

Study of 3D Model Function-coded Genome in Evolutionary Design

Jihua Wang, Hong Liu

College of Information Science and Engineering Shandong Normal University,
250014 Jinan, China.

{Jihua.Wang, Hong.Liu, jihuaaw@126.com}

Abstract. Evolutionary design is one kind of intelligently cooperative design, which is analogous to the evolutionary process of products. A forming method for the genome of 3D model in evolutionary design is proposed based on feature functions and topology functions. 3D model consists of features, which can respond to the environment by their functional faces and parameters, the genome which is represented by the features and their topology has been studied. The feature and topology functions are combined with geometry API in the CAD software. The genome which is symbolized as feature and topology functions is operated for the 3D model evolutionary design by genetic operators. So the genome can be taken out from the rich CAD model database to form enough scale initial population. At last, the test of phone models demonstrated the evolutionary design process including gaining their genomes automatically, evolutionary computation of their design and visualization, etc.

Key words: evolutionary design; genome; feature; topology; API; 3D model; representation

1 Introduction

While computer-aided design (CAD) has generally been expanded in the field of industrial design, engineering and animations, the development of artificial intelligence is giving rise to intelligent CAD (ICAD). Evolutionary computation is one kind of artificial intelligence which simulates the process of natural evolution; its application is called evolutionary design in many design fields and every design stage especially conceptual design [1, 2]. In addition to all the advantages of evolutionary computation [3], evolutionary design is superior to other intelligent design such as case-based reasoning, qualitative reasoning, etc. First, its parallelism to give many design plans at mean time is beneficial to cooperative design and rapid design. Second, its implicit learning reduces designers' workload especially eliminating the logic laws from products. Third, its creativity caused by such genetic operators as mutation, crossover, selection can give inspiration to designers directly.

Evolutionary design, like evolutionary computation, is founded on genome coding, initial population, fitness evaluation, and algorithms type, among which the genome coding is the core and one of the most important paths to efficient search and solving.

The product genome or representation has extensively been applied in all design stages of function design, conceptual design, structure design, assembly design, manufacturing design, the detailed design, CAE, CAPP, cost estimation, etc.. It also must be set up in all modern design means of parallel design, cooperative design, virtual reality, etc. in the global, network-centric, and spatially distributed environment. The product is usually decomposed into elements such as feature, geometry, function surface, face, function unit, structure unit, solid volume, entity and topology, etc., and then represented using them and their graphs or Boolean relations.

With the present gap between CAD and CAE, the multi-representation architecture, which views CAD-CAE integration as an information-intensive mapping between design models and analysis models, is described by object and constraint graph techniques [4]. A hierarchically structured constraint-based data model for solid modeling in the virtual reality environment is presented, which integrates a high-level constraint-based model for precise object definition, a mid-level CSG/B-rep hybrid solid model for supporting hierarchical geometry abstractions and object creation, and a low-level polygon model for real-time visualization and interaction [5]. In the virtual manufacturing technology, Petri net manufacturing process model representation with part, form feature, production environment, production planning and cost information is used [6]. The representation in automatic assembly process consisting of geometric information and assembly-specific features, enables the mapping of the processes and the planning and optimization of reconfigurable assembly lines [7]. The Standard for the Exchange of Product Model Data-STEP is used for the product data representation in product data management system-PDMS, storing and managing complete information about products and related activities [8]. Product functions representation based on geometric modeling including sets of objects, operations, and relations are discussed [9]. A product representation in a feature-based computer aided design and manufacture environment from detail design, process planning to assembly planning is presented, in which features are considered to be machined volumes and are described in a hierarchical taxonomy; the assembly structure is also defined hierarchically with the machining features forming the basic entities, the assembly relationships among features are defined in the form of mating relationships that are carried by the features [10]. A product representation of assembly features abstractly represented as attributes with pertinent geometrical, technological and functional data in the faces, parts and sub-assemblies is given in the automatic generation of assembly sequences [11]. Product architectures encompass the transformation of product function to alternative product layouts; a functional decomposition basis and time ordered function chains are used to formally derive functional models of products for identifying modules of product architectures in the product development process [12]. The representation scheme of the CAD model is represented in the retrieval and matching processes in a database of mechanical components using attributed graphs in which the nodes correspond to the surfaces of the component, the links correspond to the edges of the component, and the graph is based on unique STEP data for a single component regardless of the underlying CAD system [13].

