more than 5 linear edges, with an angle of 90 degrees between any adjacent pair of edges as depicted in Fig. 36 (right).
4. One inner loop within bottom plane should exist, having the same shape as outer loop.

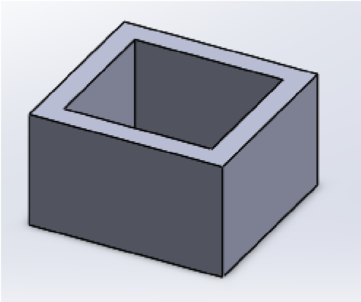


Fig. 37 Box-like cubic component compounded of rectangular prisms

5.2.3 Principal bore, rotational surface machining (3rd digit of Opitz code)

Presented in this section rules to identify 'bore'-features are shown for (0, 1, 0) direction, but they should be applied also for directions: (1, 0, 0), (0, 0, 1).

No rotational machining could be exposed with a consideration of the following aspects:

1. Bottom surface should be found which is plane (not cylindrical or conical surface) and parallel to Y-plane
2. Bottom surface should have no inner loops

One main bore can be identified with a usage of the aspects as shown in Fig. 38:

1. Bottom surface should be found which is plane (not cylindrical or conical surface) and parallel to Y-plane
2. Bottom surface should have one inner loop which is a circle
3. Adjacent surface (that is a cylinder) to this inner loop should be orthogonal to the surface of current inner loop.

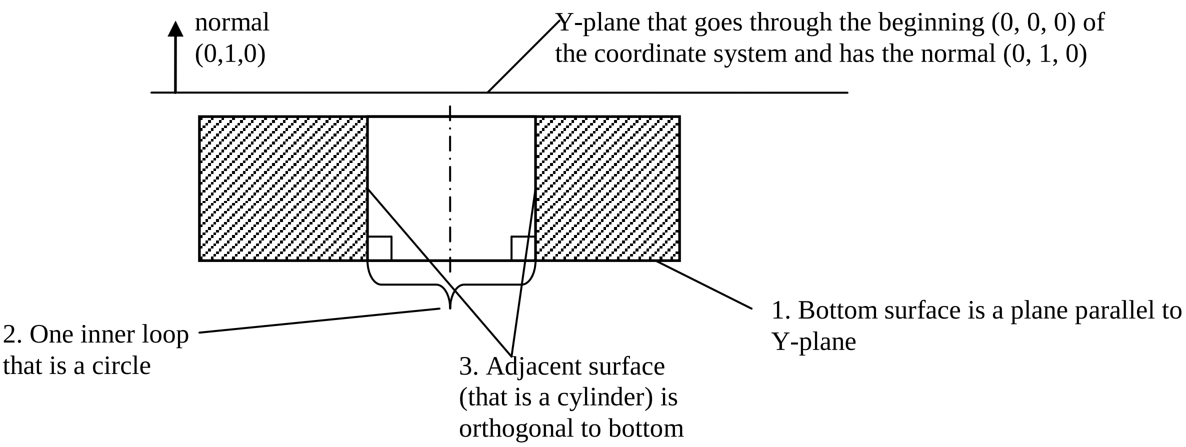


Fig. 38 Cross section by Z-plane through a part with one main bore

Two (or more than two) main bores can be extracted using the following aspects:

1. Bottom surface should be found which is plane (not cylindrical or conical surface) and parallel to Y-plane
2. Bottom surface should have two (or more) inner loop which is a circle
3. Adjacent surface (that is a cylinder) to each inner loop should be orthogonal to the surface of current inner loop.

5.2.4 Plane surface machining (4th digit of Opitz code)

Chamfers for a given part are exposed whether the following statements are satisfied:

1. Top surface should be found which is plane (not cylindrical or conical surface) and parallel to Y-plane (Fig. 39)
2. All adjacent surfaces to top plane should have the same angle between the normal of current surface and the Y-oriented normal (0, 1, 0)

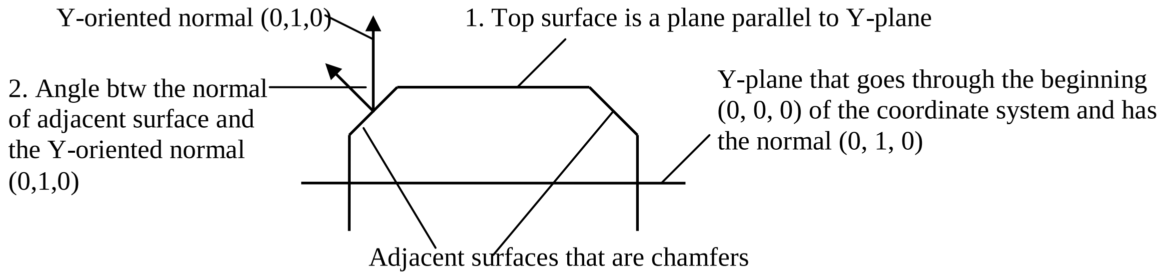


Fig. 39 Cross section by Z-plane through a part that has chamfers

Stepped plane surface

To check whether the current part has a stepped plane surface machining, only one condition should be evaluated: total amount of plane surfaces that are parallel to Y-plane must be more than 2 and there are no grooves for the current detail.

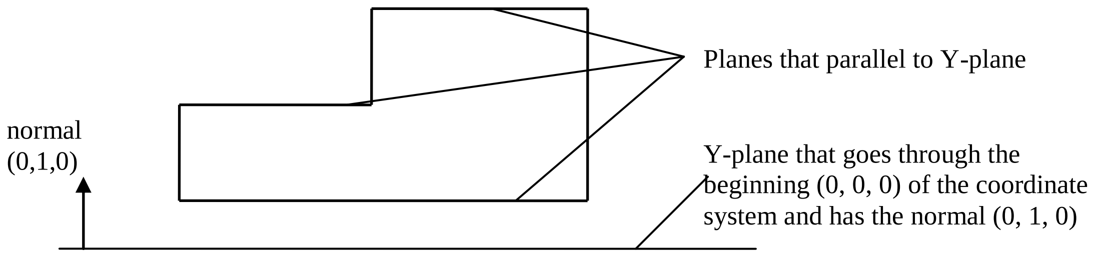


Fig. 40 Cross section by Z-plane through a part with stepped top machining

For non-rotational parts there are 3 Opitz code groups of plane surface machining: one plane surface, stepped plane surface, stepped surface vertically inclined and/or opposed. These groups differ in the methods of machining, having in the result the same part shape. It means for feature recognition of these three groups the same rules are applied.

Curved surface
The curved face machining can be identified in the following way: if any cylindrical surface with normal that lies within Y-plane is found (i.e. normal = (*, 0, *)) and a bottom plane is identified (bottom plane can be within non rotational parts only), when a positive result is concluded.

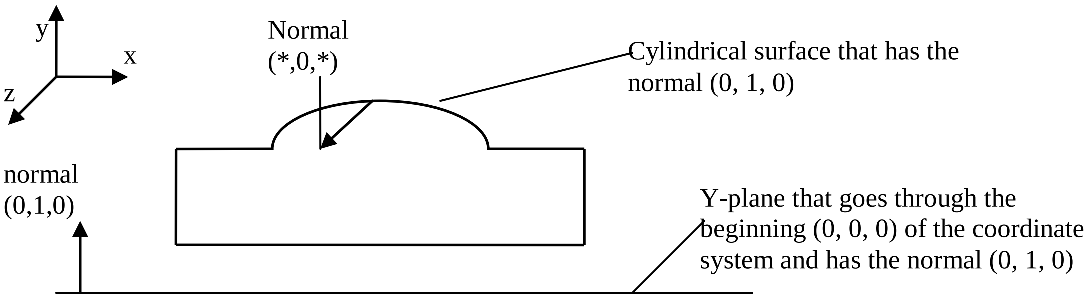


Fig. 41 Cross section by Z-plane through a part with the curved top machining

Groove and/or slot for a given part are exposed whether the following statements are satisfied as it is illustrated in Fig. 42:
1. Top surface should be found which is plane (not cylindrical or conical surface) and parallel to Y-plane
2. Found top surface should have one or more inner loops that are not circles.

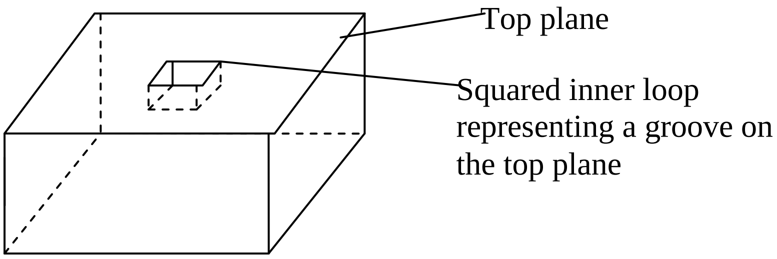


Fig. 42 Groove or slot identification

Surface with no plane surface machining is a shape that has:
1. Top surface should be found which is plane (not cylindrical or conical surface) and parallel to Y-plane
2. No chamfers are found for the found top surface
3. Current shape has no stepped plane surfaces
4. Current shape has no curved machining
5. Has no grooves (slots)

5.3 Opitz feature description of rotational components

5.3.1 Component class of rotational components (1st digit of Opitz code)
To relate a part to a predefined component class there are 3 measures should be considered: part length, part diameter as shown in Fig. 43.

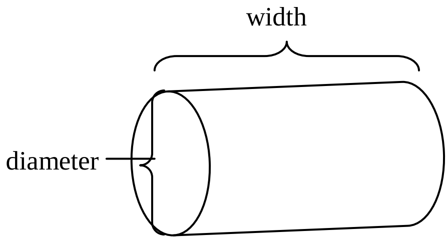


Fig. 43 Rotational part shape measures

According to Opitz code specification:

- Rotational part (1st digit of Opitz code = 0): length / diameter <= 0.5
- Rotational part (1st digit of Opitz code = 1): 0.5 < length / diameter < 3
- Rotational part (1st digit of Opitz code = 2): length / diameter >= 3

5.3.2 External shape, external shape elements of rotational components (2nd digit of Opitz code)

Presented in this section rules to identify cylinders are shown for (0, 0, 1) direction, but they should be applied also for directions: (1, 0, 0), (0, 1, 0).

Rotational component with no shape elements (2nd digit = 0) can be identified after consideration of the following aspects as depicted in Fig. 44:

1. Front and back surfaces should be found which are plane (not cylindrical or conical surface) and parallel to Z-plane having the maximal and minimal z-coordinates within current shape respectively
2. There is only one cylindrical surface should be identified that is orthogonal to the plane of the back surface.

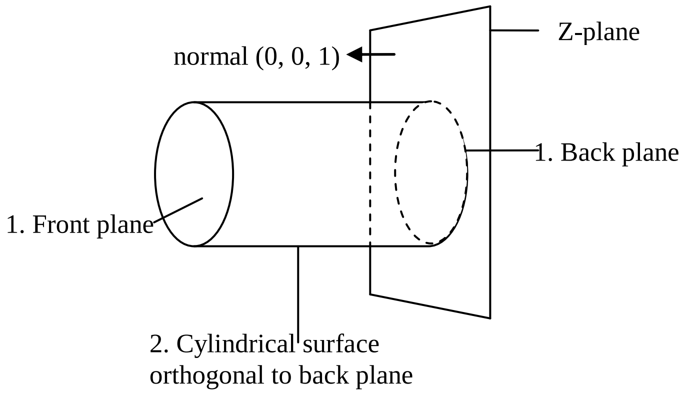


Fig. 44 Rotational part with no shape elements

Rotational component stepped to one end with no shape elements (2nd digit = 1) can be identified after consideration of the following aspects (Fig. 45):

1. Front and back surfaces should be found which are plane (not cylindrical or conical surface) and parallel to Z-plane having the maximal and minimal z-coordinates within current shape respectively
2. There are 2 cylindrical surfaces should be identified that are orthogonal to the plane of the back surface.

