CUI JIA

Design Transformations Represented by Shape Grammars for Conceptual Generative Design

(Paper and Installation)

Topic: (Computer-aided Design)

Authors:
CUI JIA
Shandong Normal University, School of Information Science & Engineering
China
http://www.sdnu.edu.cn/
TANG MING Xi
The Hong Kong Polytechnic University,
School of Design
Hong Kong
http://www.sd.polyu.edu.hk/en/
WANG JI HUA
Shandong Normal University, School of Information Science & Engineering
China

Main References:

[1] Soddu, C. Generative Art Geometry. Logical interpretations for Generative Algorithms. Generative art. Rome, Italy.2014

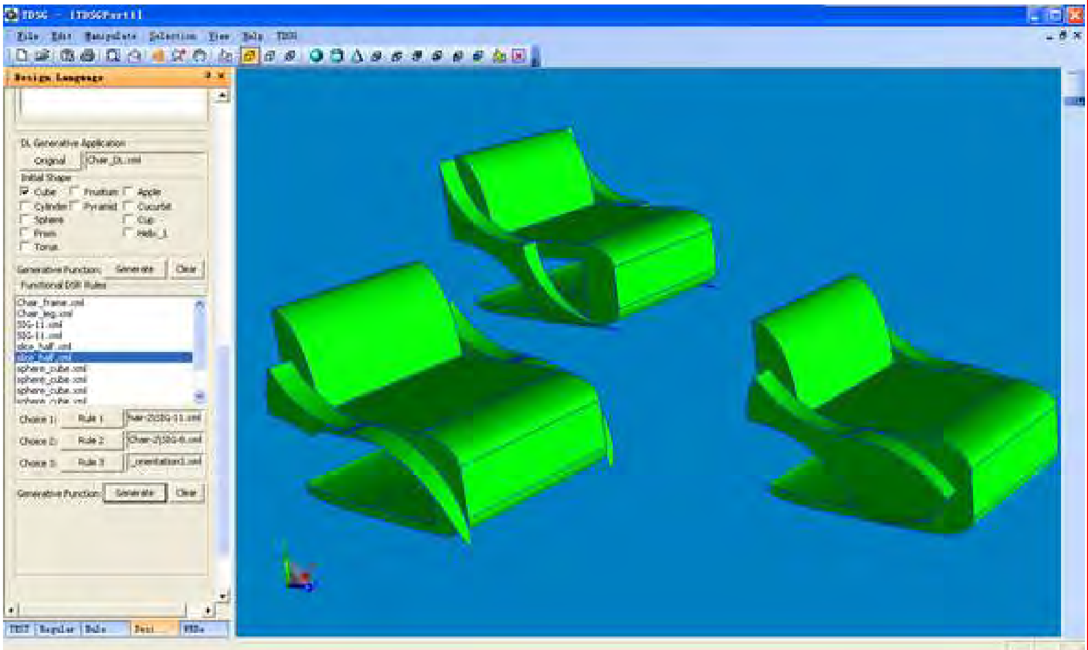
[2] Tang, M. X. and J. Cui "Supporting product innovation using 3D shape grammars in a generative design framework." International Journal of Design Engineering **5**(3): 193-210. 2014

Abstract:

Design solutions are specific decisions made by designers among various potential choices which are parallel in huge design space including shape, colour, material, spatial structure and functions etc. Design transformations are tools to evolve design towards to the desired status in whole design process. The generative mechanism can be considered as the tool for managing dynamic processes of transformation by evolving design ideas along logic way from existed to unknown possible in future.

In this paper, we use the procedural modelling approach—shape grammars to represent dynamic design transformations in terms of design rules for conceptual design. In this way, design rules including transforming information can be used in different design situation for generative exploration, which is different from the traditional application of shape rules. The geometric addiction of rule application can be released. Design languages in shape grammars, similar to the design process in real design work, is a special operational sequence to generate design alternatives. They support the multiple-reading properties even for the same design object; as a result, more creative alternatives can be explored by generative approach subsequently. As a preliminary prototype, the selected furniture design is chosen as examples to testify the feasibility.

A computer installation will be presented in GA2015.