Evolutionary design, which is developed in widely-used computers and networks, plays the distinctive roles in product creativity. The structure of a product is the most important as the carrier of function, theory, and assembly relationship, which has

implicitly contained the information through the long-time product evolutionary process. So the product genome in evolutionary design directly adopts the structure-based data in CAD.

In evolutionary computation, the gene codes are binary, real, character, big-symbol, real-valued vectors, hierarchical procedure, etc. In recent years, the product genome or representation to reflect the evolutionary design characteristics has also been emphasized. Gero used all units and combination rules as the genome of automobile panels and architecture plans [14, 15]. Li Hongjie regarded the function-structure mapping matrix as products' genomes [16]. Soddu demonstrated that the genes are a series of generative procedures in designers' logic [17]. Bentley explored the product embryogenies to map genotype to phenotype in design [18] and took the octal-tree decomposition volumes as solid genes [19]. Huang Kezheng constructed the genomes with products' function surfaces [20]. Liu Hong took the bi-tree of functions as the genes of mobile-phones and desk lamps [21, 22, 23].

Genes are the basic genetic units to control a performance, which are integrated into the genome to mirror creature things. In evolutionary design, a gene as an operation unit should be encapsulated with windows, hereditary, unique, autonomous. But the above-mentioned genomes fall short of these natures. For example, some contain too little meaningful information, even only codes far from explicit, which are encapsulated well but are insufficiently inheritable and autonomous. Some genomes are set up by decomposing the product willfully, regardless of overall standards which have existed in current CAD software for a long time, so it is difficult to get more scale initial populations contributing to the evolutionary design superiority in the finished product databases. Some gene representations with graphics or matrix increase the complexity of design solutions.

This paper describes evolutionary design at viewpoint of the design process from the start of a task to its end. The geometry features as construction genes are linked with topology relations as topology genes, which are coded with feature and topology application procedure interfaces (API) in CAD software such as ACIS. The genes like agents can autonomously work by changing their parameters in API and have explicit connotations as design units. So the genome is concise to designers' communication, which directly adopts the current technology of CAD and evolutionary computation to help designers work in creativity.

2. Evolutionary design

Evolution is one of the most powerful search processes ever discovered in the natural world, which has created an unimaginably diverse range of designs of greater complexity than mankind could ever hope to achieve. Then evolutionary computation has rapidly been developed from genetic algorithm, evolutionary programming and evolutionary strategy which have been recognized in recent years. Researchers have begun using computer algorithms that mimic this process of evolution to automate stages of the human design process^[1]. The process of finishing a design task, which is optimized iteratively and gradually from vague to clear, is in fact similar to evolutionary computation, shown in table 1.

Table 1. comparison between design process and evolutionary computation

	Design process	Evolutionary computation
1	Design goal	Fitness function
2	Search many similar products	Initial population
3	Select the typical products to imitate	selection
4	Modify and re-combine the components of these products	Crossover, mutation
5	Select the better products as candidates	Selection
6	Repeat 4-5	Iterative
7	Adjust parameters to satisfy requirements	mutation
8	Audit and evaluate the plans	fitness
9	Repeat 2-8	iterative
10	Satisfied plans	Local optimization

Evolutionary design produces the more excellent next generations with the parents' merit, by operating the population of the similar product instances using selection, crossover and mutation, etc. in evolutionary algorithms led by fitness functions (design requirements).

Mutual evolutionary design on the basis of cooperative design systems in distributed workstations give the fitness with designers' intervention.

The following is typical evolutionary design procedure:

1. Step1 set time $t = 0$
2. Step2 create initial population POP(t): extract their genomes from the products in CAD database, and set up the population of all individuals similar to the objective product.
3. Step3 evaluate POP(t): input the individuals in the distributed workstations and all designers give their scores according to fitness function.
4. WHILE (no satisfied design plans) DO
5. $t = t + 1$
6. Step4 NEWPOP(t) = create new population using POP(t-1) by selection
7. Step5 CROSSPOP (t) = create population using NEWPOP(t) by crossover of single-point or multi-points.
8. Step6 MUTPOP (t) = create population using CROSSPOP(t) by mutation.
9. Step7 go to step3
- END WHILE

The initial population should be of sufficient scale to create more new plans. The fitness function is a kind of designers' judgement in light of requirement, so the large-scale population requires additional designers and workstations to work together simultaneously and parallel to coordinate the high-speed computers. In order to get the large-scale initial population and improve their evolutionary computation and evaluation, the genomes which map from the representation space (i.e. chromosome or genotype space) onto the search space (i.e. solution or phenotype space) should also emphatically be studied.