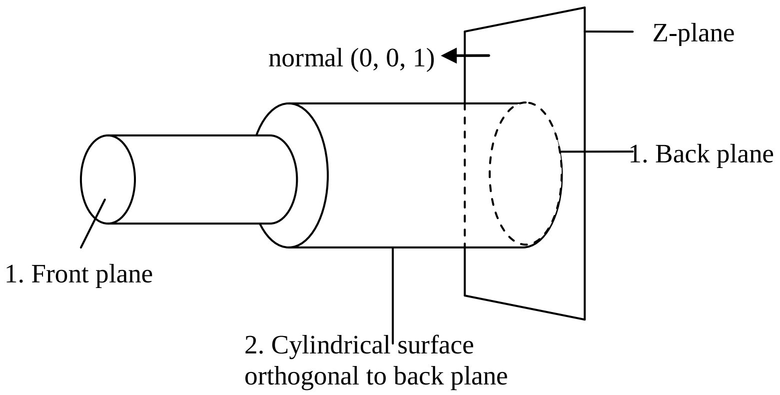


Fig. 45 Rotational part stepped to one end with no shape elements

Rotational component stepped to one end (or smooth) with a groove (slot) (2nd digit = 3) can be identified after consideration of the following aspects (Fig. 46):

1. Front and back surfaces should be found which are plane (not cylindrical or conical surface) and parallel to Z-plane having the maximal and minimal z-coordinates within current shape respectively
2. There are 2 (or 1 for smooth part) cylindrical surfaces should be identified that are orthogonal to the plane of the back surface
3. Grooves count = 1 (information about groove identification in section 5.2.4)

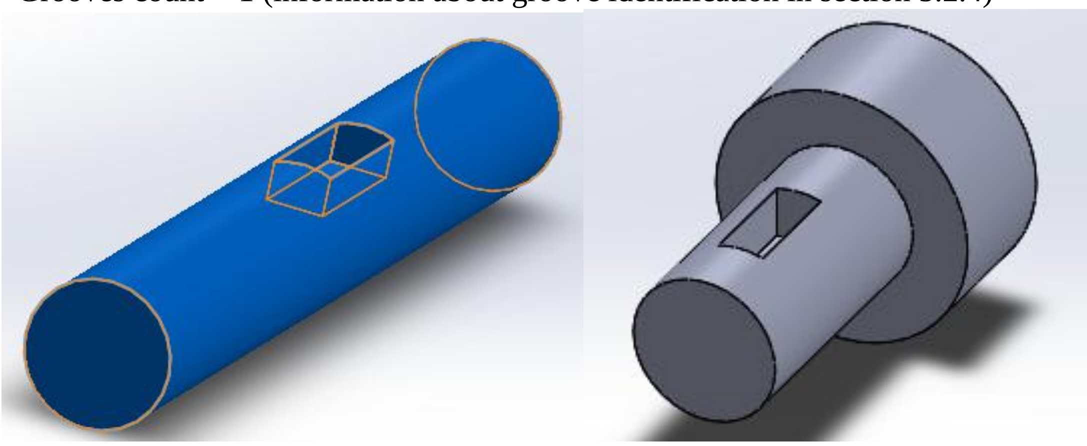


Fig. 46 Rotational part with a groove smooth (left) and stepped to one end (right)

Rotational component stepped to both ends with no shape elements (2nd digit = 4) can be identified after consideration of the following aspects (Fig. 47):

1. Front and back surfaces should be found which are plane (not cylindrical or conical surface) and parallel to Z-plane having the maximal and minimal z-coordinates within current shape respectively
2. There are 3 cylindrical surfaces should be identified that are orthogonal to the plane of the back surface.

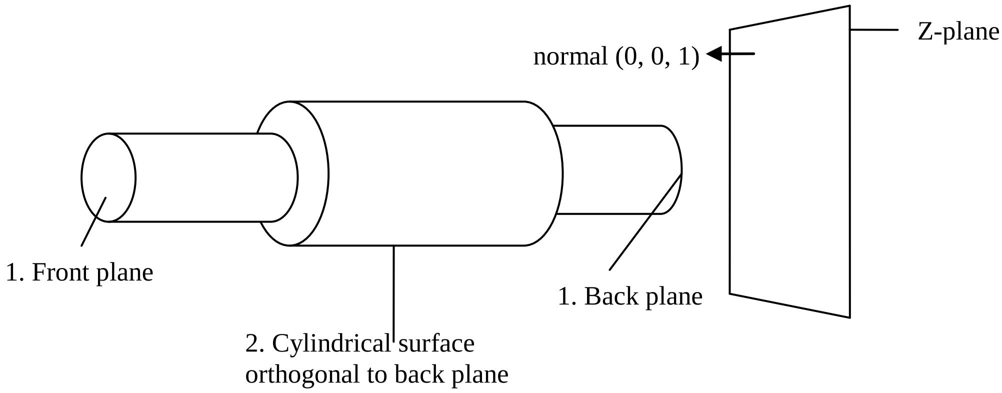


Fig. 47 Rotational part stepped to both ends with no shape elements

Rotational component stepped to both ends with grooves (or slots) (2nd digit = 6) can be identified after consideration of the following aspects (Fig. 48):

1. Front and back surfaces should be found which are plane (not cylindrical or conical surface) and parallel to Z-plane having the maximal and minimal z-coordinates within current shape respectively
2. There are 3 cylindrical surfaces should be identified that are orthogonal to the plane of the back surface
3. Grooves count = 1 or 2 (information about groove identification in section 5.2.4)

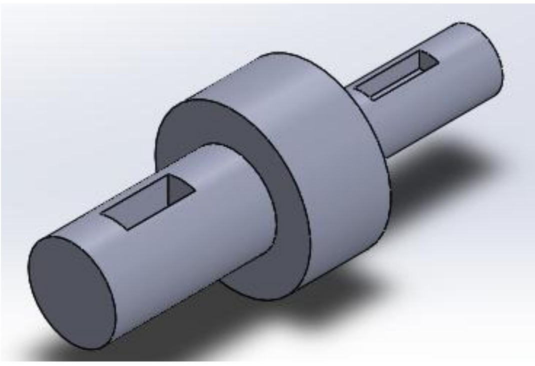


Fig. 48 Rotational part stepped to both ends with 2 grooves

Rotational component with other external shape elements (more than 10 functional diameters) (2nd digit = 9) can be identified after consideration of the following aspects:

1. Front and back surfaces should be found which are plane (not cylindrical or conical surface) and parallel to Z-plane having the maximal and minimal z-coordinates within current shape respectively
2. There are > 10 cylindrical surfaces should be identified that are orthogonal to the plane of the back surface

5.3.3 Internal shape, internal shape elements of rotational components (3rd digit of Opitz code)

Without through bore or blind hole component can be identified with the usage of the aspects (3rd digit of Opitz code = 0):

1. Front and back surfaces should be found which are plane (not cylindrical or conical surface) and parallel to Z-plane having the maximal and minimal z-coordinates within current shape respectively

2. There are > 0 cylindrical surfaces should be identified that are orthogonal to the plane of the back surface
3. There is no inner loops on the front face and on the back face

Smooth (Fig. 49 left) or stepped to one end (Fig. 49 right) component with internal no shape element can be identified with the usage of the following aspects (3^{rd} digit of Opitz code = 1):

1. Front and back surfaces should be found which are plane (not cylindrical or conical surface) and parallel to Z-plane having the maximal and minimal z-coordinates within current shape respectively
2. There are 1 or 2 cylindrical surfaces should be identified that are orthogonal to the plane of the back surface
3. There is 1 circled inner loop on the front face that equal to the one on the back face.

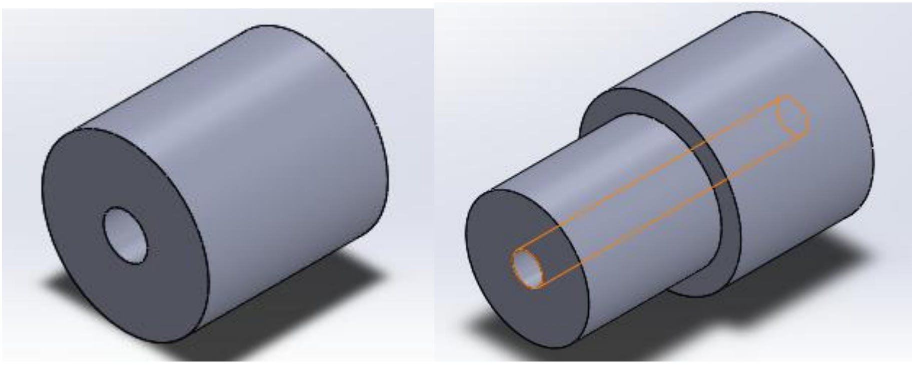


Fig. 49 Rotational smooth (left) and stepped to one end (right) part with internal no shape element

Stepped to both ends component with internal no shape element (Fig. 50) can be identified with the usage of the following aspects (3^{rd} digit of Opitz code = 4):

1. Front and back surfaces should be found which are plane (not cylindrical or conical surface) and parallel to Z-plane having the maximal and minimal z-coordinates within current shape respectively
2. There are 3 cylindrical surfaces should be identified that are orthogonal to the plane of the back surface
3. There is 1 circled inner loop on the front face that equal to the one on the back face.

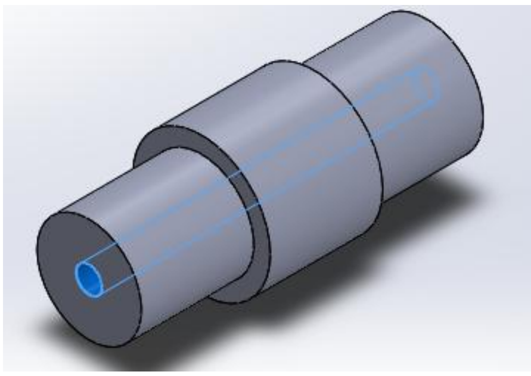


Fig. 50 Rotational stepped to both ends part with internal no shape element

5.4 Proposed rules for feature recognition

The analysis of the previous sections leaded to the following rules for feature recognition being formulated:

R1. Part dimension measures extraction (length, width, height)
R1.1 Flat non rotational part identified (length / width <= 3 and length / height >= 4)
R1.2 Cubic non rotational part identified (length / width <= 3 and length / height < 4)
R1.3 Long non rotational part identified (length / width > 3)
R1.4 Rotational part (1st digit of Opitz code = 0): length / diameter <= 0.5
R1.5 Rotational part (1st digit of Opitz code = 1): 0.5 < length / diameter < 3
R1.6 Rotational part (1st digit of Opitz code = 2): length / diameter >= 3
R2. Operations with plane surfaces parallel to Y-plane.
R3. Operations with plane surfaces parallel to Z-plane.

