



Interactive GT Code Management for Mechanical Part Similarity Search and Cost Prediction

Alfredo Liverani¹ and Alessandro Ceruti²

¹University of Bologna, alfredo.liverani@unibo.it

²University of Bologna, alessandro.ceruti@unibo.it

ABSTRACT

This paper reports the results of a PDM and CAD plug-in implementation for semi-automatic and real-time similar component search in mechanical field. The approach exploits a string based component description similar to the well-known methodology, called Group Technology (GT), in order to check interactively feature similarity over a PDM database. The GT code contains component geometric data and manufacturing information. The software developed is suitable for encoding 2D and 3D parts. A guided GUI returning the GT code has been implemented for 2D drafts. For 3D parts, instead, the encoding procedure is completely integrated in the modelling CAD interface and the code is calculated incrementally feature by feature.

So the part similarity assessment is interactive: the designer may visualize similar parts stored in the PDM and decide whether changing singular feature or using a retrieved (similar) part. Several case studies described in the paper demonstrate GUI usage, search algorithm and results. With PDM correctly configured, results are very good since the GT coding, the part retrieval and the quoting are really interactive.

Keywords: similarity assessment, cost prediction, GT code, interactivity.

DOI: 10.3722/cadaps.2010.1-15

1 INTRODUCTION

1.1 Motivation

Whereas well known techniques to reduce significantly the number of component in large mechanical assemblies have been introduced in big enterprises at least twenty years ago, in Small and Medium Industries (SMI) problems like “time to market” reduction and “lean production” are still partially solved. Companies active in the mechanical automation field are working nowadays with small number productions with very high cost per item. Furthermore, the lack of a correct usage or implementation of PDM (Product Data Management)/PLM (Product Lifecycle Management) tools sometimes prevents significant savings in many steps of the product chain, like design, drawings, bill of materials, raw material logistic, production layout, quality and marketing.

The most important benefits of a PDM/PLM system combined with a CAD 2D or, better, 3D software are clear: project/product management (parts retrieval, confrontation, time, costs, human resources...) and customer management. In other terms, these software are company Knowledge Management (KM)

repositories. But one of the most important activity for project development is cost estimation and prediction and in SMI it is still a human activity and not an automated assessment. Very experienced people are required to evaluate costs on the basis of objective and subjective parameters. Objective parameters are geometry, surface finishing, needed machine processing, company's equipments and experience; subjective parameters are preparation time before part machining, the presence of experienced operators in some frequent machining sequences, the availability in short time of external companies competent in specific operations (outsourcing). Of course many of these parameters are statistically significant.

But one of the most performing way to get a feasible component quoting, is still the search and retrieve of similar parts in the PDM/PLM database. Furthermore modern company PDM are fully integrated with production costs and customer/supplier management. Thus part cost can be retrieved by database with part ID. These are true costs because parts have been really manufactured and include also cost variation depending on number of items produced. Especially in very small numbers, costs much more influenced by preparation, NC code writing or adaptation, availability of tools..., than machining time. So in this work the part similarity has to be intended in deductive mode instead of inductive mode, like most of engineering projects. In fact the equation similar parts = similar costs is true only neglecting the strong dependence on number of machined items and date of production. That means that the method is intrinsically affected by a sort of uncertainty which is not a priori computable. A further constraint has been added in the development of the piece of software called SMARTGT and described in this paper: **interactivity**. The designer may visualize similar parts stored in the PDM at the end of the modelling session, or, in addition, decide whether change singular feature or use a retrieved part, in order to save time during the modelling of the single component. Thus single feature subset search on very large dataset is a time-consuming activity and, if SMARTGT wants to match this requirement, similarity search algorithms have to emphasize the speed much more the precision. This increases the part comparison uncertainty, but the final results have demonstrated the successful approach.

1.2 State-of-the-art

Application benefits of part similarity detection in mechanical component variety in terms of production cost and component variety reduction has been demonstrated by several authors. But the range of solutions is quite wide. The shape similarity search in a Group Technology (GT) was introduced by Cardone [4], where, in addition, computer algorithms are considered important to improve the GT strategy. Barton [2] describes the obtained reduction in factory production costs through the usage of similar part just manufactured instead of the development of new ones from scratch. That means a competitive advantage due to the sparing in raw materials, spare parts requirement, tools machine settings, and assembly sequences. Gao et al. [7] classified the main methods for part similarity detection as: Global feature-based, Manufacturing feature-based, Graph-based, Histogram-based, Product information-based, Multiresolutional Descriptor-based.

The Global feature-based method [6] is representative of all strategies that use global properties of the 3D part to assess similarity. Geometry ratios, distances, curvatures, and moments of inertia can be used for the assessment. Some authors apply also 3D Zernike moments [12], or Fourier Transforms [16]. This approach presents the advantage of the quickness of computation, but can fail if the components to compare are similar in the overall shape but different in the local geometry.

Manufacturing feature-based methods recognizes manufacturing features from the CAD models [14]: these methods are often used to compare mechanical parts.

In the paper of Cardone, Gupta et al. [3] prismatic part similarity is computed considering the access and orientation of tool during part manufacturing. The method is efficient and accurate, but similarity search over huge database components finalized to a real-time application needs a simplified string comparing and a more general approach on all kind of components.

Also the MDG (Model Dependency Graph) approach to the problem belongs to this family of methods. The main drawback of this strategy is the computational weight and the complexity of the algorithms used for cost evaluation: reference system transforms, distances in the space and other computations should be done to encode the shape and to assess similarity.

The Graph-based methods exploits an encoding shape, geometry and features of a components in a graph representation, whose structure is representative of the 3D model [11]. An original strategy to

obtain a graph after a 3D part thinning process is the skeleton graph approach. The graph based methods are limited in the complexity of the graph itself, especially when a great number of details have to be represented.