3. 3D model genome in the evolutionary design

In cognitive psychology, one object is identified according to its component features and their interrelations. So the basic units as genes a part or model is decomposed into in the light of certain rules are incorporated into the genome.

3.1 3D model structure decomposition granularity

The products' essence is functions. The functions of a product are embodied by its structure and surfaces. In FBS model [24], a product is decomposed into basic function units and structure units, and the mapping from function units to structure units is the product's information model. The structure S can further be decomposed into parts or models with function semantics, which are afterwards decomposed into bodies, faces and curves, shown in figure 1.

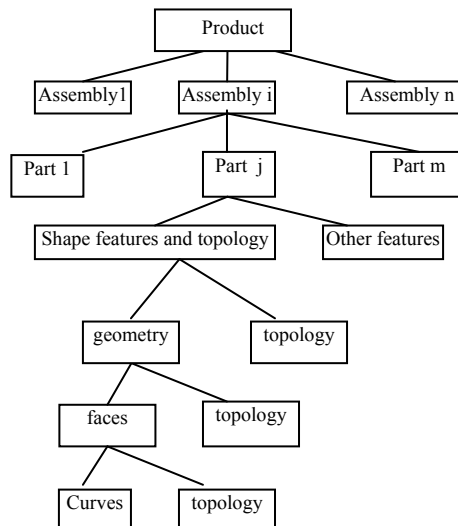


Fig. 1. product decomposition

The suitable decomposition granularity is a geometric feature, which is directly perceived and talked as the designing objective. Otherwise, smaller granularity does not conform to designers' thoughts and customs, so it occurs as communication obstacles; bigger granules is harmful to the diversity of design plans.

A feature as a basic shape unit consists of geometry and semantics set, which is defined as the essential factor of a part by STEP [25]. One engineering feature consists of geometry and topology and semantics information on the surfaces. The geometry is expressed with analytic function or parameter formula, which is mostly a body or a surface. A 3D part consists of one or more features, and a feature consists of one or more geometric shapes.

3.2 3D model gene classifications

Definition1 A feature, namely feature geometry (FG), as the basic construction unit named a feature gene is the geometry or geometry combination with engineering semantics and function-faces.

$$FG = (f(x, y, z, \dots), (\langle a_1, b_1 \rangle \langle a_2, b_2 \rangle \dots), MC) \quad (1)$$

$f(x, y, z, \dots)$ —feature geometry function,
 x, y, z, \dots —parameter such as line, face, point, real.
 $\langle a_i, b_i \rangle \in AXB = \{ \langle a, b \rangle | (a \in A) \wedge (b \in B) \}$,
 $A = \{ a_i \}$ —faces in a feature,
 $B = \{ b_i \} = \{ \text{joint, assembly, boundary} \}$ —function of a face.
 MC —the mass center of a feature.

Though theories on 3D modeling are different, they in fact are based on the same feature geometry functions to produce 3D models of all kinds. The faces and mass center of a feature are the important attributes. Some normal features are shown in table 2.

Table 2. normal feature function types

type	parameter type
cone(cylinder)	point, real
sphere	point, real
torus	point, real
polygon	point, real
Spline-surface	point, real
skin/loft	face, line, real
sweep/revolve	face, line, real
shell	body, face, real

Definition2 Feature topology (TP), as a coordinating gene or topology gene, is the relation among features or the transformation of a feature. The normal topologies are Boolean union, Boolean subtraction, Boolean intersection, transition between features, transformation of a feature, etc.

$$TP = (((FG_i, FG_j) \rightarrow FG_{ij}) \vee (FG_k \rightarrow FG_l)) \quad (2)$$

FG_{ij} —transition geometry between FG_i and FG_j ;
 if $FG_i \cap FG_j \neq \emptyset$, then $FG_{ij} = (FG_i)p(FG_j)$, $p \in \{ \cup, \cap, - \}$;
 $FG_l = p(FG_k)$, $p() \in \{ \text{array, scale, rotate, transfer} \dots \}$.