Snapshot of DSG Interpreter

Contact: email
cuijia1247@163.com

Keywords: Shape grammars, Generative design, Shape rules, Design transformation

Design Transformations Represented by Shape Grammars for Conceptual Generative Design

CUI JIA

School of Information Science & Engineering, Shandong Normal University, China

Centre for Visual Computing and Translational Informatics, China

TANG MING-XI

School of Design, the Hong Kong Polytechnic University, Hong Kong

WANG JI-HUA

School of Information Science & Engineering, Shandong Normal University, China

Design solutions are specific decisions made by designers among various potential choices which are parallel in huge design space including shape, colour, material, spatial structure and functions etc. Design transformations are tools to evolve design towards to the desired status in whole design process. The generative mechanism can be considered as the tool for managing dynamic processes of transformation by evolving design ideas along logic way from existed to unknown possible in future.

In this paper, we use the procedural modelling approach—shape grammars to represent dynamic design transformations in terms of design rules for conceptual design. In this way, design rules including transforming information can be used in different design situation for generative exploration, which is different from the traditional application of shape rules. The geometric addiction of rule application can be released. Design languages in shape grammars, similar to the design process in real design work, is a special operational sequence to generate design alternatives. They support the multiple-reading properties even for the same design object; as a result, more creative alternatives can be explored by generative approach subsequently. As a preliminary prototype, the selected furniture design is chosen as examples to testify the feasibility.

1. Design representation

People (and different designers) have different impressions when they evaluate same design products. In other words, multiple readings in design, especially in visual design, play an important role in the conceptual creative phase [1]. Aesthetic diversity provides designers opportunities to conceive different interpretations of design representations.

This diversity will cause problems for shape representation in CAD systems which require uniqueness. Design solutions are usually represented by a decomposition process in which a product is presented in terms of geometric and topological variables (the parametric design method). From another perspective, when a designer is dealing with a specific task, s/he draws experience from more than one domain [2]. For example, a fashion designer may find inspirations from his/her past experiences of building design projects or furniture design projects. The design knowledge heritage and experience accumulation may not be derived from the same domain as that of the current working project. Correspondingly, design representation should be comprehensive and complex so as to capture the design knowledge and imagination from different domains.

2. Design exploration

Design exploration is used to find optimal and satisfactory design alternatives in design spaces. AI strategies, i.e. generative system, are widely used in this area. Design knowledge is central to design exploration, whether automatic or manual. Knowledge-based engineering (KBE) is at the core place of diverse fundamental disciplines, such as AI, CAD and CP (computer programming), which perfectly match the broad and heterogeneous fields of design [3] including psychology, philosophy, engineering, business management and fine arts. Design knowledge capture is directly related to the quality of the design exploration.

Predefined knowledge is widely used in CAD research in forms of variables, formulations and symbolic strings. Knowledge is extracted from design sources before design exploration starts. The major design information is predefined, typically following a sequence; however, designers might not do this? In practice, they do not commence with an analysis of important aspects of given design problem, and then synthesize the solutions based on this analysis, and finally choose the structural way to represent the solution [4]. Designers do not go through the entire process in a clear cycle. Instead they simultaneously process many channels of information.

Rules and frames are the two most common forms of knowledge representation [5, 6]. To support flexible smart design exploration, rules and frames should be created and applied to the design process. This means that design knowledge should be collected during the design process, not in advance. Other information, besides geometric and topological information, should be recorded. For modelling natural design actions, knowledge acquisition and applications are best when combined. Better system-user communication in design exploration is required so as to initiate more powerful computer support.

3. Dynamic design shape representation

Based on our previous work[7, 8], design transformations are used to represent design actions in terms of design rules. In this way, the dynamic shape representation can represent design shapes on both static way (geometric and topology) and dynamic way (design procedures).

Component technology reduces computational difficulties for complex shape representation. Through decomposition, shapes can be represented by some low-level elements, even decreasing the dimensional property from 3D to 0D (points) [9]. For the consideration of less limitation on shape creation, we decided to employ component technology in the new our representation. The primitives (cube and sphere) are atoms to represent a DSR (dynamic shape representation, DSR) shape. For better capturing design transformation, the DSR shape uses a series of basic shape transforms, named Elemental Rules (ER), to represent fundamental design actions. Therefore, in a DSR shape, the components are not based on the primitives, but the sequential application of ERs. The ERs are only used to construct a shape using permitted primitives. Therefore, the ERs are at a mid-level between the shape representation and solid primitives. There are 9 ER families (Fig 1), 70 different single ERs for shape construction. Although it is still not enough to satisfy all shape generations, the 9 ER families can support many different shapes including free-solid forms. All the ERs are not only used as rules to transform design, but also the approach to change design objects.