One of the most interesting application of Histogram-based methods has been carried out recently by Hong [10]: the method proposed is based upon a algorithm based on two steps: in the first step it is performed a Shape Distribution Graph (SDG) analysis so that a preliminary screening is made; in the second step a detailed features analysis is started on a lower number of parts.

Product Information-based method has been used by Henderson et al. [9] for GT coding and shape similarity detection, related to Agile Manufacturing. The main drawback of this approach is the semi-automatic operation for a correct GT encoding of a general component. Multi-resolutional Descriptor-based technique [15] has been recently investigated by Gao et al. [7], who applied a Dilation Based Multi-resolutional Skeleton (DBMS) approach for part retrieving in this operations' sequence: building the component B-Rep, voxelization, DBMS graph generation and search in the database for similarity evaluation. The Reeb graph theory has been applied [1] not only to obtain part retrieval, but also for sub-part matching between part of solids. The computational cost of the method boosts as the resolution of the descriptor is increased. All the methods for similar part retrieving cited above require a computational effort that is not compatible with real-time operations. The feature-by-feature encoding during a CAD modelling session introduced in SMARTGT eliminates the calculation overhead and gives to the designer the choice to review and optimise its work during the session itself and following hints coming from the similar part searching and retrieving. All component retrieved are already manufactured and thus PDM record contains costs, project membership, usage frequency...

Through feature-by-feature GT generation, the search doesn't involve the geometry, but only the GT code, thus a string only, like the definition given by Harkalakis [8]: "a GT code is a string of alphanumeric characters that represent various design and manufacturing related characteristics".

The similarity detection can be made in an effective and fast way comparing the GT codes of components in a database. The results obtained in the components development work-flow are standardization of parts and of CNC programs and cost evaluation. The most successful and wide spread GT coding schemes developed for mechanical components with an automatic software has been developed by Optiz [13] and Henderson [9]. The GT code can be automatically obtained also by drawings in the IGES format [5].

The following section describes the software implementation either for 2D CAD drawings, either for 3D interactive feature-by-feature CAD plug-in.

2 SMARTGT: A GT CODE MANAGEMENT FOR REAL-TIME STANDARDIZATION

The real-time component encoding and part similarity search has been implemented in a computer program, called SMARTGT, which shares the same software core, but shows to the operator three different GUI (Graphical User Interface) : 2D part coding interface, 3D parts coding interface and CAD plug-in and similar GT code search and comparison.

2.1 Part Encoding

The 2D part encoding has been written having previously classified a large number of mechanical component drawings (about 5 000) and deducted a list of TAGs that identifies bi-univocally basic features (included also materials), technological features (i.e. thermal treatment), geometric features (cylindrical chamfer) and additional features (like a sheet metal bending). The final TAG list is summarized in Table 1-3. The features listed in Table 1 are to be considered as well as machining features and not in CAD meaning, that includes not only volume removal, but also mass addition. This choice has been taken in order to be more adherent to machining problems and most of these features have been extracted and compared also with ISO NC programming manual.

	Feature	Options	GT Tag	N° par.
Basic TAGS	Material		MAT	1
	Semimanufactured	casting	SLA	
		slab	SLB	3
		exagonal slab	SLC	2
		cylindric tube	SLD	3
		cylindric bar	SLT	2
		square tube	SLE	3
		HE beam (UNI5397)	SLF	2
		IPE beam (UNI5398)	SLG	1
		IPN beam (UNI5679)	SLH	1
		UPN beam (UNI10279)	SLI	4
		"T" beam (UNI5681)	SLJ	1
		"L" beam (UNI10056)	SLK	3
		"Z" beam (UNI 6763)	SLL	4
		"T" beam (UNI 10055)	SLM	1
		"U" beam (UNI EU54)	SLN	2
		"L" beam (UNI 6762)	SLO	3
	Sheet metal	SLP	3	

Tab. 1: List of TAGs used for GT code (part 1).

The GT code is generated by single feature encoding and the final string is composed by TAGs plus parameters. Each TAG (elemental linguistic and compact description of the feature) with its parameters, put between square brackets, follows the previous without any other mark. In this way some generic characteristics may be identified:

- The GT code is not biunique with the component: in other terms, the same component may be represented by several GT code with same TAGs and parameters, but different sequence. The contrary is not true.
- There are not one-to-one relationships between TAGs in order to define a priority relative position.
- An internal order by groups is given only by four TAG sets: Basic, Technologic, Geometric, Additional, that will be better described further in the paragraph.
- The TAGs sequence is not artificially reordered.
- The GT code has different length depending on component complexity and sometimes the alphanumeric string results quite long, but that doesn't affect significantly search time.
- The TAG list is open. The implemented software is able to work perfectly also with an wider list of TAGs, where some of them are defined by the user.

Technologic TAGS	Thermal treatment	TERMTR	1
	Tecnologic treatment	TECNTR	1
	Global finishing	FINSGL	1
	Local finishing on plane surface	FINSLP	3
	Local finishing on cylindric surface	FINSLC	3
Geometric TAGS	External cylinder	CIL_EST	2
	Internal cylinder	CIL_INT	2
	External cone	CONINT	3
	Internal cone	CONEST	3
	Developed sheet metal	LAMSVI	3
	External cylindric chamfer	SM_EST	2
	Internal cylindric chamfer	SM_INT	2
	Chamfer on plane edges	SM_BOR	3
	Rounding on planar edges	RA_BOR	2
	External cylindric rounding	RACEST	1
Internal cylindric rounding	RACINT	1	

Tab. 2: List of TAGs used for GT code (part 2).