3D model genome consists of a string of feature genes and topology genes, the set of genes $\{G_i\} = \{FG_i\} \cup \{TP_k\}$, $\{FG_i\}$ —feature gene set, $\{TP_k\}$ —topology gene set.

The roles of genes to make up 3D model are different; then the genes are classified into 5 groups. (1) The crucial features of a part which play the most important roles in 3D model, are called primary features or primary genes (PG) corresponding to BLANK in topology computation [16]. PG developed for long time is firm and can be

inherited by their descendants. (2) The secondary genes (SG) corresponding to TOOL in topology computation are the features such as holes, ribs, etc., which are added to or reduced from primary genes. (3) The auxiliary genes (AG) are transition geometries such as chamfer, blend, etc between genes, which can self-grow according to physics laws on the basis of primary and secondary genes and cannot be expressed by normal functions. (4) equivalent genes. (5) topology genes (TP).

Definition3 Equivalent gene is the feature gene $\rho(u, v, w, t \dots)$, which is still similar to oneself, but its parameters have changed after topology operations σ such as rotation, scale, etc.

$$\sigma(\rho(u, v, w, t \dots)) = \rho(u', v', w', t' \dots) \quad (3)$$

3.3 3D model genome pattern

Extract primary genes series from a group of model instances: $PG_1 PG_2 \dots PG_n$, $i=1, 2, \dots, n$, $PG_i \in \{FG_i\}$. they need to be ranked in local order. PG_i is followed by topology gene TP_k and their secondary genes SG_{ij} , $j=1, 2, \dots, m$, $SG_{ij} \in \{FG_i\}$. Each primary gene can be followed by one or more topology and secondary genes; for example, $PG_i TP_k SG_{ij} SG_{i, j+1}$, $PG_i PG_{i+1} TP_k SG_{ij} SG_{i, j+1}$. So the genotype is set up for computations and the phenotype is for shown on screen after topology operations. The genome pattern is so represented.

$$3D \text{ model genome} = (VF, VT, VS, VA, P, S, G, T) \quad (4)$$

VF –feature gene set $\{FG_i\}$

VT –topology gene set $\{TP_k\}$

VS –secondary gene set $\{SG_{ij}\}$ $\{SG_{ij}\} \subset VF$

VA –auxiliary gene set $\{AG_m\}$

S –starting primary gene series $S \in \{PG_j\}^*$ $\{PG_j\} \subseteq VF$

G –genotype series $G = \{Gh\}^*$, $\{Gh\} \subseteq (\{FG_i\} \cup \{TP_k\})$

T –phenotype series namely terminal codes

P –topology rules, $P \subseteq \{TP_k\}$, normal rules: intersection P1, transition P2, self-transformation P3, array P4.

If $a \in \{PG_j\}$, $b \in \{SG_{ij}\}$, $a \cap b \neq \emptyset$, $p \in P1 = \{\cup, \cap, -\}$, $g \in \{Gh\}^*$, then $g = apb$.

If $a \in \{PG_j\}$, $b \in \{SG_{ij}\} / \{PG_j\}$, $p \in P2 = \{\text{chamfer, blend} \dots\}$, $g \in \{Gh\}^*$, $w \in \{AG_m\}$, then $g = apb$, $w = (apb)$.

If $a \in \{PG_j\}$, $p \in P3 = \{\text{rotate, transfer, scale, transform} \dots\}$, $g \in \{Gh\}^*$, then $g = ap$.

If $a \in \{PG_j\}$, $p \in P4 = \{\text{circular array, array, mirror} \dots\}$, $g \in \{Gh\}^*$, then $g = ap$.

4. 3D model function-coded genome

4.1 API in ACIS

The major current 3D geometry modeling engines such as ACIS integrate frame, surface, body modeling as a kind of identical data structure, which most of commercial 3D modeling software used in industry design, computer animation and engineering design is based on. It defines the boundary which is closed by a set of faces to represent one object. It is incomprehensible to designers to model by choosing the classes of body, face, curve and topology, so application procedure interfaces (API) above these classes are developed. After encapsulating these classes, designers simply use API such as (block (0,0,0),(10,10,10)) to finish the complicated 3D models, which are shown on screen with rendering engines ^[26].

API is the cornerstone of modeling engines, under which is the core of a modeling system and above which are magnificent graphical interfaces. Commercial CAD software gives friendly visual interface to activate the API in order to fit for designers' work. The API on feature and topology is the communication media among designers, customers and products.