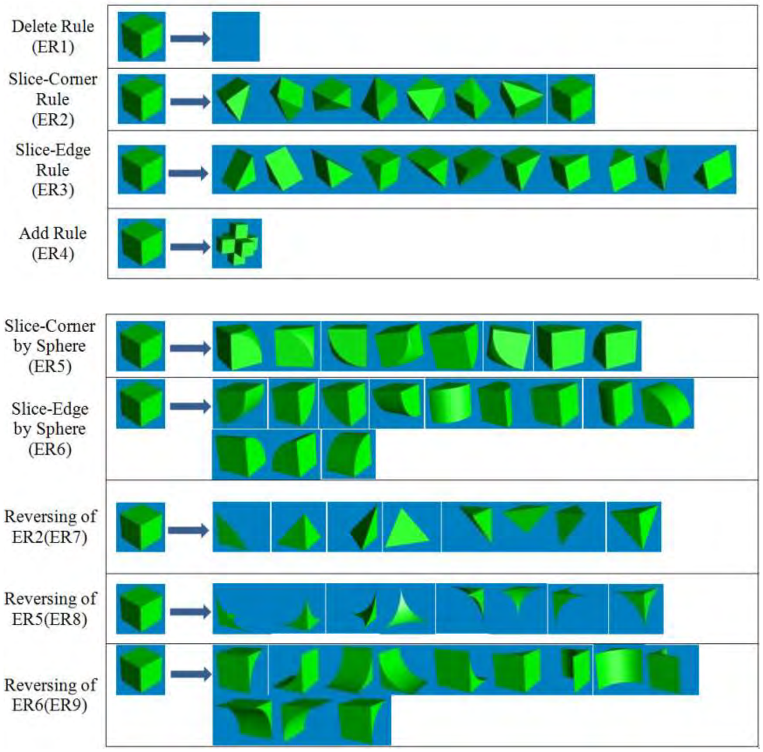


Figure 1. Elemental Rule Families

Definition: DSR shape. DSR shape is a finite set of primitives, which are manipulated through a family of Elemental Rules in a specific order. A DSR shape can be formally represented by the primitive set and the family of ERs following the operational sequence, {S*|ER (i1), ER (i2), … , ER (in)}.

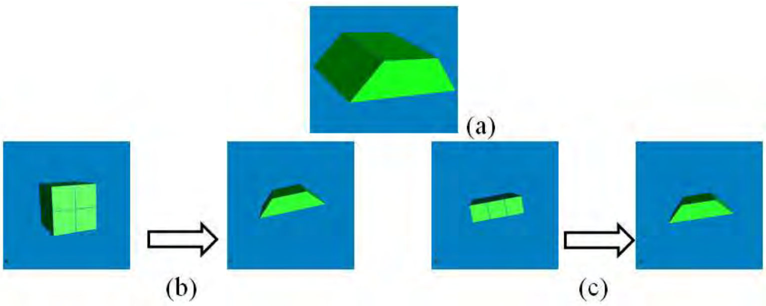
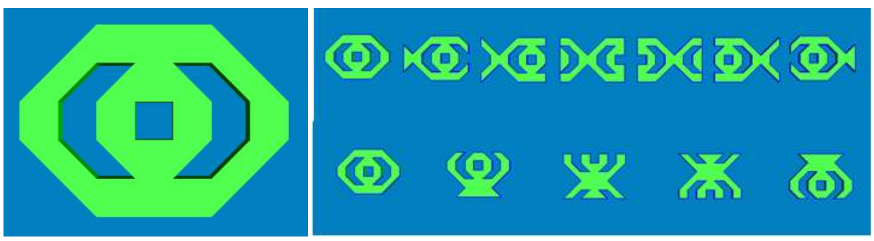


Figure 2. Multi-representation of DSR shape

There are two benefits to use the DSR shape. One is the multi-representation for one same shape. In Fig 2.a is a rhombus shape. The different generating process can be represented by DSR shapes, such as Fig 2.b and Fig 2.c. In Fig 2.b, for example, the formal DSR shape representation is {S1 | ER(3), ER(3), ER(3), ER(3), ER(3), ER(1), ER(1)}. The Fig 2.c can be formally represented as {S2 | ER(3), ER(3), ER(3), ER(3), ER(3)}. The two DSR shapes are the same from their final status (a rhombus shape). However, the way of generating is different. Another benefit is the convenient generative transform from existed one to new ones. In Fig 3, the same shape (a) can be easily transformed to different shapes (Fig 3.b) by changing the position of every primitive.