Additional Feature	TAGS			
	External cylindric relief		GOLEST	2
	Internal cylindric relief		GOLINT	2
	External planar relief		GOLEP	2
	Internal planar relief		GOLIP	2
	Complex relief		GOLCOM	7
	Generic hole		FOR	2
	Hole passing through		FORP	1
	Axial hole		FORASS	2
	Threadings	Fil.Metric	FILM	2
		Fil.Gas	FILG	2
		Fil.Whitworth	FILW	2
		Fil.Trapezoidal	FILT	2
	Chasings	Fil.Metric	FASSIM	2
		Fil.Gas	FASSIG	2
		Fil.Whitworth	FASSIW	2
		Fil.Trapezoidal	FASSIT	2
	Boring		FILALE	2
	Conic hole		FORCON	3
	Levelling		SP	2
	Levelling on shaft		SPALB	2
	Oblique levelling		SPSGH	4
	Cut on a curve		SCAPER	2
	Planar cut		SCAPIA	3
	Cut for wedge		SC_CHI	2
	Cut for tang		SC_LIN	2
	Cut for circular tang		SC_DIS	2
	Internal cut		SCI_CH	2
	Cut with circular section (for Seeger)		SC_SEE	2
	sheet metal angular bending		PIEGA	3
	sheet metal circular bending		PIECIR	2
	Sheet metal drawing		IMBUT	4
	Sheet metal cut		TALAM	1
	Cut on a curve		TAGPER	1
	Welding		SALD	
	Sheet metal bending		PIEGA	
	More complex shape		FEASUP	

Tab. 3: List of TAGs used for GT code (part 3).

The TAG list has been divided in 4 groups: Basic, Technologic, Geometric and Additional features. Beneath a one-by-one feature priority relationships and constraints have not been introduced, the proposed TAG grouping comes both from logic considerations and practical experience. It should be natural that final GT code shows TAGs concatenated in the sequence listed in Table 3. In fact Basic features represent material properties and semi-manufactured components; Technologic features are thermal treatments and global/local finishing; Geometric features are machining operations dedicated to transform the semi-manufactured and Additional features are needed to complete the component.

The morphological string sequence (GT code) starts with the TAG **MAT**, which describes the material used, and a semi-manufactured TAG, outlining the basic shape. Further, some TAGs on treatment and finishing. The following TAGs are dedicated to features eventually added to the base shape. For instance the TAG **GOLIP** describes an internal planar relief with two parameters: nominal shaft diameter and relief radius.

External roundings (Fig. 1 and Fig. 2) are added to the GT string using the TAG **RACEST**, which requires only one parameter, the radius. The TAG **SM_EST** translate into the string a chamfer over an external edge: in this case two parameters are required, the length of the chamfer and the angle. All the most wide spread features, also the most complex like wedges (Fig. 3), can be represented by a TAG. For instance, the **SC_CHI** requires two dimensions to describe the wedge: length and depth.

Finally, beneath the TAG list is targeted to encode the largest set of components, it is not possible to represent all type of mechanical parts and a generic "More complex shape" feature has been introduced in order to include also some geometric or technological features that cannot be as a sequence of other simpler features. Especially chasings, free-form modelled external coverings, particularly shaped guides...are examples where all evident and detailed TAGs have been listed

together with a coarse feature without parameters because are too complex. Secondly the FEASUP is useful for the operator to extend the TAG list with user-defined feature.

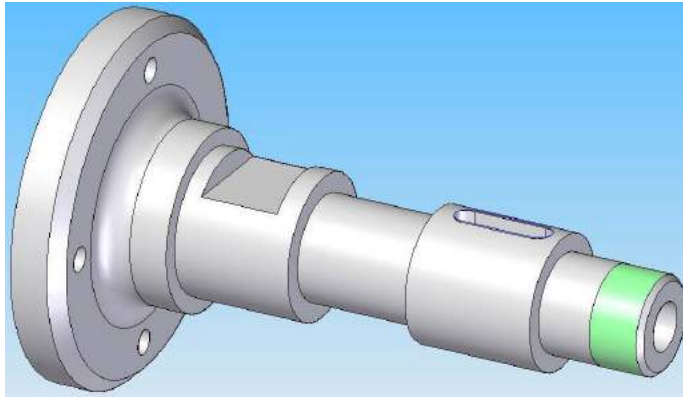


Fig. 1: 3D component for 2D encoding applet.

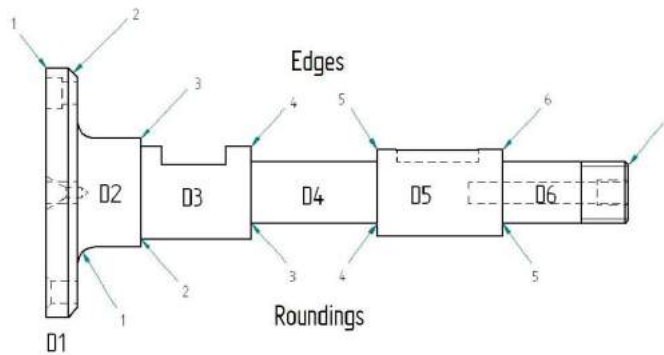


Fig. 2: Roundings and edges.

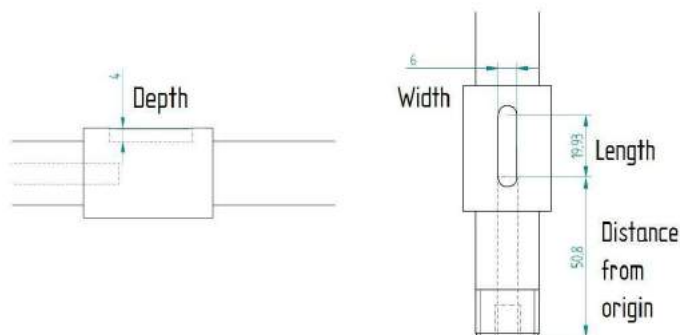


Fig. 3: Wedge specifications.

2.2 2D Encoding Interface

The component draft is still fundamental to correctly evaluate machining operations. Some textual additions introduced in the 2D draw are generally not present in the 3D CAD modelled part. So the bi-dimensional draft is considered available also in 3D case.