Only 3 or 4 modeling engines exist today, but some of API are not identical with each other. The part models produced in different engines need to be consistent while they are put together in one initial population in evolutionary design. The genomes in this paper are set up in ACIS.

In ACIS, geometry API and topology API are the basic feature functions and topology functions which are expressed by the following form:

Feature API= function name (parameter 1, parameter 2, ...) (attribute 1, attribute 2, ...).

Topology API=topology name(feature 1, feature 2,...).

All geometry and topology API functions in ACIS can completely represent the features and topology relations in reality models. Some normal API functions are shown in table 3.

4.2 Genome forming procedure

The function-coded genotype genome need to be standardized for evolutionary design according to the following procedure.

1. the function-coded string consists of primary genes, secondary genes and topology genes extracted from the part modeling database.

$$MS=PG_1TP_1SG_{1j}PG_2\dots PG_iTP_kSG_{ij}\dots PG_n$$

2. MS is operated in certain order.

$$TG=P (SORT (MS))$$

TG—phenotype genome;

SORT—sort operation;

P— topology computation.

Table 3. Feature and topology API in ACIS

	API (api_solid_) in ACIS	Feature / topology in reality
Feature API(FG)	cone()	cone
	sphere()	sphere
	torus()	torus
	polygon()	polygon
	mk_ed_int_ctrlpt	Spline-surface
	skin/loft()	skin/loft
	sweep/revolve()	revolve/sweep
	shell()	shell
Topology API (TP)	apply_transf()	transfer
	scale()	scale
	array()	array
	fix_blends()	chamfer
	unite/intersection/subtract	Boolean operation
	SDM_ds	Transformation

4.3 Algorithms description

1. Extracting primary, secondary, topology genes from model database:

$$MS = MS \oplus f(A_i),$$

$$f(A_i) = \begin{cases} A_i, & \text{if } A_i \in V \\ \emptyset, & \text{if } A_i \notin V \end{cases}$$

\oplus — direct sum

f — extracting function

A — programming sentence set in 3D model $\{A_i\}$

V — $\{FG_i\} \cup \{TP_k\}$

The genes of every model are transported to the genome server synchronously with designing by communication on the cooperative computer plat based on ACIS and MPI.

2. SORT (MS)

Step1 set up new coordinate at the genome mass center MC as the origin, and all genes change their coordinate at mean time, $MC = \text{massprop}(MS)$, $MS = \text{transco}(MS)$.

Step2 local sort: $PG_i \oplus TP_k \oplus SG_{ij} \oplus SG_{i, j+1} \oplus \dots$

Step3 replace the genes with its equivalent genes.

Step4 built minimum box containing the genome, and get its vertex in the space of $-X$, $-Y$, $-Z$ as the base point, $BASE = \text{getp}(\text{box}(MS))$.

Step5 compute the Euclidean distance $d_i(BASE, \text{massprop}(PG_i))$ between the base point and the mass center MC_i of a gene.

Step6 sort MS in the order of d_i .

5. Demonstration

The mobile-phone is modeled by using body and surface as construction elements. In the test of phone design, API functions of body and surface are coded as its genome for evolutionary design, and designers give the fitness of each plan. The evolutionary mechanism is the standard genetic algorithm.

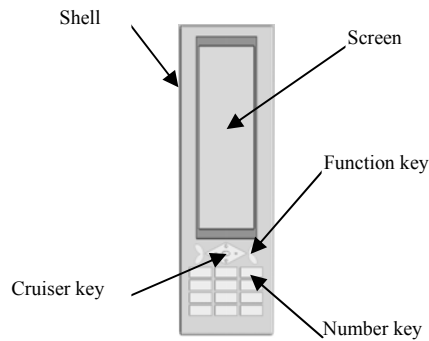


Fig. 2. mobile-phone model structure

See figure 2, the mobile-phone model includes shell, screen, function key, number key, etc. Their gene representation by API is the following:

Shell feature API: `api_brick()`, `api_sweep()`, `api_extrude()`;

topology API: `api_fillet()`, `api_add()`.

Screen feature API: `api_brick()`, `api_extrude()`;

topology API: `api_offset()`, `api_scale()`.

Key feature API: `api_cylinder()`, `api_brick()`, `api_polygon()`;

topology API: `api_array()`, `api_mirror()`...