R2 rule set can be presented as a tree depicted in Fig. 51. All rules within current tree are hierarchy dependent, starting from the root node going to its child nodes. For example, to identify a ring machining on some non-rotational part, there is should be satisfied a chain of rules R2 -> R2.1 -> R2.1.2 -> R2.1.2.3 -> R2.1.2.3.1 within given tree.

R3 rule set can also be presented as a tree illustrated in Fig. 52. All rules within this tree are hierarchy dependent, starting from the root node going to its child nodes. For instance, to identify a long non-rotational part that has a rectangle as a cross section, there is should be satisfied the following chain of rules R3 -> R3.1 -> R3.1.1 -> R3.1.1.1 within given tree.

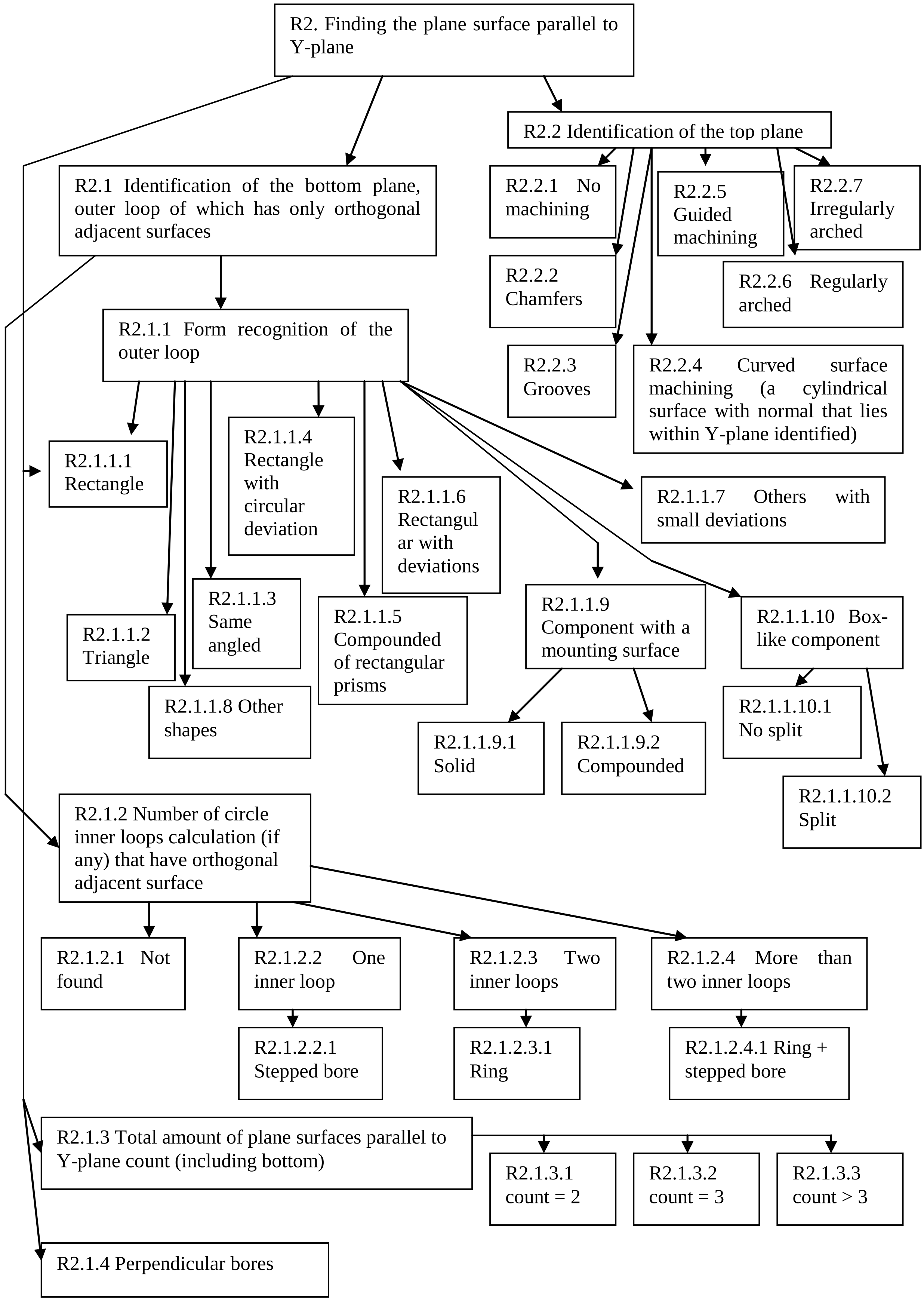


Fig. 51 Rules of feature recognition for surfaces parallel to Y-plane

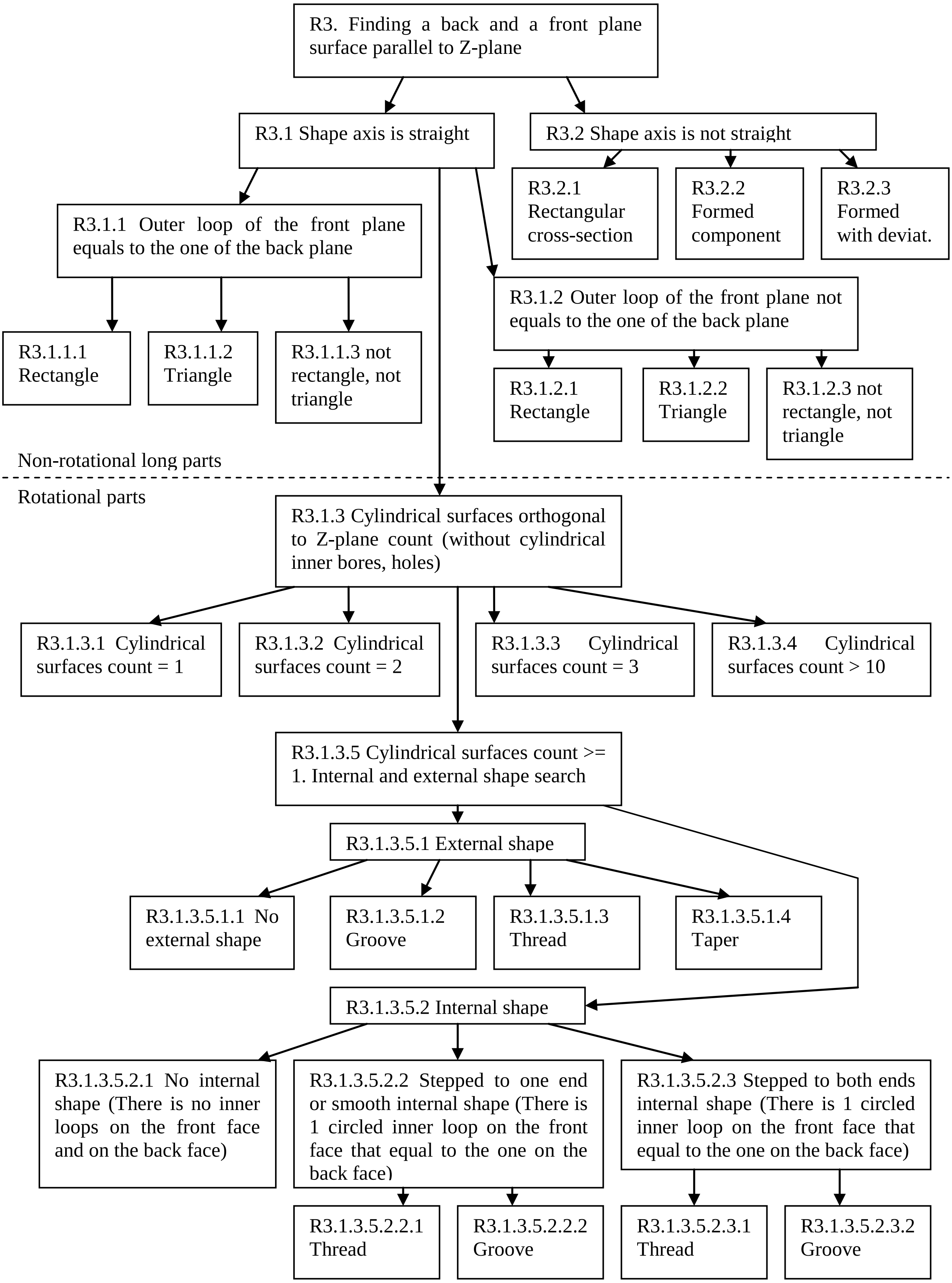


Fig. 52 Rules of feature recognition for surfaces parallel to Z-plane

CHAPTER 6. IMPLEMENTATION AND EVALUATION

6.1 Implementation

In order to implement, test and evaluate the proposed methodology Java SE programming language by Oracle with JUnit testing abilities and SolidWorks 2013 – a CAD software to design engineering parts were employed. Java was used to implement current methodology and algorithms; Graphical User Interface (GUI) was also developed by Java to support user interaction with basic programmed functionality (Fig. 54). When algorithm finishes its work, detailed trace log is printed out on the main form showing the identified engineering part features together with generated Opitz Code of current detail. For example, for cubic non-rotational part with upper groove depicted in Fig. 53 the following features are recognized and rendered (Fig. 54): non-rotational, cubic, machining 2 stepped groove, no bores. And the result Opitz Code is also printed out in presented GUI as it is depicted in Fig. 54.

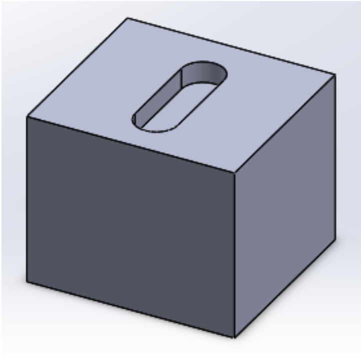


Fig. 53 Cubic example part

There is a button “Open STEP file” located on the main form of developed program to invoke file chooser window that allows to pick demanded STEP-file (*.stp). After the file is chosen, path to it is rendered on the top region of the main form and the feature recognition algorithm is started.

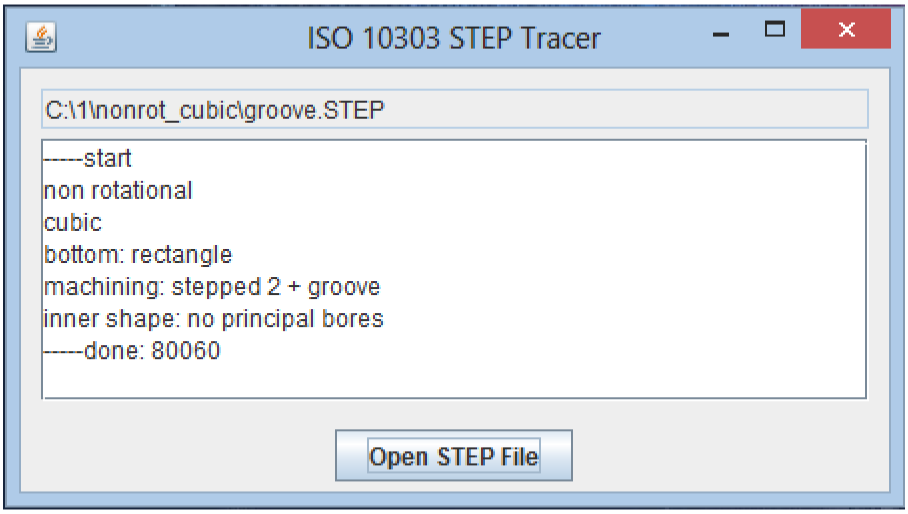


Fig. 54 Main form of developed software

In this way during feature recognition process STEP-file of some part is parsed by means of Java to extract all data entities of given file for further processing. As well as the method developed in Chapter 5 was implemented to provide proper system performance.