SMARTGT has been designed to work with only a paper 2D draft or a 3D shaped component with its digital draft either way. In fact it exploits both a 2D interface and a 3D CAD plug-in. The GUI for 2D drafts, needed to produce the GT code also starting from non digital drawings, still available in many SMI, is useful in particular when outsourcing service is active. The interface is represented by a wizard with a question sequence targeted to concatenate the GT code.

The automation level is very limited because the operator is charged to choose within several iconic and textual requests. Also operation order is mediated and filtered by human experience and wizard GUI. The interface has been programmed in VBA (Visual Basic for Applications) and interacts directly with PDM database to store digitalized draft and final GT code.

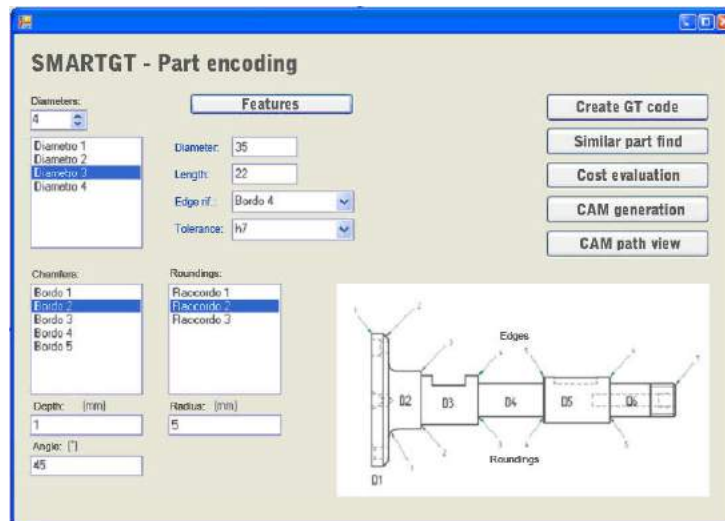


Fig. 4: The code generation window with options.



Fig. 5: The final GT code.

Fig. 4 and Fig. 5 are screenshots from the main guided interface: a simplified picture and the corresponding TAG guides the operator to build the correct GT code.

Basic shapes and external finishing choices are in a hierarchical arrangement in order to exclude and simplify the procedure, avoiding also duplicated requests. The feature form viewable in Fig. 4 is

different and customized over the single TAG in order to better highlight single parameters. From this form it is simple to access to the fundamental SMARTGT functions, like computing the GT code (Fig. 5), similar part finding through PDM interrogation, cost evaluation and confrontation once the similar components have been retrieved. I available, also already used NC programs for these parts, can be opened and evaluated.

2.3 3D CAD Integration

The CAD plug-in is the piece of software that involves directly the new part design and modelling. SMARTGT is a menu (floating window) (**Error! Reference source not found.**) and is available during the modeling session (Fig. 6), like an electronic tutor. UGS Solid Edge has been used for implementation due to easy procedure in VBA and good integration with SmartTeam PDM/PLM. While the operator proceeds with the part modelling adding features, the SMARTGT Features Analysis software updates the GT string, working in background. The floating window (**Error! Reference source not found.**) reports in the upper side choices that are independent from the shape (like material, semi-manufactured...), but mandatory for the operator.

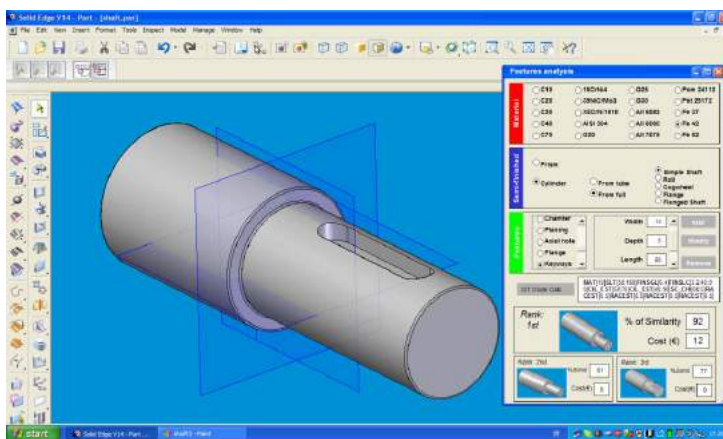


Fig. 6: 3D Solid Edge plug-in for part encoding.

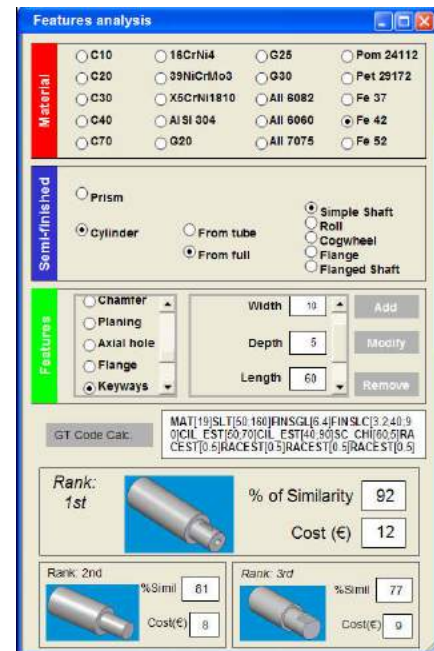


Fig. 7: SMARTGT Feature Analysis window.

Of course these features are related with basic shapes and surface finishing. Furthermore, coded features (TAGs) have been programmed using variational shapes in order to avoid time consuming feature recognition and reducing errors in GT code creation. While the operator chooses the standard feature and introduces geometrical parameters, the SMARTGT adds to the string the correct TAG with parameters/dimensions. The incremental GT code is showed in a small box in the menu window: this code is used to start a background task which is dedicated to search similar components by matching GT codes. This task may be also activated by the operator directly.