(a) (b)

Figure 3. Generative transform of DSR shape

The shape representation for conceptual design creation needs to satisfy two requirements: shape ambiguity [9] and shape emergency [10]. The former one represents the multi-reading of design. Different designers have different feelings when they a design. Therefore, the design representation should have enough flexibility to describe the same design. The example shown in Fig 2 proves the shape ambiguity on the DSR shape. The Fig 3 shows the changeable ability on shape representation. One single shape does not mean too much for design ideation. The changeable shape can inspire designers at conceptual design stage. When a DSR shape is generated, various new shape generations show the shape emergent features of DSR shapes.

4. Rule-based 3D shape generation system

The DSR shape interpreter is a general three dimensional CAD system. It is designed as a modular structure for better functional extension and maintenance. There are seven main modular which concentrate on different functional aims: UI commander, Design actioner, Interpreter, Shape Engine, ER Pool, Shape Modeller and Data centre shown in Fig 4.

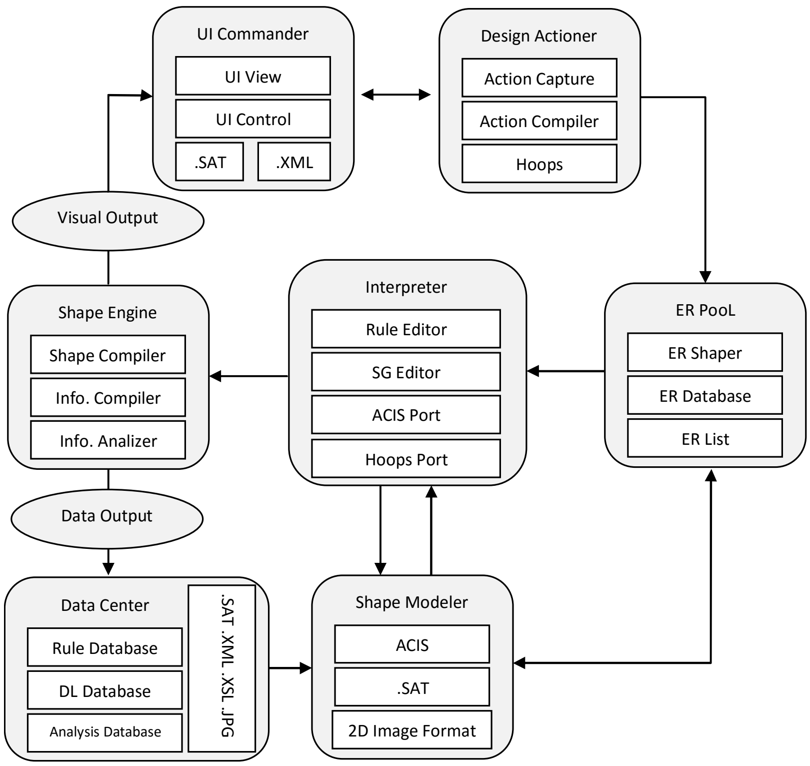


Figure 4. Modular structure

There are three control panels in the DSR shape interpreter, the DSR rule creation panel, the design generation panel and the design language panel. The system works in 3D environment. All generations are 3D entities, which support the directly visual response to designers. The interface is shown in Fig 5. There is a three-dimensional orientation arrow to indicate the current working 3D view at the lower left hand side of the screen. In the working environment, there is a tool bar at the top of the window. These tool functions may help users to adjust the 3D views and drag, move the entities and zoom in/zoom out the objects. At the left down corner, there is a ‘switch tab’ which is used to switch the three control panels.