SolidWorks is able to design 2D and 3D engineering parts. Current thesis was focused on the models of 3D parts only; these models were designed using Boundary Representation techniques (BRep). To test the basic functionality of developed software a database of details representing basic part features was created using SolidWorks environment with an ability to export designed part to STEP-file. To automate the testing of mentioned database JUnit technology was employed that allows a programmed placement of STEP-files one by one to developed software with a comparison to desired result.

6.2 Evaluation

As mentioned in previous section a set of STEP-files was designed by means of SolidWorks to test basic functionality of the program during the process of its development. But for the evaluation of the final implemented methodology there is a need in real engineering parts taken from up-to-date industries. For this purpose online sources of various manufacturers were investigated in order to find real production CAD-models covering an entire area of different product features. As a result of this investigation 50 parts were selected on web aggregator http://www.tracepartsonline.net/ that combines various CAD parts sources for further feature extraction and analysis of generated Opitz Code.

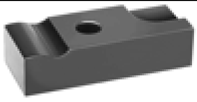
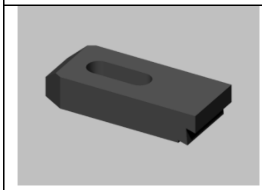
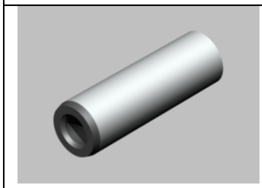
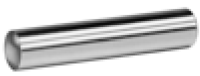
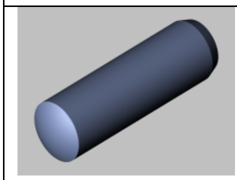
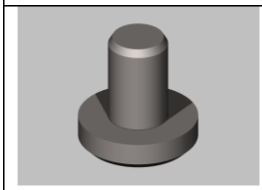
In Table 1 presented four columns: an image of investigated detail, a reference to the manufacturer that provides it, a result trace log that generated by developed program, a comment describing the result of feature recognition.

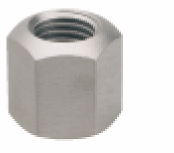
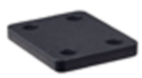
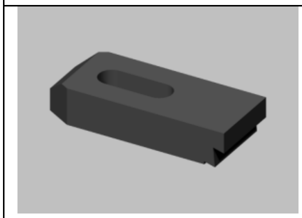
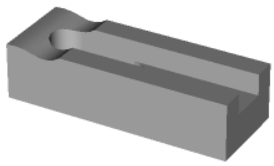
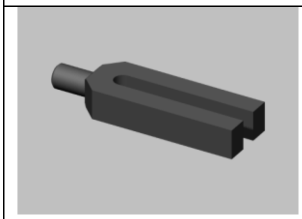
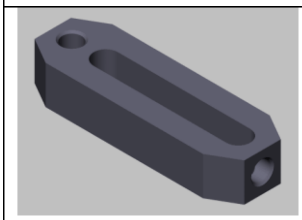
After the analysis of the implemented methodology the following problems were formulated:

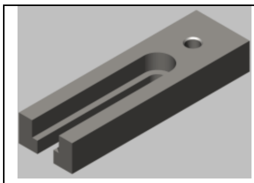
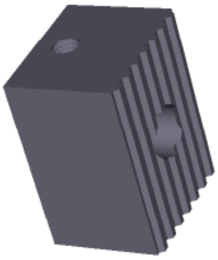
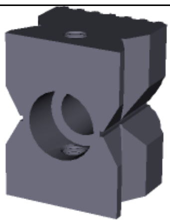
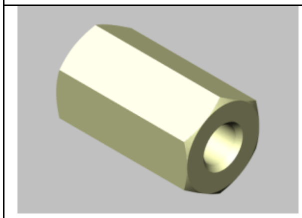
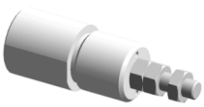
- A problem connected with Opitz Code (part number 2) when groove is not found while an upper curved machining is identified for non-rotational parts. For this case Opitz Code System gives only classification for groove or upper machining, not for both features at the same time. That is why the algorithm must select one of them ignoring the other one.
- For non-rotational parts there are 3 Opitz code groups of plane surface machining: one plane surface, stepped plane surface, stepped surface vertically inclined and/or opposed (4^{th} digit of Opitz code = 2, 3 and 4 respectively). These groups differ in the methods of machining, having in the result the same part shape. It means that for features of these 3 groups the algorithm has only shape geometry, not machining method; so it is impossible to relate a detail to the strict class, and by default there is a relation to the class of 4^{th} digit Opitz code = 2.
- There are problems with cylindrical surfaces counting for rotational parts. This is a programmatic error, not methodological.
- Rare problems with stepped bores and grooves identification are found.

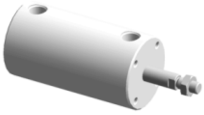
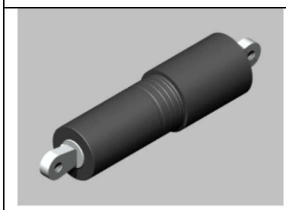
Along with the identified problems which were evaluated as minor, the basic implemented algorithms and the developed functionality showed good outcome results with a proved ability to recognize an entire set of product features for non-rotational long, flat, cubic and rotational parts having a proper accordance to Opitz Code Classification System.

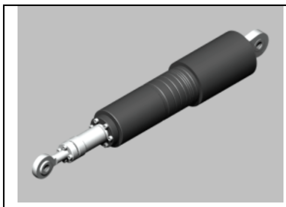
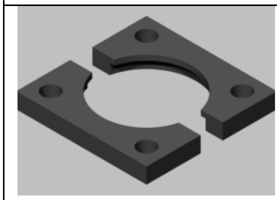
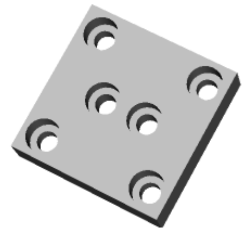
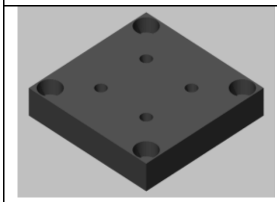
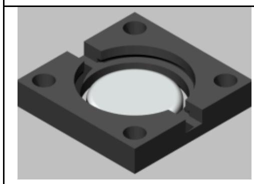
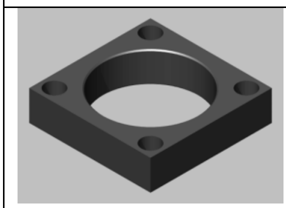
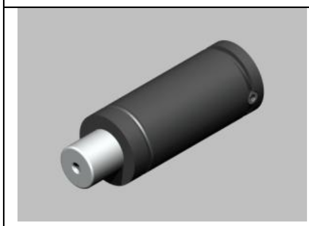
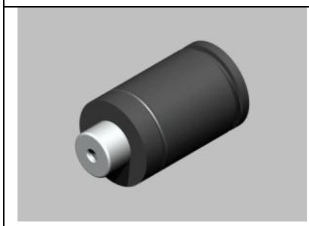
Image	Manufacturer	Result	Comment
 1.	EMILE MAURIN www.emile-maurin.fr	rotational shape three external cylinders no external machining no inner shape is found auxiliary holes found: 1 -----done: 24001	Ok
 2.	norelem www.norelem.fr	non rotational long front plane, back plane: uniform (equal) cross sections machining: curved surface one principal bore	Groove is not printed out, because upper curved machining is firstly identified.

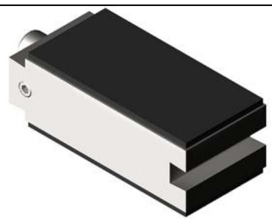
		-----done: 72170	
 3.	norelem www.norelem.fr	non rotational flat bottom: rectangle machining: curved surface one principal bore -----done: 60170	Ok
 4.	EMILE MAURIN www.emile-maurin.fr	rotational shape two external cylinders no external machining no inner shape is found auxiliary holes found: 1 -----done: 21001	Actually there are 3 cylinders: one external, one as auxiliary hole, one as inner hole. So 1 external cylinder should be, not 2.
 5.	norelem www.norelem.fr	non rotational flat machining: external groove is found inner shape: no principal bores -----done: 69050	Ok
 6.	Béné inox www.bene-inox.com	rotational shape two external cylinders no external machining inner bore is found -----done: 21100	Should be 1 external shape, not 2
 7.	ENOMAX GROUPE SUPRATEC www.enomax.fr	rotational shape two external cylinders no external machining no inner shape is found -----done: 21000	Should be 1 external cylinder, not 2
 8.	norelem www.norelem.fr	-----start rotational shape two external cylinders no external machining no inner shape is found -----done: 21000	Should be 1 external cylinder, not 2
 9	norelem www.norelem.fr	rotational shape two external cylinders no external machining no inner shape is found -----done: 21000	Should be 1 external cylinder, not 2
 10.	RABOURDIN INDUSTRIE www.rabourdin.fr	rotational shape two external cylinders no external machining no inner shape is found -----done: 21000	Ok
 11.	EMILE MAURIN http://www.emile-maurin.fr/	rotational shape three external cylinders no external machining no inner shape is found auxiliary holes found: 1 -----done: 24001	Should be 2 external cylinders, not 3

 12.	Béné inox www.bene-inox.com	rotational shape one external cylinder no external machining inner bore is found -----done: 10100	External machining should be found
 13.	EMILE MAURIN www.emile-maurin.fr	rotational shape two external cylinders no external machining inner bore is found -----done: 11100	External machining should be found, 1 external cylinder should be
 14.	EMILE MAURIN http://www.emile-maurin.fr/	rotational shape one external cylinder no external machining inner bore is found -----done: 10100	External machining should be found
 15.	GANTER GRIFF www.ganter-griff.de	non rotational long front plane, back plane: uniform (equal) cross sections machining: curved surface several principal bores, parallel -----done: 72570	Ok
 16.	norelem www.norelem.fr	non rotational long front plane, back plane: uniform (equal) cross sections machining: curved surface inner shape: no principal bores -----done: 72070	Groove not found
 17.	norelem www.norelem.fr	non rotational flat bottom: rectangle machining: curved surface inner shape: no principal bores -----done: 60070	Groove is not printed out, because upper curved machining is firstly identified.
 18.	norelem www.norelem.fr	non rotational long front plane, back plane: uniform (equal) cross sections machining: curved surface inner shape: no principal bores -----done: 72070	Ok
 19.	ENOMAX GROUPE SUPRATEC www.enomax.fr	non rotational long front plane, back plane: uniform (equal) cross sections machining: curved surface one principal bore -----done: 72170	No machining, but bore chamfers