So SMARTGT searches in the SmartTeam database the three more similar parts, listing in the lower box percentage of similarity (more details on this value will be given in the next section) and the related costs. The retrieved similar parts are painted in small pictures in the lower end of the window. Thus the designer is interactively informed on similar components available in the PDM with estimated cost and may decide to review the work, changing some features, or definitely replace the current model

with one of the retrieved parts. So the standardization strategy is included in the design process in the very first and generative step.

2.4 Similarity Ratio Computation

The similarity assessment is performed exploiting standard pattern matching algorithm between GT TAGs, related to the component under development, and GT codes stored in SmartTeam™ PDM database. Confrontation means evaluation of intrinsic parameters in both strings:

- number of TAGs present in both GT codes (n_t); this parameter evaluates the number of identical TAGs with respect the average length of GT codes;
- TAG position in the GT code (p_t) (same TAGs in different order may correspond to different components); the p_t coefficient evaluates the features order;
- dimensions in same TAGs (d_t) (denote the local or global component scaling);
- TAG typology (t_t) (basic, technologic, geometric and additional).

Thus a similarity coefficient can be calculated from a balanced combination of these normalized parameters:

$$S_c = \sum_{i=1}^4 p_i r_i \quad (1)$$

where:

$$(a) \quad p_i = \{n_t, p_t, d_t, t_t\};$$

$$(b) \quad n_t = \frac{n_{com}}{(n_1 + n_2)} \cdot 2$$

with:

n_{com} = number of common TAGs,

n_1 = total number of TAGs in GT code under development,

n_2 = total number of TAGs in GT code in PDM;

$$(c) \quad p_t = \frac{\alpha}{n_{com}} \cdot \sum_{i=1}^{n_{com}} \left(1 - \frac{|p_{i,1}^* - p_{i,2}^*|}{n_{com}} \right) + \frac{\beta}{\min(n_1, n_2)} \cdot \sum_{i=1}^{n_{com}} \left(1 - \frac{|p_{i,1} - p_{i,2}|}{\max(n_1, n_2)} \right)$$

$p_{i,1}$ = TAG position in GT code under development;

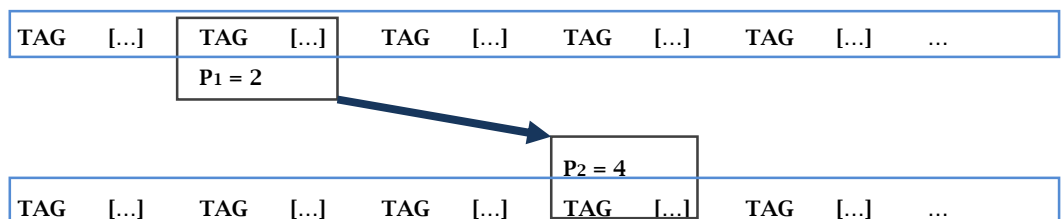
$p_{i,2}$ = TAG position in GT code in PDM;

$p_{i,1}^*$ = TAG position of parts under development in virtual common string;

$p_{i,2}^*$ = TAG position in virtual common string of the GT code in PDM;

$\alpha = 0.3$; $\beta = 0.7$; assuming that, if $n_{com} = 0$, then $p_t = 0$.

In general, TAG position refers to the ordinal position number in the string, as viewable in the following example:



Coefficients $p_{i,1}$ and $p_{i,2}$ are calculated using the configuration above. In order to calculate $p_{i,1}^*$ and $p_{i,2}^*$ instead, a virtual string obtained excluding all non common TAGs is used. In this way the GT code has identical length = n_{com} .

$$(d) \quad d_t = \frac{1}{n_{com}} \sum_{i=1}^{n_{com}} \frac{1}{m_i} \sum_{j=1}^{m_i} \left(1 - \frac{|d_{1_{ij}} - d_{2_{ij}}|}{\max(d_{1_{ij}}, d_{2_{ij}})} \right)$$

with:

$d_{1_{ij}}$ = dimension in the same TAG within GT code under development,

$d_{2_{ij}}$ = dimension in the same TAG within GT code in PDM;

m_i = number of parameters in compared TAGs (in general TAGs have different parameters).

$$(e) \quad t_t = \frac{1}{n_{com}} \sum_{i=1}^{n_{com}} t_i \quad \text{where: } t_i = \begin{cases} 0.55 & \text{in case of } \textit{basic feature} \\ 0.25 & \text{in case of } \textit{technologic feature} \\ 0.15 & \text{in case of } \textit{geometric feature} \\ 0.05 & \text{in case of } \textit{additional feature} \end{cases}$$

More in detail, n_i represents the influence on the final similarity given by the common TAGs. The term named p_i evaluates the contribution to similarity given by the analogue position of common TAGs. The first part is dedicated to weight position exchanges between TAGs eliminating different TAGs; the second part is targeted to understand if these exchanges are also due to different string length. The coefficient d_i is useful to estimate differences given by TAGs parameters. The last coefficient t_i assesses the distribution of TAGs in the previous described categories (basic, technologic, geometric, additional). Finally r_i in Equation 1 are the relevance coefficient given by Table 4.

Intrinsic parameter	Relevance
Number of TAGs (n)	0.65
Tag position (p)	0.15
Dimensions (d)	0.10
Tag typology (t)	0.10

Tab. 4: Relevance coefficients.

Part matching on the basis only of GT code guarantees great benefits in terms of flexibility and search speed: not only this rules may be changed by the operator at any time, but also a custom query in the PDM database may be also done assembling TAGs and parameters in an unconventional way. That may be useful to find special components or correlate features with costs for project management. The single feature is added to the part code, using the interface described in **Error! Reference source not found.**, where all manually introduced data can be inserted in a simple way.

3 CASE STUDIES

This section is dedicated to describe some GT code building from practical examples.