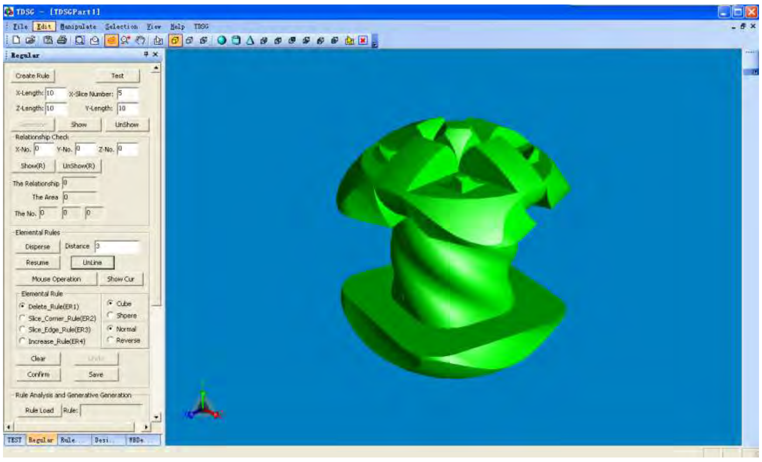


Figure 5. Snapshot of Interface

The DSR rule creation panel takes charge of the DSR rule generation by the ERs. Through mouse movements, the 9 ER families can be used to model the initial union of primitives to the desired form. The ‘Un-do’ operation is supported. During DSR rule creation process, when the final shape is finished, the ‘Confirm’ button will be used to add the finish mark in the DSR shape, and the ‘Save’ button will be used to call a new window to save the object to specified place in hard-disk.

The Design generation panel is used to create design solutions by the rule based system. There are three kinds of rules working in this panel, the layout rule, functional rule and auxiliary rule, which will be illustrated in next chapter. There is a DSR rule database. Users can choose their designed DSR rules to work in their design process. Through ‘update’ button, user can see the newest DSR rules in the rule database.

For the panel of design language, there are two parts of functions. The first one is the language reading function, which is used to read the existed design language and re-generate the whole process. The manually step-by-step way and automatically one-time way are supported for design demonstration. Another is the generative application function of design language, which is used to read the selected design language, and re-create more novel solutions for design exploration.

Parameter modelling technology becomes prevailing in current academic areas because of its computing compatibilities. In design, designers, even the new generation, still cannot control the number-driven methods easily. Their preferences on sketching and hand drawing are due to the vision-driven working habits. Therefore, as mentioned before, if a system permits users to design their shapes in a visual environment and generate the parameters unconsciously, then it will be a more suitable approach to the visual parameter modelling. My rule-based generative system supports the validation of this thinking. In our system, users can generate their designs by mouse and keyboard events under the vision involvement, simultaneously; DSR shape interpreter translates the visual operations into the data information in terms of parameters (Fig 6). By the recursion of rules, these parameters can guide the automate generation for better design effectiveness.

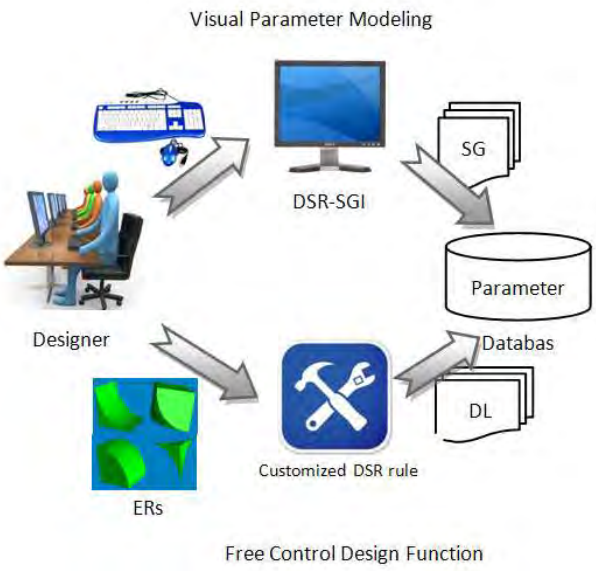


Figure 6. Working flow

The visual parametric modelling technology is not conflicted with the current parametric design. The difference is how to generate parameters. For traditional parametric design, the parametric representation is the formal representation on a design system. Different values demonstrate different results. However, users/designers can only change the values but not the parameters, which means that if a new design object needs to be generated, then the re-parameterized process is necessary. For the visual parametric modelling, the parameters are generated based on the visual operations which are captured from users/designers. Different design actions will generate different parameters. The designers understand the parameterized process as the visual procedures. In this way, more designers can get involved in the design formalization process. As a result, the parameters can load more design information than in the traditional way.

