Multi-Agent Feature based Shape Grammar Implementation for Concept Generation of Industrial Product Design

Sitaram Soni[1], Pritee Khanna[2] and Puneet Tandon[3]

[1]PDPM IIITDM, sonis@iiitdm.in
[2]PDPM IIITDM, pkhanna@iiitdm.in
[3]PDPM IIITDM, ptandon@iiitdm.in

ABSTRACT

The paper reports an agent based framework for the conceptual design generation stage of the Industrial product design process. The conceptual design is characterized by the multiple design generation and evaluation of the product concepts. The framework simulates the design team paradigm. Agent based architecture offers the advantage of being flexible, extendable and distributed. Added multiplicity of the shapes is achieved through the parametric variation and morphing of the feature elements. The implementation is demonstrated using a running example of a mobile phone, as it is simple enough to demonstrate the concept as well as complex enough to incorporate the team based paradigm of the design environment.
The research is an effort to develop a design system to support the automatic concept generation and evaluation for the mass customization of industrial products.

Keywords: style, grammar based design, agents, conceptual design, morphing.
DOI: 10.3722/cadaps.2010.797-807

1 INTRODUCTION

The industrial design is characterized by the rapid generation of products to cater the changing taste of the market and individual preferences. As the industrial product market is moving from the technology to aesthetics and ergonomics driven purchases, there is a pressing need to have a computational system for the quick generation and evaluation of the product concepts to reduce the cost and time leads. The market needs have out grown the capacity of the human designers to provide the innovative solution at a regular and sustainable basis. The automated concept design generation systems promise a solution to this problem.

The current CAD systems are excellent in description and subsequent treatment of the product designs but they lack gravely to support the concept generation process of the industrial products. This phase of the design process is still a largely human centric activity. The present work is a step in the direction to implement such a concept generation system for the industrial products.

The paper presents a model framework based on the feature based shape grammar implementation using parametric variation and morphing to expand the design space represented by the model. The design space is explored and constrained by the style sketches representing the intention of the designer.

The concept generation is an important activity in the design process [14]. The current market scenario demands fast and consistent generation of the product form to cater the changing market needs. The human capabilities in generating the alternative product concepts are limited and time consuming. This creates a pressing need for the computational support to expedite the concept generation and evaluation process. The generated alternatives need to conform to consumer/ designer preferences. This paper presents a style guided design framework for the concept generation using multi-agent paradigm with shape grammar and morphing.

The paper begins with the introduction and the motivation for the present work. Section 2 describes various technologies and domains involved in the concept generation process like style sketches, shape grammar, the agent system and morphing process. These technologies are used in the proposed framework for the concept generation. Section 3 describes the proposed framework involving the configuration map, agent hierarchy and the morphing process. The optimization process and the three dimension form generation are also discussed in this section. Section 4 discusses the implementation details of the framework for the test case of the mobile phone design. It also details the modes of operation possible for the design. Section 5 shows some preliminary results obtained in this exercise. Section 6 of the paper presents the discussion and the conclusions.

2 STYLE FORM GENERATION

Sketches are integral part of the concept design. They externalize the design solution conceived by the designers and also support the revision and refinement of the ideas. They are helpful in generating concepts, and facilitate problem solving [1]. The style is the most visible and the striking visual element of any object. Every object has a form. The form is composed of characteristic features. The shapes of these features define the product style [9].

In the conceptual design phase, there are four stages of the design process, starting with concrete experience. It is followed by observations and reflections, which leads abstraction, generalizations and testing the implications of concepts in new situations. The whole process is realized through the medium of sketches. This is a circular process in which, the activities are repeated until an optimum design solution is found [15].

Morphing is the process of interpolating the elements between two sets to generate a new element. The morphing has been extensively used in the field of computer graphics and animation to morph between two images or surface forms, but its use in the product design has been limited. Shape morphing methods are explored by Hsiao and various authors [4-6] for the semantic driven design of the industrial products. Chen and Parent [2] used morphing to generate a series of extended test samples to be used in image evaluation experiments. The use of the morphing in the shapes rules is presented as under:

Assume a new shape rule S_{new} is obtained by morphing two different shapes rules S_A and S_B having n corresponding elements, such that there is a correspondence between the right sides of the shape rule S_A and S_B , and both the rules have the same left hand side.