The mobile-phone evolutionary design system is developed in HOOPS12 and ACIS15. 15 patterns of mobile-phones under the market investigation are used as the initial population to start the mobile-phone evolutionary design. The evolutionary process is seen in figure 3. The operators are single-point crossover, mutation with probability of 0.1 and selection with probability of 0.6. An example in design evolution is shown in figure 4.

A passage of genes from mobile-phone genome:

```
brick(position(0,0,0),position(30,80,10),t1);blend(2,t1);ellipse(position(15,50,8),20,
25,t2);extrude(t2,5);polygon(position(3,3,8),3,t3);extrude(t3,4);
```

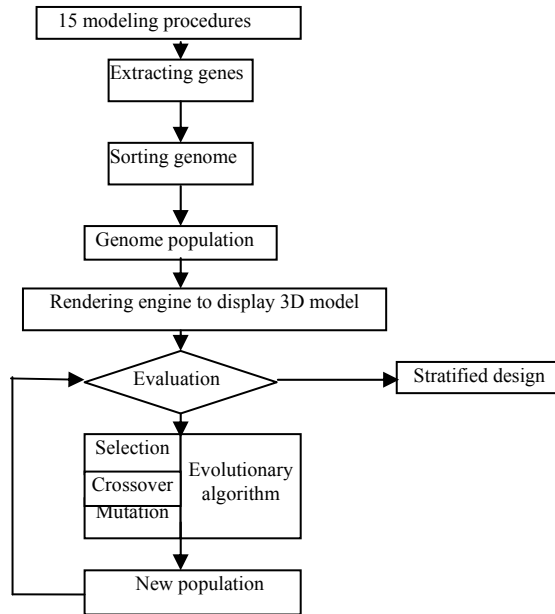


Fig. 3. mobile-phone model evolutionary design process

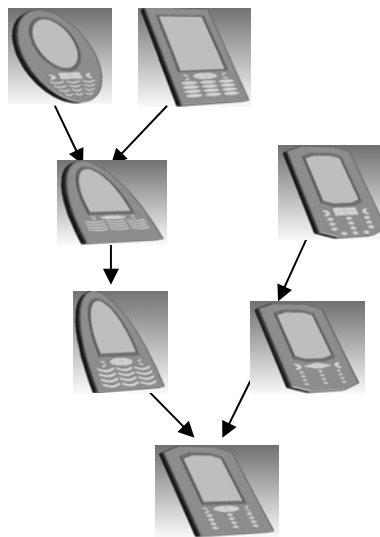


Fig. 4. mobile-phone in evolution

6. Conclusion

Evolutionary design simulates the process of designing a task and optimizes the plans by evolutionary computation. The appropriate genome construction is helpful to design evolution efficiency. The normal and meaningful feature and topology consist of 3D model genome, which is more easily extracted from current CAD database and more easily understood. The genes represented by geometry feature API and topology API in ACIS is concise and efficient to parallel, evolutionary and cooperative design.

But there are many problems to discuss in the next step. It is difficult to get the genomes of completed product CAD models because the current commercial CAD software has not opened to output API; the API from the different modeling engines need to be unified.

Acknowledgements

This project is funded by National Natural Science Foundation of China (No. 69975010, No. 60374054) and supported by Natural Science Foundation of Shandong Province (No. Y2003G01, Z2006G09).

References

1. Bentley, P.: Evolutionary Design by Computers [M]. San Francisco: Morgan Kaufmann, 1-74(1999)
2. Parmee, I.C.: Evolutionary and Adaptive Computing in Engineering Design [M]. London: Springer, 59-202(2001)
3. Goldberg, D.E.: Genetic Algorithms in Search, Optimization and Machine Learning [M]. Boston: Addison-Wesley, 1-55(1989)
4. Russell, S., Robert, E., Ichirou, N., Noriaki, O.: Integrating Engineering Design and Analysis Using a Multi-Representation Approach [J]. Engineering with Computers, 14(2): 93-114(1998)
5. Yongmin Zhong, Mueller Wittig, Weiyin Ma.: A Model Representation for Solid Modeling in a Virtual Reality Environment [C]. Shape Modeling International, proceedings, 17-22 May, Banff, Alta., Canada, 183-190(2002)
6. Horvath, L., Rudas, I.J.: Evaluation of Petri Net Process Model Representation as a Tool of Virtual Manufacturing [C]. Systems, Man, and Cybernetics IEEE International Conference, 11-14 Oct, 178-183(1998)
7. Tallinen, T., Osuna, R.V., Lastra J.L.M., Tuokko, R.: Product Model Representation Concept for the Purpose of Assembly Process Modelling [C]. Assembly and Task Planning Proceedings of the IEEE International Symposium, 10-11 July, 222-227(2003)
8. Pingyi Chao., Shen, Chout Yeh.: Representation of Standard Mechanical Components in a STEP-based Product Data Management System [J]. International Journal of Computer Applications in Technology, 12(1): 16-26(1999)