Another restriction of modelling system is the predefined functions. There are some improvements which have been done in this research. DSR rules are generated by the sequential applications of ERs which are predefined as the elemental components. The main body of the running mechanism—the DSR rules are generated by users themselves. System users can generate the DSR rules by the ERs and 'UI Commander', and then save them into the 'Data Centre'. There is no limitation on the DSR rule creation which means users can design their desired rules as tools applied in their design process. In a conceptual design process, users can make use of the rules in the database to shape their designs. The real-time rule creation in design process is supported.

Generative design is helpful for creating more alternative to stimulate designers' inspiration and save time during design process. There are two kinds of generative design mechanisms in computers: the automatic process and semi-automatic one. No matter which one is used, the basic computation always involves the use of parameters. For the automatic form, some AI technologies and pre-defined constraints can be used to limit the design generations towards to a preferred direction. The evaluating criteria are generated from the parameterization process of design knowledge by both designers and design researchers. For the semi-automatic form, user interactions are necessary to select the interested ones from solution space. Therefore, it can be concluded that the designers' involvement is necessary for both kinds of generative design process.

5. Experiment

There are two experiments to show the feasibility of the dynamic shape representation: conceptual cup case and generative chair case.

In the first case, designers create relevant DSR rules based on the DSR shape in our system. The single ER used in the DSR shape is so elemental that can be captured by common computer operations, such as the keyboard and mouse events. Actually, after finishing shape transformations, the atom operations, ERs, are recorded automatically by computer for generating new DSR rules. There are three DSR rules for a cup design: cup-frame rule, cup-handler rule, handle-refine rule, as shown in Fig 7.

The DSR shape and DSR rule represent single design actions in design process. The consecutive design actions can describe a complete design idea of designers. By the initial shapes and the set of DSR rules, the design languages which capture the design procedures can be generated. Design languages are helpful for understanding design processes by designers as the visual demonstration. At the same time, the grammatical rule-based approach permits the possible for computer to analyse the logic of design proceeding.

When the design languages are generated by DSR shapes & DSR rules, the design procedures can be represented and captured step by step. It supports an avenue to test the effects of the same language working on other initial shapes or design environment. The design knowledge inside the recorded information leads the design automation towards the direction represented by design languages. There is a simple example of an application of design language on conceptual cup design, given below.

At first, designers create relevant DSR rules based on the DSR shape in an interactive computer system. The single ER used in the DSR shape is so elemental that it can be captured by common computer operations, such as the keyboard and mouse events. Actually, after finishing the shape transformation visually, the atom operations, ERs, are recorded automatically by computer for generating new DSR rules. There are three DSR rules for a cup design: cup-frame rule, cup-handler rule, handle-refine rule, as shown in Fig 7.