 20.	ENOMAX GROUPE SUPRATEC www.enomax.fr	non rotational long front plane, back plane: varying cross sections machining: stepped 2 one principal bore -----done: 75120	Ok
 21.	HALDER NORM+TECHNIK http://www.halder.de/	non rotational cubic bottom: other shape machining: curved surface one principal bore auxiliary holes found: 1 -----done: 85171	No machining
 22.	HALDER NORM+TECHNIK http://www.halder.de/	non rotational cubic bottom: other shape machining: curved surface one principal bore -----done: 85170	No machining
 23.	HALDER NORM+TECHNIK http://www.halder.de/	non rotational cubic bottom: other shape machining: curved surface one principal bore auxiliary holes found: 1 -----done: 85171	No machining
 24.	MICHAUD CHAILL www.michaud-chailly.fr	non rotational cubic -----done: 8-1-1-10	External and internal shape not recognized
 25.	CCB www.ccb.fr	rotational shape one external cylinder no external machining inner bore is found -----done: 10100	No external machining
 26.	mayr® www.mayr.de	non rotational long -----done: 7-1000	Rotational
	FUJIKURA COMPOSITE www.fujikura-control.com	rotational shape more than 3 external cylinders no external machining no inner shape is found -----done: 24000	Ok

27.			
 28.	www.fujikura-control.com	rotational shape more than 3 external cylinders no external machining no inner shape is found auxiliary holes found: 1 -----done: 14001	Ok
 29	www.smac.fr	rotational shape more than 3 external cylinders no external machining no inner shape is found auxiliary holes found: 1 -----done: 10001	Non rotational. Input file format is wrong
 30.	www.norelem.fr	rotational shape three external cylinders no external machining no inner shape is found -----done: 24000	Ok
 31.	www.norelem.fr	rotational shape more than 3 external cylinders no external machining no inner shape is found -----done: 14000	Ok
 32.	www.norelem.fr	non rotational long front plane, back plane: varying cross sections machining: curved surface inner shape: no principal bores -----done: 75070	Bores not found
 33.	www.norelem.fr	rotational shape one external cylinder no external machining inner bore is found -----done: 20100	More than 1 external cylinder, no inner bore
 34.	www.quiri.com	rotational shape more than 3 external cylinders no external machining no inner shape is found -----done: 24000	Ok

 35.	www.quiri.com	rotational shape more than 3 external cylinders no external machining no inner shape is found auxiliary holes found: 1 -----done: 14001	Ok
 36.	www.mdl-rodis.com	non rotational flat bottom: CircularAndOrtogonal machining: stepped 2 two principal bores, parallel -----done: 63420	Ok
 37.	www.mdl-rodis.com	non rotational flat bottom: rectangle machining: stepped > 2 inner shape: no principal bores -----done: 60030	Stepped bores not found
 38.	www.mdl-rodis.com	non rotational flat bottom: rectangle machining: stepped > 2 several principal bores, parallel -----done: 60530	Ok. Probably stepped bores.
 39.	www.mdl-rodis.com	non rotational cubic bottom: other shape machining: has chambers inner shape: no principal bores -----done: 85010	Ok. Probably stepped bores.
 40.	www.mdl-rodis.com	non rotational cubic bottom: rectangle machining: stepped 2 one principal bore -----done: 80120	Ok. Probably stepped bores.
 41.	www.enomax.fr	rotational shape one external cylinder no external machining no inner shape is found -----done: 10000	Two cylinders should be
 42.	www.enomax.fr	rotational shape three external cylinders no external machining no inner shape is found -----done: 14000	Ok?

 43.	norelem www.norelem.fr	non rotational cubic -----done: 8-1-1-10	No all features recognized
 44.	norelem www.norelem.fr	rotational shape two external cylinders no external machining no inner shape is found auxiliary holes found: 1 -----done: 21001	More than 2 external cylinders
 45.	norelem www.norelem.fr	non rotational cubic bottom: other shape machining: stepped > 2 inner shape: no principal bores auxiliary holes found: 2 -----done: 85031	Inner bore not found
 46.	norelem www.norelem.fr	non rotational cubic bottom: other shape machining: curved surface inner shape: no principal bores -----done: 85070	More than 2 external cylinders
 47.	EMILE MAURIN www.emile-maurin.fr	rotational shape three external cylinders no external machining no inner shape is found -----done: 14000	Auxiliary hole not found
 48.	www.tea.net.au	non rotational flat bottom: rectangle inner shape: no principal bores -----done: 60000	Cubic should be
 49.	norelem www.norelem.fr	non rotational long front plane, back plane: uniform (equal) cross sections (rectangular) machining: curved surface inner shape: no principal bores -----done: 70070	Groove not found

 50.	norelem www.norelem.fr	non rotational flat machining: no machining several principal bores, parallel -----done: 64500	Ok

Table 1. Developed methodology outcome analysis

CONCLUSION

The main objective of the current research was to develop a method that enables product feature recognition and extraction from some standardized part shape format in conformity with Opitz Code Classification System. This novel technique should facilitate engineers to reuse knowledge information in the field of 3D solid modeling with an optimization of new product design process. To achieve this goal a classification system based on Opitz Code for rotational and non-rotational parts was developed.

To find the most appropriate shape format, different types of modern product representations were concerned. Native CAD representation appeared as bulky and proprietary; their specifications are restricted that contradicts the ideas of effective information exchange. On the other hand lightweight representations always imply the data losses that are not suitable for tasks of this research: only precise geometrical data can be used. As a result the neutral format group was chosen as the most effective one.

Further the neutral format group was analyzed with figuring out two market leaders: IGES and STEP format. After a closer look IGES proved a few crucial drawbacks: it does not have a formal data model that causes ambiguities in some cases; the lack of any conformance requirements leaded to IGES file deviations among different vendors; it provides only drawing or 3D modeling data neglecting manufacturing view that would be of higher demand. Thus, STEP was selected for reasons of its strengths which overcome mentioned issues.

On other side, Group Technology as a classification method that implies feature recognition was evaluated and Opitz Code as a method of GT was implemented for this research. There are several advantages were obtained by its adopting:

- Design simplification
- Setting time, and Batch quantities abilities
- Materials handling simplification
- Higher production and inventory control
- Process planning ability
- Effective Supervision

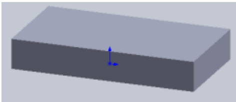
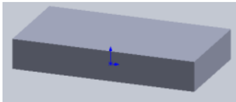
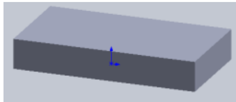
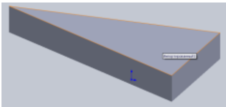
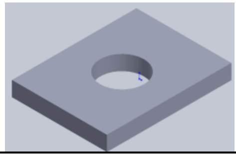
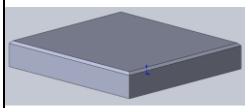
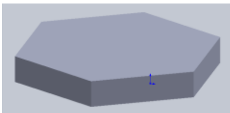
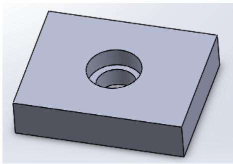
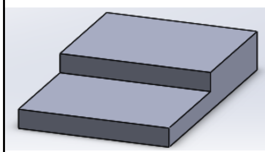
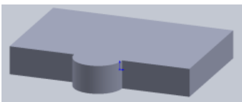
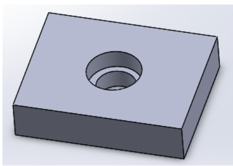
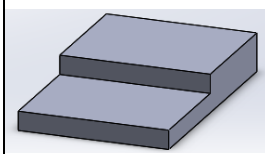
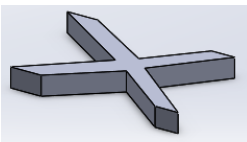
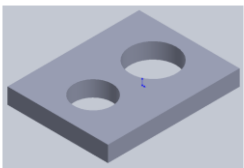
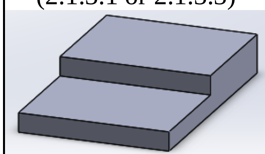
As disadvantage of Group Technology was identified the machine utilization that likely lower than with the traditional functional layout. As it was presented, this is more realistically regarded as offsetting of some of the benefit of lower investment in work-in-progress. Probably the biggest disadvantage is the effort required to changeover to a Group Technology method of working, perhaps combined with some risk, if one has not done it before, of not obtaining sufficient benefit to justify the effort.

The developed system was implemented by means of Java programming language; STEP representation format was used to reflect particular part shape geometry and topology. After a STEP file is loaded to the system, feature extraction process starts together with generation of Opitz Code signature. This process can also be named “classification” while having Opitz Code signature as a result implies a predefined group according to Group Technology. A Graphical User Interface has been also implemented to allow user to choose preferred STEP file and to see feature recognition progress with informative notifications and the outcoming Opitz signature.

The evaluation and testing proves proper functional abilities of the system with the correct implementation. Classification and comparison of 50 rotational and non-rotational parts taken from the sources of official manufacturers has resulted in a proper outcome.

During evaluation two groups of minor problems were identified: implementation errors and Opitz Code problems. The first group refers to the programming errors and includes: false calculation of cylindrical surfaces within rotational parts, some grooves and stepped bores neglecting. The later group includes errors caused by lack of manufacturing data: Opitz Classification has groupings by type of machining which is not provided by STEP shape representations. These problems can be overcome by means of programmed code refinement and inclusion of required manufacturing data about product to STEP presentation.

APPENDIX

DIGIT 2. MAIN FORM			DIGIT 3. Main bore and rotational machining		DIGIT 4. Machining of plane surfaces		DIGIT 5. Other holes, teeths and forming		
0	Plane/flat	Rectangular plane Rules: 1.1; 2.1.1.1 	0	No machining or bore(s) Rules: 1.1; 2.1.1.*; 2.1.2.1 	0	Without surface machining Rules: 1.1; 2.1.1.*; 2.2.1 	0		Without features
1		Right-angled or triangle Rules: 1.1; 2.1.1.2 	1	One principal bore Rules: 1.1; 2.1.1.*; 2.1.2.2 	1	Chamfers Rules: 1.1; 2.1.1.*; 2.2.2 	1	Without transformation / without gearing	One bore direction
2		Angularly Rules: 1.1; 2.1.1.3 	2	One principle bore stepped Rules: 1.1; 2.1.1.*; 2.1.2.2.1 	2	One plane surface Rules: 1.1; 2.1.1.*; (2.1.3.1 or 2.2.1 or 2.1.3.2) 	2		Several bore directions
3		Circular and rectangular Rules: 1.1; 2.1.1.4 	3	One principle bore stepped with machining elements Rules: 1.1; 2.1.1.*; 2.1.2.2.1 	3	Stepped plane surface Rules: 1.1; 2.1.1.*; (2.1.3.1 or 2.1.3.3) 	3	With hole	One bore direction
4		Other Rules: 1.1; 2.1.1.8 	4	Two main bores parallel Rules: 1.1; 2.1.1.*; 2.1.2.3 	4	Stepped surface vertically inclined and/or opposed Rules: 1.1; 2.1.1.*; (2.1.3.1 or 2.1.3.3) 	4		Several bore directions