3.1 2D Case Study

The part depicted in the drawing of Fig. 8 has been coded into the SMARTGT software using the interface for 2D draft.

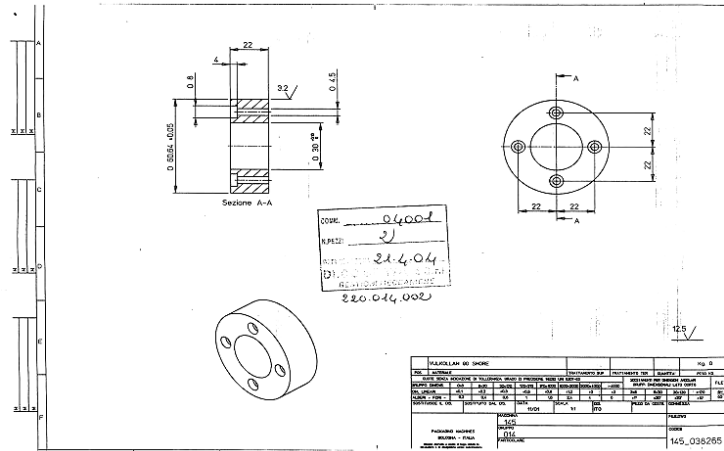


Fig. 8: Draft to be encoded.

The associated code is:

`MAT[16]SLT[22;60]FINSGL[6.4]FINSCL[3.2;60;22]CIL_EST[60;22];CIL_INT[30;PASS]FOR[8;4]FORP[4.5]FOR[8;4]FORP[4.5] FOR[8;4]FORP[4.5] FOR[8;4]FORP[4.5]`. The code TAGs are explained in the following lines.

- `MAT[16]` material: rubber
- `SLT[60;22]` the raw material is made by a round bar with a diameter of 60 mm and a length of 22mm.
- `FINSGL[6.4]` global finishing of Ra 6.4
- `FINSCL[3.2;60;22]` local finishing of Ra 3.2 over a diameter of 60mm, for a length of 22 mm
- `CIL_EST[60;22]` external cylinder of 60 mm diameter and 22mm length
- `CIL_INT[30;PASS]` passing cylindrical hole of 30mm in diameter
- `FOR[8;4]` hole with a diameter of 8 mm and a depth of 4 mm
- `FORP[4.5]` passing hole with a diameter of 4.5 mm

Another example is depicted in Fig. 9, where the mechanical component is represented in 3D and 2D using Solid Edge CAD environment and encoded in SMARTGT.

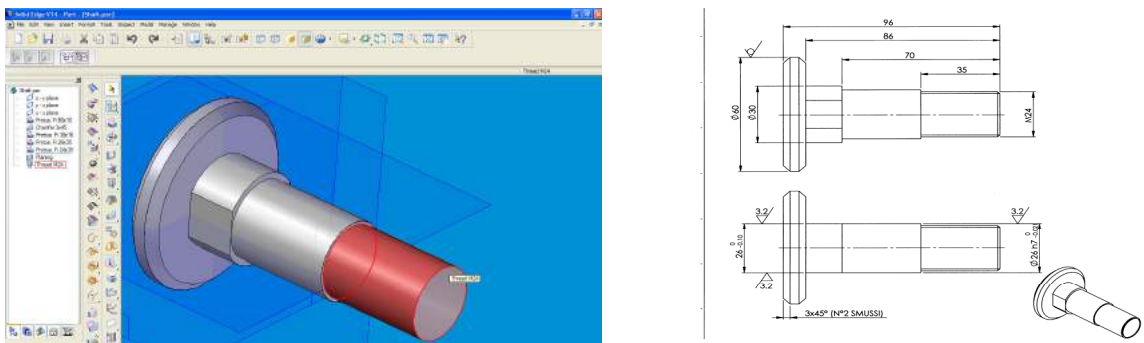


Fig. 9: Part encoded in the 3D CAD interface and in the draft.

The resulting TAG list is listed below:

MAT[3]SLT[60;96]TECNTR[1]FINSGL[12.5]FINSLP[3,2;10;15]FINSLC[1,6;60;4]CONEST[60;54;3]CIL_EST[60;4]CONEST[60;54;3] CIL_EST[30,16]CIL_EST[26,35]FASSIM[24;35]SPALB[16;13]SPALB[16;13]

The code TAGs are explained in the following lines: it should be noted as the TAGs are added in parallel with the sketching in the CAD environment, following a sort depending on the design procedure.

MAT[3]	Indication of material: C40
SLT[60;96]	Indication of raw material: round bar with a diameter 60 mm and a length of 96 mm
TECNTR [1]	Indication of technological treatment: burnishing
FINSGL[12.5]	Global finishing of Ra12.5
FINSLP[3,2;10;15]	Local finishing of Ra3,2 in a plane zone, with dimensions 10x15mm
FINSLC[1,6;60;4]	Local finishing of Ra1,6 over round zone 60 mm diameter and a length of 4 mm
CONEST[60;54;3]	Cone with a width of 3 mm
CIL_EST[60;4]	Cylinder with a width of 4 mm
CONEST[60;54;3]	Cone with a width of 3 mm
CIL_EST[30;16]	Cylinder of diameter of 30mm, length of 16 mm
CIL_EST[26;35]	Cylinder of diameter of 26 mm, length of 35 mm
FASSIM[24;35]	External threading M24 for a length of 35 mm
SPALB[16;13]	Levelling on shaft with a length of 16 mm, and a distance of 13 mm from the axis
SPALB[16;13]	Levelling on shaft with a length of 16 mm, and a distance of 13 mm from the axis.

3.2 Similarity Evaluation

The current paragraph is dedicated to some practical calculus of GT code and comparison between real components in order to test the similarity assessment.