9. Pasko, V., Adzhiev, A., Sourin, V.: Function Representation in Geometric Modeling: Concepts, Implementation and Applications [J]. *The Visual Computer*, 11(8): 429-446(1995)
10. Case, K., Harun, W.A.W.: Feature-based Representation for Manufacturing Planning [J]. *International Journal of Production Research*, 38(17): 4285-4300(2000)
11. Toni, J., Christian, M., Roland, M.: A Database for the Representation of Assembly Features in Mechanical Products [J]. *International Journal of Computational Geometry & Applications (IJCGA)*, 8(5): 483-507(1998)
12. Robert, B., Kristin, L., Richard, H.: A Heuristic Method for Identifying Modules for Product Architectures [J]. *Design Studies*, 21(1): 5-31(2000)
13. Mohamed, El-M., Allen, M.: Database System of Mechanical Components Based on Geometric and Topological Similarity, Part I: Representation [J]. *Computer-Aided Design*, 35(1): 83-94(2003)
14. Gero, JS, Sosa, R.: Complexity Measures as a Basis for Mass Customization of Novel Designs [J]. *Environment and Planning B: Planning and Design (S0265-8135)*, 35(1): 3-15(2008)
15. Rosenman, MA, Gero, JS.: Evolving Designs by Generating Useful Complex Gene Structures [M]. P. Bentley (ed.), *Evolutionary Design by Computers*, London : Morgan Kaufmann , 345-364(1999)
16. LI, Hongjie., XIAO, Renbin.: Genetic Model in Evolutionary Design of Complicated Product Based on Function Construction [J]. *Chinese Journal of Mechanical Engineering*, 39(5), 42-48(2003)
17. Soddu, C.: Recognizability of the Idea: the Evolutionary Process of Argenia [M]. Bentley P (ed.), *Creative evolutionary systems*. San Francisco: Morgan Kaufmann, 109-127(2001)
18. Bentley, P., Kumar, S.: Three Ways to Grow Designs: a Comparison of Embryogenies for an Evolutionary Design Problem [C]. *Genetic and Evolutionary Computation Conference (GECCO '99)*, 14-17 July, Orlando, Florida USA, RN/99/2, 35-43(1999)
19. Bentley, P., Jonathan, P.: Generic Representation of Solid-Object Geometry for Genetic Search [J]. *Microcomputers in Civil Engineering(S0885-9507)*, 11(3), 154-163(1996)
20. Chen, Hongwu., Huang, Kezheng.: On Growth Design Based on Mechanical Product Gene [J]. *Mechanical Science and Technology*, 24(4): 447-450(2005)
21. Liu Hong., Tang, Mingxi.: Evolutionary Design in a Multi-agent Design Environment[J]. *Applied Soft Computing Journal(S1568-4946)*, 6(2): 207-220(2006)
22. Liu Hong., Tang, Mingxi., John Frazer.: Supporting Creative Design in a Visual Evolutionary Computing Environment[J]. *International Journal of Advances in Engineering Software(S0965-9978)*, 35(5): 261-271(2004)
23. Liu Hong., Tang, Mingxi., John Frazer.: Supporting Dynamic Management in a Multi-agent Collaborative Design System[J]. *International Journal of Advances in Engineering Software(S0965-9978)*, 35(8-9): 493-502(2004)
24. Gero, J.S.: Design Prototypes: a Knowledge Representation Schema for Design [J]. *AI Magazine(S0738-4602)*, 11(4): 26-36(1990)

25. Peng, Yinghong., Hu Jie.: KBE and Application in Product Design[M]. Shanghai: Shanghai Transportation University Publisher, 94-140(2007)
26. Zhan, Haisheng., LI, Guangxin., Ma, Zhixin.: Geometry Modeling Technology and System Development Based on ACIS [M]. Beijing: Tsinghua University Publisher, 107-116(2002)