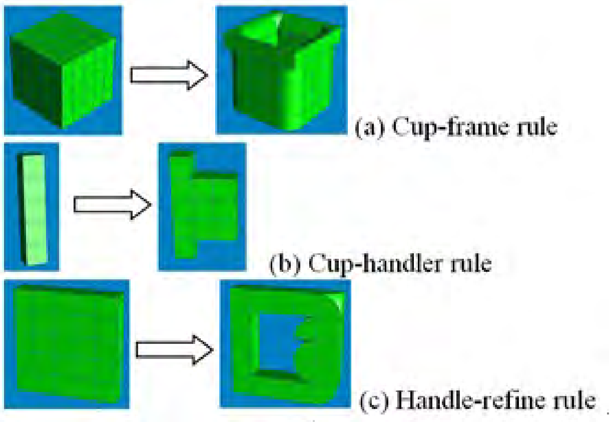


Figure 7. Three DSR rules for cup design

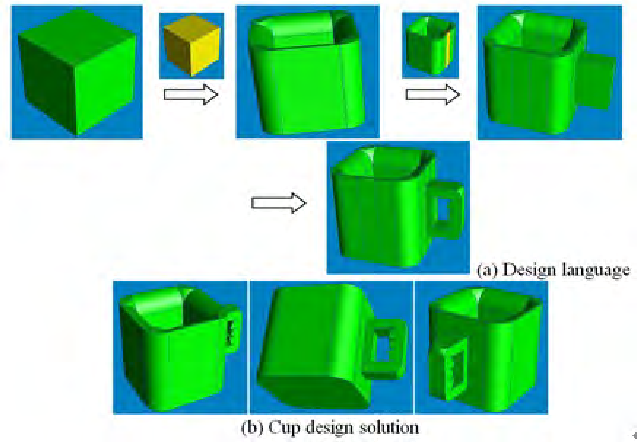


Figure 8.Cup design language

The design language is shown in Fig 8.a. The parts marked yellow are the object shape or sub-shape for the next rule application. When the initial shape is a cube, the design solution is shown in Fig 8.b from different views. As mentioned above, the intended design actions are the priorities for the application of DSR rules. Therefore, when getting a design language by DSR shape, not only the design solution itself, but also the design actions are saved. If applying the design language to other different initial shapes, some novel solutions can be generated. Meanwhile, the desired design actions are kept in the new solutions. This can enhance the functions of the design language. When having one design language by DSR shape, the more alternatives can be generated within a generative system.

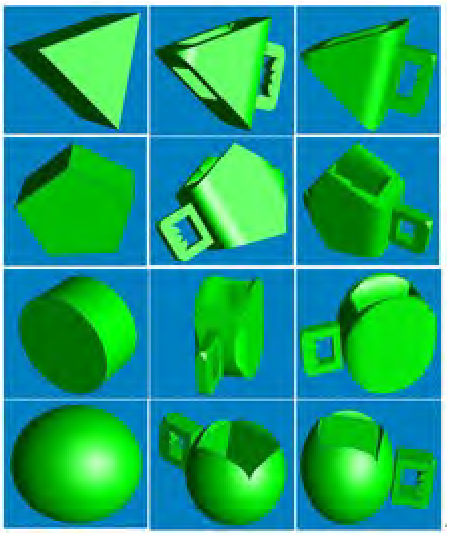
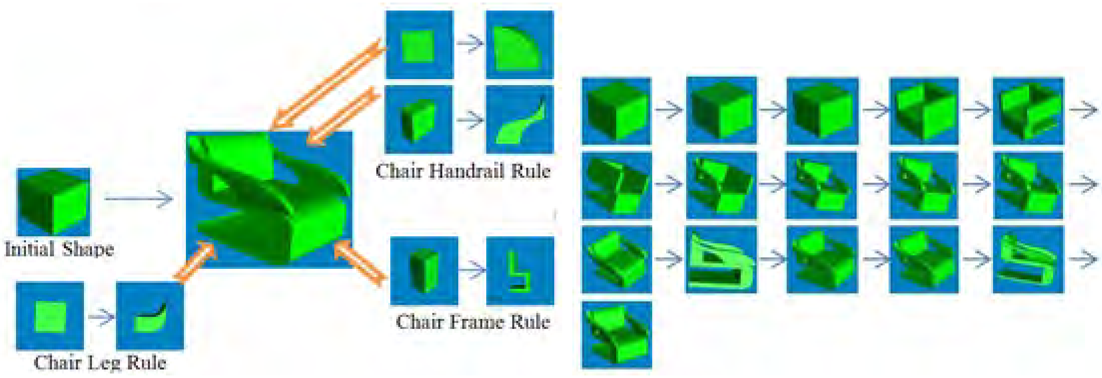


Figure 9. Generative results of cup design case

In the second case, a chair is created in our system, shown in Fig 10.a. There are four DSR Rules used to create this chair: two 'Chair handrail Rule', 'Chair Frame Rule' and 'Chair Leg Rule'. The design language is shown in Fig 10.b.



(a) (b)

Figure 10. Conceptual chair design case

We invited several candidates to re-build the chair in system. As the multi-reading, different candidate generate the same chair in different ways. There are three design languages are chosen here (Fig 11).