As an example of similarity evaluation we assess the two parts, whose code has been obtained by SMARTGT, Fig. 10. The GT code of Flange 1 is made of 7 TAGs:

MAT[12]SLT[5.5,16]TECNTR[7]FINSGL[12.5]CIL_EST[16;1.5]CIL_EST[11;4]CIL_INT[9;5.5].

The GT code of Flange 2 is made of 14 TAGs:

MAT[16]SLT[22;60]FINSGL[6.4]FINSLC[3,2;60;22]CIL_EST[60;22];CIL_INT[30;22]; →
 → FOR[8;4]FORP[4.5]FOR[8;4]FORP[4.5]FOR[8;4]FORP[4.5]FOR[8;4]FORP[4.5]

The number of common TAGs is 5, while their position (indicated with #) is quite similar.

Flange 1 → MAT[12] (#1) SLT[5.5;16] (#2) FINSGL[12.5] (#4) CIL_EST[16;1.5] (#5) CIL_INT[9;5.5] (#7)

Flange 2 → MAT[16] (#1) SLT[22;60] (#2) FINSGL[6.4] (#3) CIL_EST[60;22] (#5) CIL_INT[30;22] (#6)

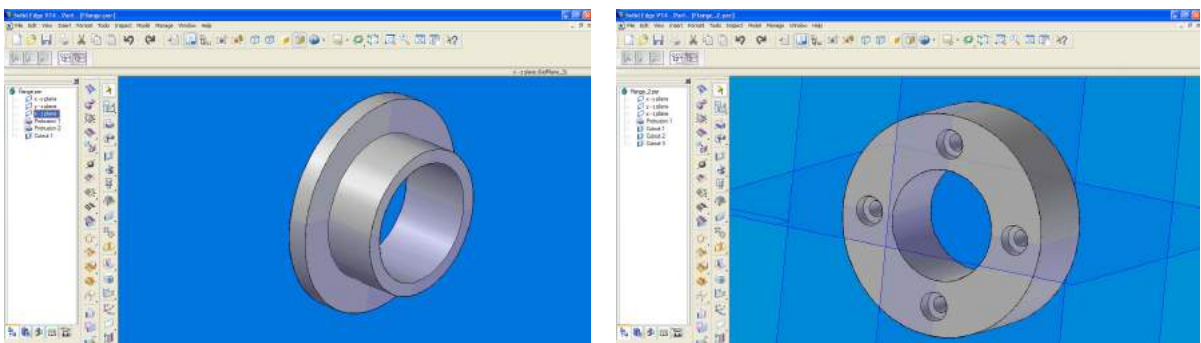


Fig. 10: Similarity evaluation: Flange 1 (left) and Flange 2 (right).

Thus, calculating the coefficients for Equation 1, the similarity rate is given by:

$$p_i = \{n_i, p_i, d_i, t_i\} = \{0.476, 0.786, 0.411, 0.330\}$$

$$r_i = \{0.65, 0.15, 0.1, 0.1\}$$

$$S_c = \sum_{i=1}^4 p_i r_i = 0.501 \rightarrow 50\%$$

This rather good matching is due to a similar starting semi-finished parts, but the different string lengths and n_{com} quite near to n_1 determine a good basic matching, but several additional machining operations prevent the similarity assessment to grow over 50%.

3.3 Range Sensitivity

Further and more complex examples are provided in this paragraph to test SMARTGT software usability and final results, in particular when the number of TAGs is growing.

The examples in Fig. 11: Shaft 1 (left), Shaft 2 (middle), Slide (right). Fig. 11 show two similar axial symmetric parts, named shaft 1 and shaft 2, and a prismatic one: a slide for lathe. These test cases have been chosen in order to test different level of similarity: in case of Shaft 1 and Shaft 2 the similarity rate is expected very high, whereas the part matching between Shaft 1 and Slide is expected to return a very low rate, of course.



Fig. 11: Shaft 1 (left), Shaft 2 (middle), Slide (right).

The Shaft 1 is coded in the following string, made of 24 TAGs:

```
MAT[6]SLT[255;95;]TERMTR[2]FINSG[6.4]FINSLC[3.2;32;70]CIL_EST[30;40]SM_EST[45;2]FASSIM[30;40]GOLEST[28;2]CIL_EST[32;70]GOLEST[30;2]CONEST[60;50;30]CIL_EST[92;15]GOLEST[58;2]CONEST[60;26;56]GOLEST[28;2]CIL_EST[30;30]SM_EST[45;2]FASSIM[30;30]FORASS[10;35]FASSIM[10;30]SC_LIN[8;60]SCAPIA[15;14;6]SCAPIA[15;14;6]
```

The Shaft 2 is coded in the following string, made of 22 TAGs:

```
MAT[6]SLT[220;75;]TERMTR[2]FINSG[6.4]FINSLC[3.2;30;85]CIL_EST[30;25]SM_EST[45;2]FASSIM[30;26]GOLEST[28;2]CIL_EST[32;86]GOLEST[30;2]CONEST[60;50;30]CIL_EST[72;15]GOLEST[58;2]CONEST[60;30;32]GOLEST[28;2]CIL_EST[30;24]SM_EST[45;2]FASSIM[30;24]SC_LIN[8;50]SCAPIA[15;14;5]SCAPIA[15;14;5]
```

The Slide is coded in the following string, made of 36 TAGs:

```
MAT[3]SLB[300;150;35]TERMTR[3]TECNTR[3]FINSG[3.2]FINSLP[1.6;108;300]SM_BOR[45;2;300]SM_BOR[45;2;300]SM_BOR[45;2;300]SM_BOR[45;2;34]SM_BOR[45;2;34]SM_BOR[45;2;34]SM_BOR[45;2;34]SM_BOR[45;2;150]SM_BOR[45;2;34]SM_BOR[45;2;34]SM_BOR[45;2;34]SCAPER[300;15]SCAPER[300;15]SCAPER[200;13]SCAPER[200;13]FORP[15]FORP[15]FORP[80;5]SM_EST[45;2]FORP[60]SM_EST[45;2]FORP[10]FILM[10;19]FORP[10]FILM[10;19]FORP[10]FILM[10;19]FORP[10]FILM[10;19]
```