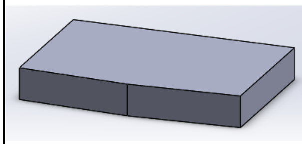
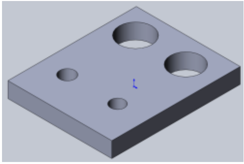
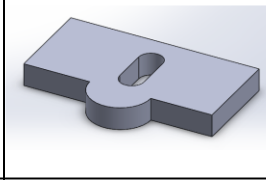
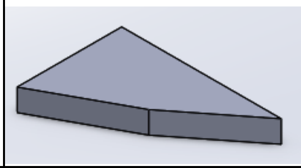
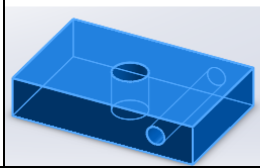
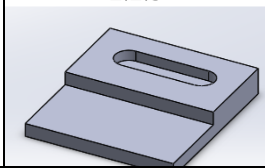
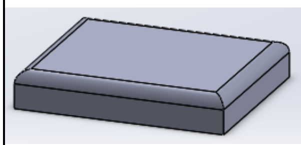
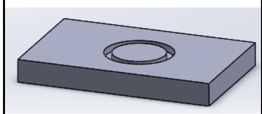
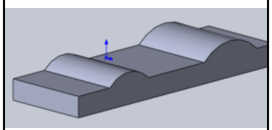
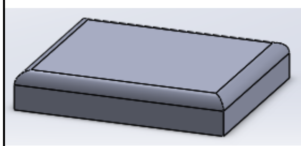
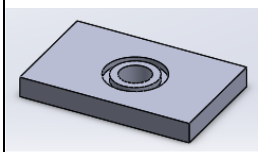
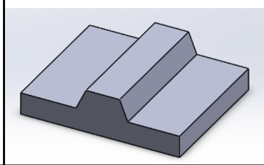
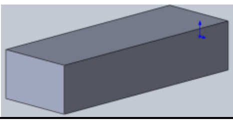
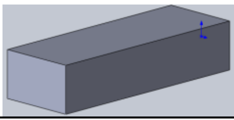
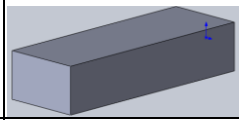
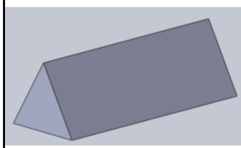
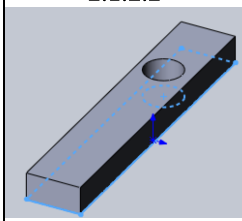
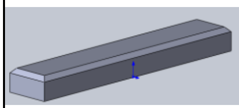
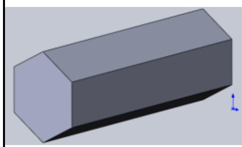
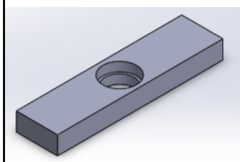
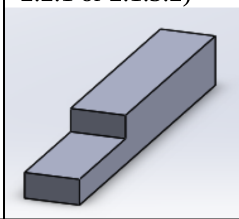
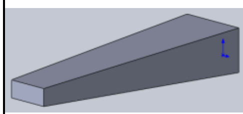
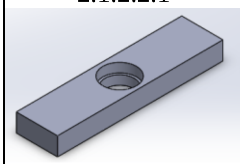
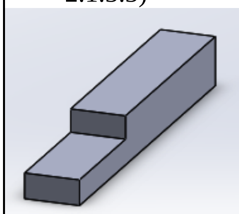
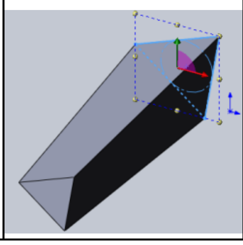
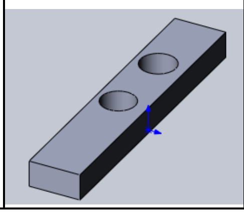
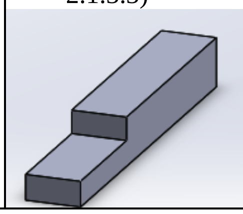
5	Flat part rectangular or right angled with small deviations Rules: 1.1; 2.1.1.6; 	5	More than two main bores, parallel Rules: 1.1; 2.1.1.*; 2.1.2.4 	5	Groove and/or slot Rules: 1.1; 2.1.1.*; 2.2.3 	5	Transformation /without gearing	Formed without drilling
6	Flat part round or any other shape with small deviations Rules: 1.1; 2.1.1.7; 	6	Many main bored perpendicular Rules: 1.1; 2.1.1.*; 2.1.4 	6	Groove and/or slot and 4 Rules: 1.1; 2.1.1.*; (2.1.3.2 or 2.1.3.3); 2.2.3 	6		Formed with drilling
7	Flat part with regularly arched form Rules: 1.1; 2.2.6; 	7	Ring groove machining surfaces Rules: 1.1; 2.1.1.*; 2.1.2.3.1 	7	Curved surface Rules: 1.1; 2.1.1.*; 2.2.4 	7	Gearing	
8	Flat part with irregularly arched form Rules: 1.1; 2.2.7; 	8	7 + main bore Rules: 1.1; 2.1.1.*; 2.1.2.4.1 	8	Guided surface Rules: 1.1; 2.1.1.*; 2.2.5 	8	Gearing with hole	
9	Other	9	Other	9	Other	9	Other	

Table 1. Non-rotational flat parts (1st Opitz code digit = 6) classification [53]

DIGIT 2. MAIN FORM			DIGIT 3. Main bore and rotational machining		DIGIT 4. Machining of plane surfaces		DIGIT 5. Other holes, teeths and forming		
Code	Category	Description	Code	Description	Code	Description	Code	Category	Description
0	Shape axis straight / Uniform cross-section	Rectangular Rules: 1.2; 3.1.1.1 	0	No machining or bore(s) Rules: 1.2; 3.1.*.*; 2.1.2.1 	0	Without surface machining Rules: 1.2; 3.1.*.*; 2.2.1 	0		Without features
1	Shape axis straight / Uniform cross-section	Right angle or triangular Rules: 1.2; 3.1.1.2 	1	One principal bore Rules: 1.2; 3.1.*.*; 2.1.2.2 	1	Chamfers Rules: 1.2; 3.1.*.*; 2.2.2 	1	Without transformation / without gearing	One bore direction
2	Shape axis straight / Uniform cross-section	Any cross-section other than 0 and 1 Rules: 1.2; 3.1.1.3 	2	One principle bore stepped Rules: 1.2; 3.1.*.*; 2.1.2.2.1 	2	One plane surface Rules: 1.2; 3.1.*.*; (2.1.3.1 or 2.2.1 or 2.1.3.2) 	2	Without transformation / without gearing	Several bore directions
3	Shape axis straight / Varying cross-section	Rectangular Rules: 1.2; 3.1.2.1 	3	One principle bore stepped with machining elements Rules: 1.2; 3.1.*.*; 2.1.2.2.1 	3	Stepped plane surface Rules: 1.2; 3.1.*.*; (2.1.3.1 or 2.1.3.3) 	3	Without transformation / without gearing / With hole	One bore direction
4	Shape axis straight / Varying cross-section	Right angle or triangular Rules: 1.2; 3.1.2.2 	4	Two main bores parallel Rules: 1.2; 3.1.*.*; 2.1.2.3 	4	Stepped surface vertically inclined and/or opposed Rules: 1.2; 3.1.*.*; (2.1.3.1 or 2.1.3.3) 	4	Without transformation / without gearing / With hole	Several bore directions

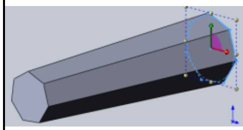
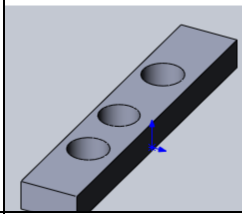
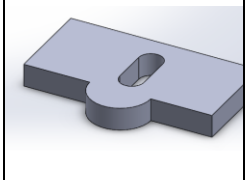
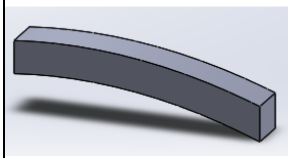
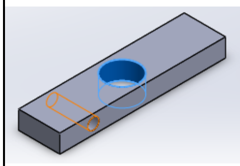
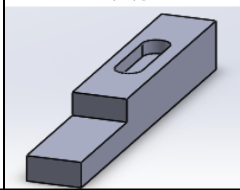
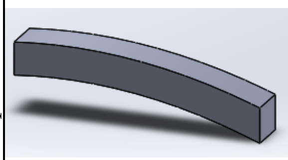
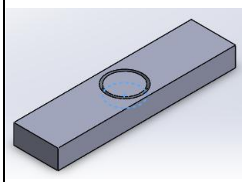
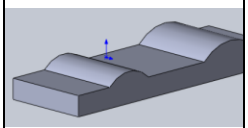
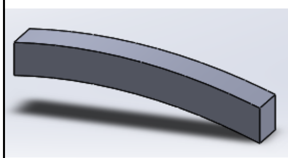
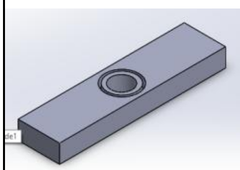
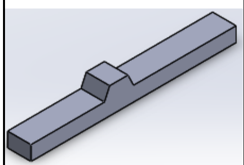
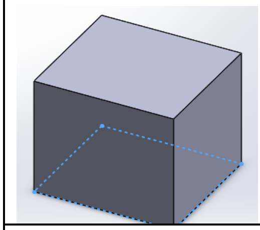
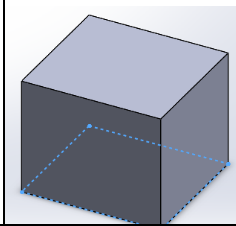
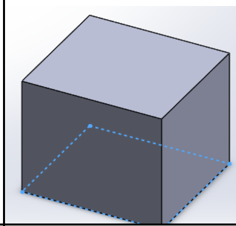
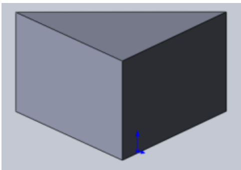
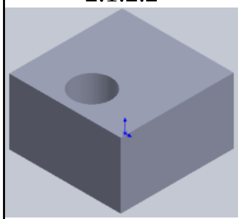
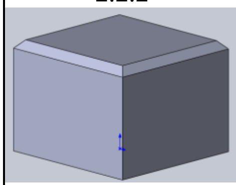
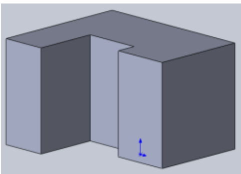
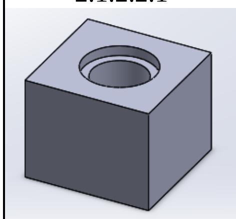
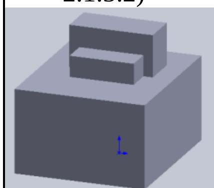
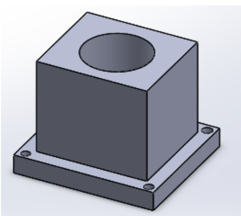
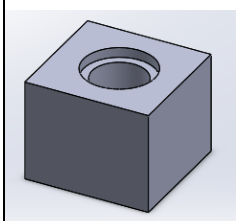
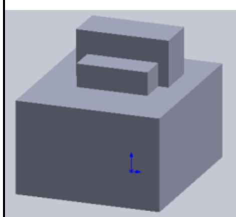
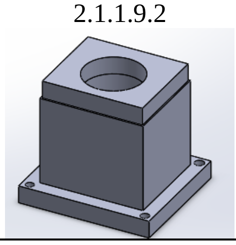
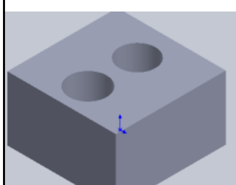
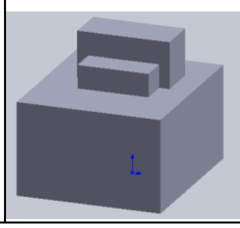
5		Any cross-section other than 3 and 4 Rules: 1.2; 3.1.2.3 	5	More than two main bores, parallel Rules: 1.2; 3.1.*.*; 2.1.2.4 	5	Groove and/or slot Rules: 1.2; 3.1.*.*; 2.2.3 	5	Transformation /without gearing	Formed without drilling
6		Rectangular, angular or other cross-section Rules: 1.2; 3.2.1 	6	Many main bored perpendicular Rules: 1.2; 3.1.*.*; 2.1.4 	6	Groove and/or slot and 4 Rules: 1.2; 3.1.*.*; (2.1.3.2 or 2.1.3.3); 2.2.3 	6		Formed with drilling
7	Shape axis curved (bent)	Formed component Rules: 1.2; 3.2.2 	7	Ring groove machining surfaces Rules: 1.2; 3.1.*.*; 2.1.2.3.1 	7	Curved surface Rules: 1.2; 3.1.*.*; 2.2.4 	7	Gearing	
8		Formed component with deviations in the main axis Rules: 1.2; 3.2.3 	8	7 + main bore Rules: 1.2; 3.1.*.*; 2.1.2.4.1 	8	Guided surface Rules: 1.2; 3.1.*.*; 2.2.5 	8	Gearing with hole	
9		others	9	Other	9	Other	9	Other	