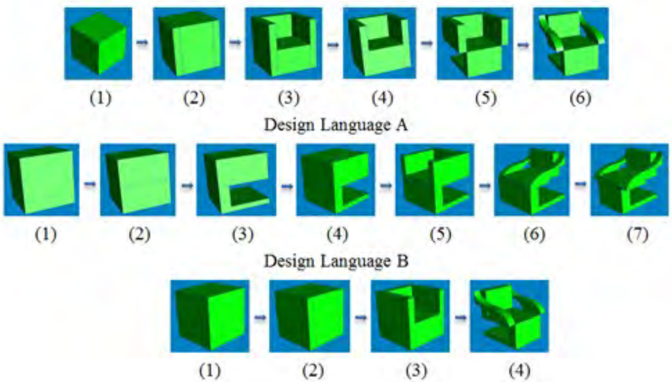
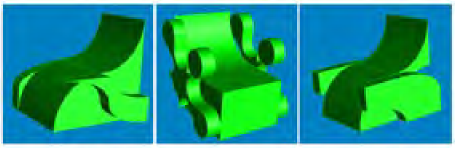


Figure 11.Three design language by three candidates

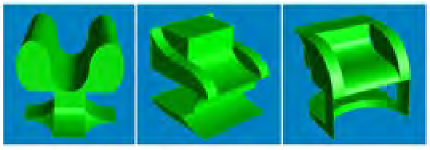
In Fig 11, there were 6 steps in Design Language A. The step 2 and 4 used layout rules. The former one divided the whole cube into three parts in the order of (handler, body, handler), and the latter one divided the two handlers into two parts (upside and downside). The step 3, 5 and 6 involved DSR rules which were used to change the shape of the chair.

In Design language B, there were 7 steps, in which step 2 and step 4 involved layout rules, whilst the others were DSR rules. In Design language C, there were 4 steps and only step 2 used the layout rules, which divided the cube into three parts. For the three parts, three times of using DSR rules generated the final 3D chair model.

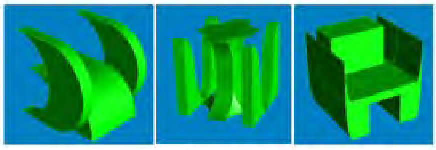
After receiving the three design languages, we used them in the generative design system. In a short time, three design families were created, shown in Fig 12.



(a). New generations based on Design language A



(b). New generations based on design language B



(c). New generations based on design language C

Figure 12.Generative results from different design languages

6. Conclusion and discussion

In this work, design transformations can be represented by shapes and rules in terms of dynamic shape representation. Shape rules can represent basic shape transformations which are dynamic information in design process. In this way, the procedure knowledge can be captured by generative mechanism which supports more flexible and rich approach to generatively create more novel design alternatives related to the existed design knowledge.

Currently, the system can only capture basic shape transformations, such as add, delete and slice. More meaningful design transformations should be considered in future for better understanding to semantic information of design process. As the Boolean operations are widely used in shape modelling operations, the effectiveness of shape generating should be improved further.

Acknowledgement

This paper is sponsored by NSFC project (61472233, 61502285, 61572300), Shandong Natural Science Foundation (ZR2014FQ013).

References

1. Gross, M.D., Emergence in a recognition based drawing interface. Visual and Spatial Reasoning II. BTJ Gero, T. Purcell. Sydney Australia, Key Centre for Design Cognition and Computing, 2001: p. 51-65.
2. Schon, D.A. and G. Wiggins, Kinds of seeing and their functions in designing. Design studies, 1992. 13(2): p. 135-156.
3. Rocca, G.L., Knowledge based engineering: Between AI and CAD. Review of a language based technology to support engineering design. Advanced Engineering Informatics, 2012. 26(2): p. 159-179.
4. Suwa, M., J. Gero, and T. Purcell, Unexpected discoveries and S-invention of design requirements: important vehicles for a design process. Design studies, 2000. 21(6): p. 539-567.
5. Milton, N.R., Knowledge technologies. Vol. 3. 2008: Polimetrica sas, Monza, IT.
6. Negnevitsky, M., Artificial intelligence: a guide to intelligent systems. second ed. 2005: Addison-Wesley, Harlow, England.

7. Tang, M.X. and J. Cui, Supporting product innovation using 3D shape grammars in a generative design framework. International Journal of Design Engineering, 2014. 5(3): p. 193-210.
8. Cui, J. and M.X. Tang, Representing 3D Shape Grammars in a Generative Product Design System, in FIFTH INTERNATIONAL CONFERENCE ON DESIGN COMPUTING AND COGNITION (DCC'12 OR DCC12) J. Gero, Editor. 2012: College Station, Texas USA.
9. Knight, T., Computing with ambiguity. Environment and Planning B, 2003. 30(2): p. 165-180.
10. Knight, T., Computing with emergence. Environment and Planning B, 2003. 30(1): p. 125-156.