The similarity coefficient between shaft 1 and shaft 2 can be computed as follows:

$$p_i = \{n_i, p_i, d_i, t_i\} = \{0.957, 0.992, 0.927, 0.159\}$$

$$S_c = \sum_{i=1}^4 p_i r_i = 0.879 \rightarrow 88\%$$

As expected, the similarity of the two part is high. At the contrary the similarity between shaft 1 and slide is very poor; the algorithm gives a rate of:

$$p_i = \{n_i, p_i, d_i, t_i\} = \{0.1, 0.387, 0.55, 0.35\}$$

$$S_c = \sum_{i=1}^4 p_i r_i = 0.213 \rightarrow 21.3\%$$

These examples have been introduced in order to check the similarity ratio sensitivity both in very similar parts and in clearly different shaped and machined components.

4 RESULTS AND CONCLUSIONS

The paper reports a real-time implementation of a special designed Group Technology code for mechanical component standardization, part similarity and cost evaluation. A fast and almost interactive comparing GT codes method has been also proposed with a strong integration with 3D CAD modelling, respecting also CAM directives. Several tests performed in DI.CO Service company proved the efficiency in similar part retrieving, while search time haven't ever exceeded 2 seconds for very long GT codes. The testing stage is still in progress due to the need to improve the components that have to be encoded: this operation is a essential prerequisite for the correct SMARTGT working. At the moment, the number of records used for tests is around 8 000 on a global database containing 20 000 items. All tests have proved the SMARTGT interactivity and the CAD interface have positively stimulated designers is standardization from the very first design stage. After a couple of months from the introduction of SMARTGT in the enterprise PLM, the components dissimilarity and variety has been reduced by 9% and that makes confident of further improvements.

5 ACKNOWLEDGEMENT

Authors wish to thank the Italian company DI.CO Service s.r.l., Ozzano Emilia - Bologna, for the help and the support in the research.

6 REFERENCES

- [1] Biasotti, S.; Marini, S.; Spagnuolo, M.; Falcidieno, B.: Sub-part correspondence by structural descriptors of 3D shapes, *Computer-Aided Design, Computer-Aided Design*, 38(9), 2006, 1002-1019.
- [2] Burton, J.; Love, D.: Retrieving designs from a sketch using an automated GT coding and classification system, *Production Planning & Control*, Taylor & Francis, 16(8), 2005, 763-773.
- [3] Cardone, A.; Gupta, S. K.; Deshmukh, A.; Karnik, M.: Machining feature-based similarity assessment algorithms for prismatic machined parts, *Computer-Aided Design, Computer-Aided Design*, 38(9), 2006, 954-972.
- [4] Cardone, A.; Gupta, S. K.; Karnik, M.: A survey of shape similarity assessment algorithms for product design and manufacturing applications, *Journal of Computing and Information Science in Engineering, Journal of Computing and Information Science in Engineering*, 3(2), 2003, 109-118.
- [5] Chen, C. S.: A Form Feature Oriented Coding Scheme, *Computers and Industrial Engineering, Computers and Industrial Engineering*, 17(1), 1989, 227-233.
- [6] Funkhouser, T.; Min, P.; Chen, J.; Halderman, A.; Dobkin, D.: A search engine for 3D model, *ACM Transaction on Graphics*, 2003, 83-105.

- [7] Gao, W.; Gao, S. M.; Liu, Y. S.; Bai, J.; Hu, B. K.: Multiresolutional similarity assessment and retrieval of solid models based on DBMS, *Computer-Aided Design*, *Computer-Aided Design*, 2006, 985-1001.
- [8] Harhalakis, G.; Kinsey, A.; Minis, I.: Automated Group Technology Code Generation Using PDES, In *Proceedings of Third International Conference on Computer Integrated Manufacturing*, Rennsealer Polytechnic Institute, Troy, New York (USA), 20-22 May, 1992, Rennsealer Polytechnic Institute, Troy, New York (USA), 4-14.
- [9] Henderson, M.; Musti, S.: Automated Group Technology Part Coding from a Three-Dimensional CAD Database, *Journal of Engineering in Industry*, 110(3), 1988, 278-287.
- [10] Hong, T.; Lee, K.; Kim, S.: Similarity comparison of mechanical parts to reuse existing designs, *Computer-Aided Design*, *Computer-Aided Design*, 38(9), 2006, 973-984.
- [11] Joshi, S.; Chang, T. C.: Graph-based Heuristics for Recognition of Machined Features from a 3D Solid Model, *Computer-Aided Design*, 20(2), 1988, 58-66.
- [12] Novotni, M.; Klein, R.: Shape retrieval using 3D Zernike descriptors, In *Proceedings of Eighth ACM symposium on Solid modeling and applications*, Seattle (USA), 2003, 216-225.
- [13] Opitz, H.: *A Classification System to Describe Workpieces*, Pergamon Press, New York (USA), 1970.
- [14] Shah, J. J.; Bhatnagar, A.: Group Technology Classification from Feature Based Geometric Models, *Manufacturing Review*, 2(3), 1989, 204-213.
- [15] Tung, T.; Schmitt, F.: The augmented multiresolution Reeb graph approach for content-based retrieval of 3D shapes, *International Journal of Shape Modelling*, 11(1), 2005, 91-120.
- [16] Vranic, D. V.; Saupe, D.: Descriptor Based on 3D Fourier Transform, In *Proceedings of EURASIP Conference on Digital Signal Processing for Multimedia Communications and Services (ECMCS 2001)*, Budapest, Hungary, 2001, 271-274.