Table 2. Non-rotational long parts (1st Opitz code digit = 7) classification [53]

DIGIT 2. MAIN FORM			DIGIT 3. Main bore and rotational machining		DIGIT 4. Machining of plane surfaces		DIGIT 5. Other holes, teeths and forming			
0	Block and block-like components	Rectangular prism Rules: 1.3; 2.1.1.1 	0	No machining or bore(s) Rules: 1.3; 2.1.1.*; 2.1.2.1 	0	Without surface machining Rules: 1.3; 2.1.1.*; 2.2.1 	0	Without features		
1		Right angle or triangular Rules: 1.3; 2.1.1.2 	1	One principal bore Rules: 1.3; 2.1.1.*; 2.1.2.2 	1	Chamfers Rules: 1.3; 2.1.1.*; 2.2.2 	1	Without transformation /without gearing		One bore direction
2		Compounded of rectangular prisms Rules: 1.3; 2.1.1.5 	2	One principle bore stepped Rules: 1.3; 2.1.1.*; 2.1.2.2.1 	2	One plane surface Rules: 1.3; 2.1.1.*; (2.1.3.1 or 2.2.1 or 2.1.3.2) 	2			Several bore directions
3		Components with a mounting or locating surface and principal bore Rules: 1.3; 2.1.1.5 2.1.1.9.1 	3	One principle bore stepped with shape elements Rules: 1.3; 2.1.1.*; 2.1.2.2.1 	3	Stepped plane surface Rules: 1.3; 2.1.1.*; (2.1.3.1 or 2.1.3.3) 	3		With hole	One bore direction
4		Components with a mounting or locating surface, principal bore with dividing surface Rules: 1.3; 2.1.1.5 2.1.1.9.2 	4	Two main bores parallel Rules: 1.3; 2.1.1.*; 2.1.2.3 	4	Stepped surface vertically inclined and/or opposed Rules: 1.3; 2.1.1.*; (2.1.3.1 or 2.1.3.3) 	4			Several bore directions

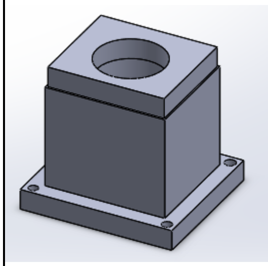
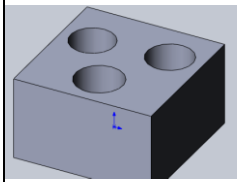
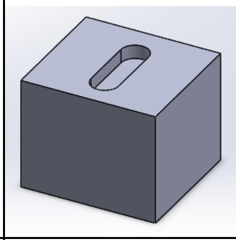
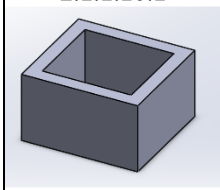
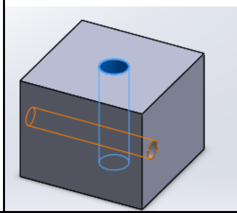
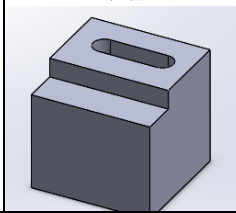
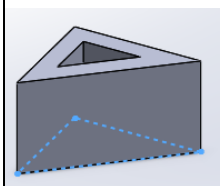
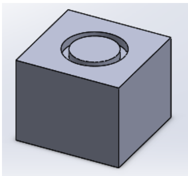
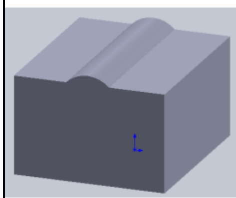
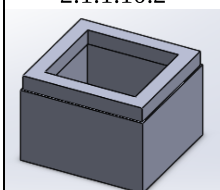
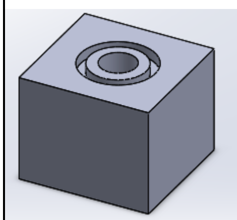
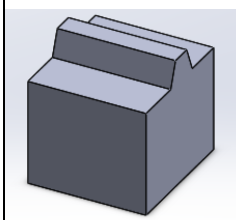
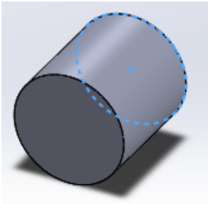
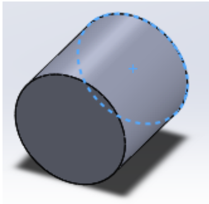
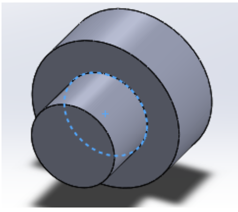
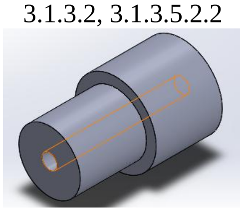
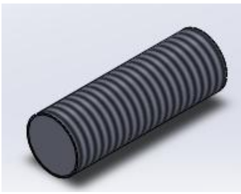
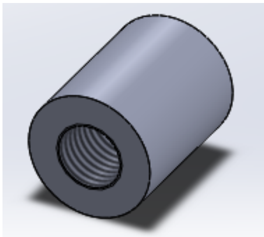
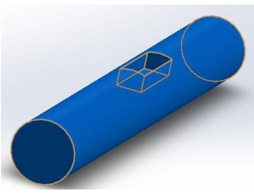
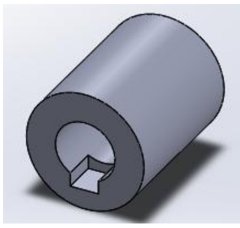
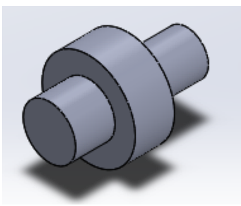
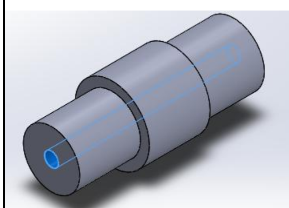
5			Components other than 0 to 4 Rules: 1.3; 2.1.1.8 	5	More than two main bores, parallel Rules: 1.3; 2.1.1.*; 2.1.2.4 	5	Groove and/or slot Rules: 1.3; 2.1.1.*; 2.2.3 	5	Transformation /without gearing	Formed without drilling
6	Box and box-like components	Not split	Approximate or compounded of rectangular prisms Rules: 1.3; 2.1.1.10.1 	6	Many main bored perpendicular Rules: 1.3; 2.1.1.*; 2.1.4 	6	Groove and/or slot and 4 Rules: 1.3; 2.1.1.*; (2.1.3.2 or 2.1.3.3); 2.2.3 	6		Formed with drilling
7			Components other than 6 Rules: 1.3; 2.1.1.10.1 	7	Ring machining surfaces Rules: 1.3; 2.1.1.*; 2.1.2.3.1 	7	Curved surface Rules: 1.3; 2.1.1.*; 2.2.4 	7	Gearing	
8		Split	Approximate or compounded of rectangular prisms Rules: 1.3; 2.1.1.10.2 	8	7 + main bore Rules: 1.3; 2.1.1.*; 2.1.2.4.1 	8	Guided surface Rules: 1.3; 2.1.1.*; 2.2.5 	8	Gearing with hole	
9			Components other than 8	9	Other	9	Other	9	Other	

Table 3. Non-rotational cubic parts (1st Opitz code digit = 8) classification [53]

DIGIT 2. External shape, external shape elements			DIGIT 3. Internal shape, internal shape elements			DIGIT 4. Plane surface machining		DIGIT 5. Auxiliary holes and gear teeth		
0		Smooth, no shape elements Rules: 1.4-1.6, 3.1.3.1, 3.1.3.5.1.1 	0		Smooth, no shape elements Rules: 1.4-1.6, 3.1.3.1, 3.1.3.5.2.1 	0	Without surface machining	0		No auxiliary hole(s)
1		No shape elements Rules: 1.4-1.6, 3.1.3.2, 3.1.3.5.1.1 	1		No shape elements Rules: 1.4-1.6, 3.1.3.1 or 3.1.3.2, 3.1.3.5.2.2 	1	External plane surface and / or surface curved in one direction	1		Axial hole(s) not related by a drilling pattern
2	Stepped to one end	With screwthread Rules: 1.4-1.6, (3.1.3.1 or 3.1.3.2), 3.1.3.5.1.3 	2	Smooth or stepped to one end	With screwthread Rules: 1.4-1.6, (3.1.3.1 or 3.1.3.2), 3.1.3.5.2.2.1 	2	External plane surfaces related to one another by graduation around a circle	2	No gear teeth	Axial hole(s) related by a drilling pattern
3	Or smooth	With functional groove Rules: 1.4-1.6, (3.1.3.1 or 3.1.3.2), 3.1.3.5.1.2 	3		With functional groove Rules: 1.4-1.6, (3.1.3.1 or 3.1.3.2), 3.1.3.5.2.2.2 	3	External groove and / or slot	3		Radial hole(s) not related by a drilling pattern
4		No shape elements Rules: 1.4-1.6, 3.1.3.3, 3.1.3.5.1.1 	4		No shape elements Rules: 1.4-1.6, 3.1.3.3, 3.1.3.5.2.3 	4	External spline and / or polygon	4		Holes axial and / or radial and / or in other directions, not related

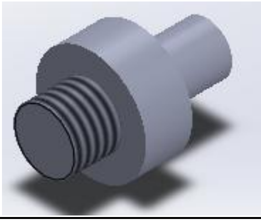
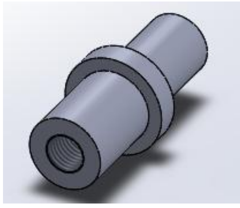
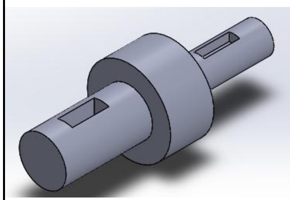
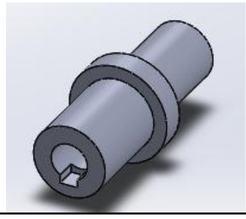
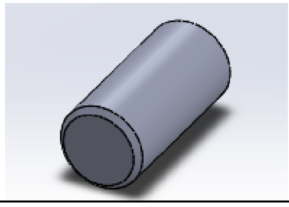
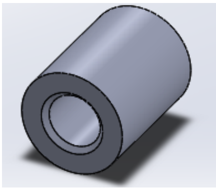
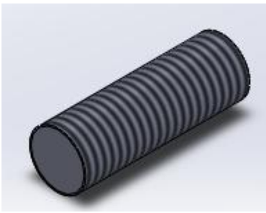
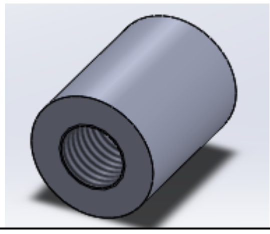
5	Stepped to both ends	With screwthread Rules: 1.4-1.6, 3.1.3.3, 3.1.3.5.1.3 	5	Stepped to both ends	With screwthread Rules: 1.4-1.6, 3.1.3.3, 3.1.3.5.2.3.1 	5	External plane surface and / or slot and / or groove, spline	5		Holes axial and / or radial and / or in other directions, related by drilling pattern
6		With functional groove Rules: 1.4-1.6, 3.1.3.3, 3.1.3.5.1.2 	6		With functional groove Rules: 1.4-1.6, 3.1.3.3, 3.1.3.5.2.3.2 	6	Internal plane surface and / or groove	6	With gear teeth	Spur gear teeth
7	Functional taper Rules: 1.4-1.6, 3.1.3.1, 3.1.3.5.1.4 		7	Functional taper Rules: 1.4-1.6, 3.1.3.1, 3.1.3.5.2.4 		7	Internal spline and / or polygon	7		Bevel gear teeth
8	Operating thread Rules: 1.4-1.6, 3.1.3.1, 3.1.3.5.1.3 		8	Operating thread Rules: 1.4-1.6, 3.1.3.1, 3.1.3.5.2.2.1 		8	External and Internal splines and / or slot and / or groove	8		Other gear teeth
9	Others (> 10 functional diameters) Rules: 1.4-1.6, 3.1.34		9	Others (> 10 functional diameters)		9	Others	9		others

Table 4. Rotational parts (1^{st} Opitz code digit = 0..2) classification [53]

REFERENCES

1. The Initial Graphics Exchange Specification (IGES) Version 6.0 // Baseline 1/98, November 30, 2001, pp. 2.
2. V. Quintana, Rivest L., Re-engineering the Engineering Change Management process for a drawing-less environment // Computers in Industry, 2012, Vol. 63(1), pp. 79–90.
3. J. Matthews, McMahon C., Product representation in lightweight formats for product lifecycle management (PLM) // 4th International Conference on Digital Enterprise Technology, Bath, UK, 2007.
4. STEP Application Handbook ISO 10303 version 3 // SCRA, North Charleston, SC, 30 June 2006.
5. Bhandarkar Mangesh P., Rakesh Nagi, STEP-based feature extraction from STEP geometry for Agile Manufacturing // Computers in Industry, 2000, Vol. 41, pp. 3–24.
6. Bhandarkar Mangesh P., Blair Downie, Migrating from IGES to STEP: one to one translation of IGES drawing to STEP drafting data // Computers in Industry, 2000, Vol. 41, pp. 261–277.
7. Bhandarkar Mangesh P., Rakesh Nagi, STEP product information models in Agile manufacturing // Department of Industrial Engineering, 342 Bell Hall, State University of New York at Buffalo, NY 14260, 1999.
8. O Owodunni, Hinduja S, Evaluation of existing and new feature recognition algorithms: Part 1: Theory and Implementation // Proceedings of the Institution of Mechanical Engineers, Part B: Journal of Engineering Manufacture, 2002, pp. 216.
9. Babic Bojan, Nenad Nesic, A review of automated feature recognition with rule-based pattern recognition // Computers in Industry, 2008(59), pp. 321-337.
10. Han J.H., M. Pratt, Manufacturing feature recognition from solid models: a status report // IEEE Transactions on Robotics and Automation, 2000(16), pp. 782–796.
11. Jain P.K., S. Kumar, Automatic feature extraction in PRIZCAPP // International Journal of Computer Integrated Manufacturing, 1998(11), pp. 500-512.
12. Henderson M.R., D.C. Anderson, Computer recognition and extraction of form features: a CAD/CAPP link // Computers in Industry, 1984(5), pp. 329–339.
13. Bouzakis H.K.-D., G. Andreadis, A feature-based algorithm for computer aided process planning for prismatic parts // International Journal of Production Engineering and Computers, 2000, pp. 17–22.
14. Sadaiah M., D.R. Yadav, A generative CAPP system for prismatic components // International Journal of Advanced Manufacturing Technology, (20), pp. 709–719.
15. Venuvinod P.K., S.Y. Wong, A graph-based expert system approach to geometric feature recognition // Journal of Intelligent Manufacturing, 1994(6), pp. 401–413.
16. Yuen C.F., P.K. Venuvinod, Geometric feature recognition: coping with the complexity and infinite variety of features // International Journal of Computer Integrated Manufacturing, 1999(12), pp. 439–452.
17. Marefat M., R.L. Kashyap, Geometric reasoning for recognition of three-dimensional object features // IEEE Transactions on Pattern Analysis and Machine Intelligence, 1990(12), pp. 949–965.
18. Ji Q., M.M. Marefat, A Dempster-Shafer approach for recognizing machine features from CAD models // Pattern recognition, 2003(36), pp. 1355–1368.
19. Ji Q., M.M. Marefat, Bayesian approach for extracting and identifying features // Computer-Aided Design, 1995(27), pp. 435–454.
20. Qamhiyah A.Z., R.D. Venter, Geometric reasoning for the extraction of form features // Computer-Aided Design, 1996(28), pp. 887–903.
21. Huang Z., D. Yip-Hoi, High-level feature recognition using feature relationship graphs // Computer-Aided Design, 2002(34), pp. 561–582.

22. Stefano P. Di, F. Bianconi, An approach for feature semantics recognition in geometric models // Computer-Aided Design, 2004(36), pp. 993–1009.
23. Kim Y.S.,Recognition of form features using convex decomposition // Computer-Aided Design, 1992(24), pp. 461–476.
24. Miao H.K., N. Sridharan, CAD-CAM integration using machining features // International Journal of Computer Integrated Manufacturing, 2002(15), pp. 296–318.
25. Dong J., S. Vijayan, Features extraction with the consideration of manufacturing processes // International Journal of Production Research, 1997(35), pp. 2135–2155.
26. Nagaraj H.S., B. Gurumoorthy, Automatic extraction of machining primitives with respect to preformed stock for process planning // Journal of Manufacturing Systems, 2001(20), pp. 210–222.
27. Sakurai H., C. Chin, Definition and recognition of volume features for process planning // Advances in Feature Based Manufacturing, Elsevier, 1994, pp. 65–80.
28. Sakurai H., P. Dave, Volume decomposition and feature recognition. Part 2. Curved objects // Computer-Aided Design, 1996(28), pp. 519–537.
29. Woo Y.,Fast cell-based decomposition and applications to solid modeling // Computer-Aided Design, 2003(35), pp. 969–977.
30. Tseng Y.J., S.B. Joshi, Recognition of interacting rotational and prismatic machining features from 3D mill-turn parts // International Journal of Production Research, 1998, Vol. 36, pp. 3147–3165.
31. Bronsvoort W.F., A. Noort, Multiple-view feature modeling for integral product development // Computer-Aided Design, 2004(36), pp. 929–946.
32. Han J.H., I. Han, Manufacturing feature recognition toward integration with process planning // EEE Transactions Systems, Man and Cybernetics. Part B. Cybernetics, 2001(21), pp. 373–380.
33. Meeran S., J.M. Taib, Recognizing features from engineering drawings without using hidden lines: a framework to link feature recognition and inspection systems // International Journal of Production Research, 2003, Vol. 41, pp. 465–495.
34. McCormack A.D., R.N. Ibrahim, Process planning using adjacency-based features // International Journal of Advanced Manufacturing Technology, 2002(20), pp. 817–823.
35. Gao S., J.J. Shah, Automatic recognition of interacting machining features based on minimal condition subgraph // Computer-Aided Design, 1998(30), pp. 727–739.
36. Little G., R. Tuttle, The Heriot-Watt FeatureFinder: CIE97 results // Computer-Aided Design, 1998(30), pp. 991–996.
37. Ye X.G., J.Y.H. Fuh, A hybrid method for recognition of undercut features from moulded parts // Computer-Aided Design, 2001(33), pp. 1023–1034.
38. Rahmani K., B. Arezoo, Boundary analysis and geometric completion for recognition of interacting machining features // Computer-Aided Design, 2006(38), pp. 845–856.
39. Zhang C., Y.H. Chen, A hybrid method for recognizing feature interactions // Integrated Manufacturing Systems, 1998(9), pp. 120–128.
40. Sundararajan V., P.K. Wright, Volumetric feature recognition for machining components with freeform surfaces // Computer-Aided Design, 2004(36), pp. 11–25.
41. Gilman C.,The use of the PDES form feature information model as the primary representation for a form feature-based design environment // Master's Thesis, Rensselaer Polytechnic Institute, Troy, NY, 1990.
42. Shah J.J., A. Mathew, Experimental investigation of the STEP form-feature information model // Computer Aided Design, 1991, Vol. 23(4), pp. 282–296.
43. Ssemakula M.E., A. Satsangi, Application of PDES to CAD/CAPP integration // Computers and Industrial Engineering, 1990, Vol. 18(4), pp. 435–444.
44. Velden Christian Van der, Hao-Lan Zhang, Extracting engineering features from B-Rep geometric models // ICAS 2010, 27th international congress of the aeronautical sciences, 2010.

45. El-Mehalawi M., R. Miller, A database system of mechanical components based on geometric and topological similarity. Part I: representation // Comput-Aided Des, 2003, Vol. 35, pp. 83-94.
46. Fu M.W., S.K. Ong, An approach to identify design and manufacturing features from a data exchanged part model // Computer-Aided Design, 2003, Vol. 35, pp. 979-993.
47. Bespalov Dmitriy, William C. Regli, Local Feature Extraction and Matching Partial Objects // Department of Computer Science, College of Engineering Drexel University, 2006.
48. Arn E.A.,Group Technology: An integrated planning and implementationconcept for small and medium batch production // Springer-Verlag, Berlin, 1976.
49. Mitrofanov S. P.,The Scientific Principles of Group Technology // National Lending Library for Science and Technology, Boston, 1966.
50. Hartl Richard F., Margaretha Preusser, Layout and Design // Produktion und Logistik 2009.
51. Sacchetti Joseph, Soundar Kumara, A group technology based classification and coding system for reinforced conrete structures // Computer integrated construction, 1989, Vol. 13.
52. Narayan K. Lalit, K. Mallikarjuna Rao, // Computer Aided Design and Manufacturing, 2008.
53. Opitz Herwart, A Classification System to Describe Workpieces // 1970.
54. Iyer Shekhar, Rakesh Nagi, Automated retrieval and ranking of similar parts in Agile Manufacturing // IEE Transactions, 1997.
55. www.scribd.com,The CODE system for Computer Aided Design and Manufacturing //.
56. Haileeyesus Yosef,Group technology based classification and coding system // Addis Ababa University, 2002.

Erklärung

Hiermit versichere ich, diese Arbeit selbstständig verfasst und nur die angegebenen Quellen benutzt zu haben.

Unterschrift:

Stuttgart, den 13.05.2014