$S_A = \{\alpha_i\}$ and $S_B = \{\beta_i\}$ where α_i and β_i are the i^{th} element in shape S_A and S_B respectively.

The new shape rule can be expressed mathematically as follows:

$S_{new} = S_A \circ S_B = \{\alpha_i \circ \beta_i\}$, $i = 1,\ldots,n$; The symbol $\circ$ represents the morphing operation, which can be defined as:

$$\alpha_i \circ \beta_i = (1-\rho_i)\cdot\alpha_i + \rho_i \cdot \beta_i \qquad (2.1)$$

Here ρ_i is the element-wise percentage of morphing.

Since the left side of the shape morphing rules is same, this operation is to be applied to the right side of the shape rules.

The concept of a shape grammar is that a set of rules is constructed to capture specific domain knowledge about a certain type of artifact. The development of these rules encapsulate a set a valid operations that can occur in the development of a design. This technique creates a formal language for generating and updating complex designs from a simple initial specification, or seed.

Shape grammars have proved to be applicable in a range of design domains, which sets them as an appropriate technique for product design. They employ a generative approach to create a design using match and replace operations described by a set of rules for a particular domain [16].

An agent is defined as a computational element in a distributed, modular and object oriented environment, and is capable of flexible and autonomous action, in order to meets its design objectives [17]. Over the past decade, the agent based technology has attracted significant attention and has seen many applications in research. The flexibility of design and implementation distinguishes an agent from any other architecture. An agent is characterized by exhibiting the following characteristics:

- Responsive : the agent perceives changes in its environment and reacts appropriately
- Proactive : the agent exhibits directed behavior, and
- Networked: the agent communicates with other agents [17].

Multi-agent system represents one of the most promising technological paradigms for the development of open, distributed, cooperative, and intelligent software systems [7]. Multi-agent systems are systems with multiple agents and usually involve distributed knowledge sources. These multi-agent systems may be designed with different kinds of strategic behaviors, governing their interaction, such as communication, cooperation, and conflict resolution. Agents are generally viewed as a higher level of abstraction than object-orientation to develop complex and robust applications. In the domain of the product design process, Madhusudan [10] presented an agent based framework for autonomous coordination of distributed design process management using a workflow-based principle. Product information sharing is critical for effective design collaboration. Hao et al. used the agent based system for the product design engineering [3].

3 THE FRAMEWORK

The framework of the model adopted in this research is a multi-agent shape grammar based concept generation system that governs the design space by a set of grammar rules. The design variations are generated through parametric design and morphing.

The framework provides an iterative loop based pipeline structure for the implementation of the model. Fig. 1 presents the framework of the model. The model has four major divisions, namely:

- Style form generation
- Two Dimensional (2D) design
- Optimization
- Three Dimensional (3D) design

Each division comprised of various modules. These modules are discussed as under.

3.1 The Configuration Map and the Agent System

The configuration map represents the structural description of the products features [3]. These features may be driven by the function, ergonomics or the aesthetic considerations. The features of the products can be represented in a hierarchical form, based on their relationship with each other, as is generally represented in any CAD system. This hierarchy governs the order of addition of the features. This hierarchy may change from designer to designer, as different designers may follow different paths to add features in the product. The concept of features represented here connotes the entities in the products, which are decomposed based on some characteristics, like the function or aesthetics. The configuration map can be represented in the form of a network of nodes and edges. The nodes represent the features and the edges represent the characteristics shared among the features. For example, a car can be decomposed into functional features like body, tires, suspension, engine and the likes.

The agent system corresponds to the configuration map. The configuration map governs the agents system, their domain and the communication protocol among the agents. The configuration map is used to generate the skeletal form of product as well as the agent hierarchy also.

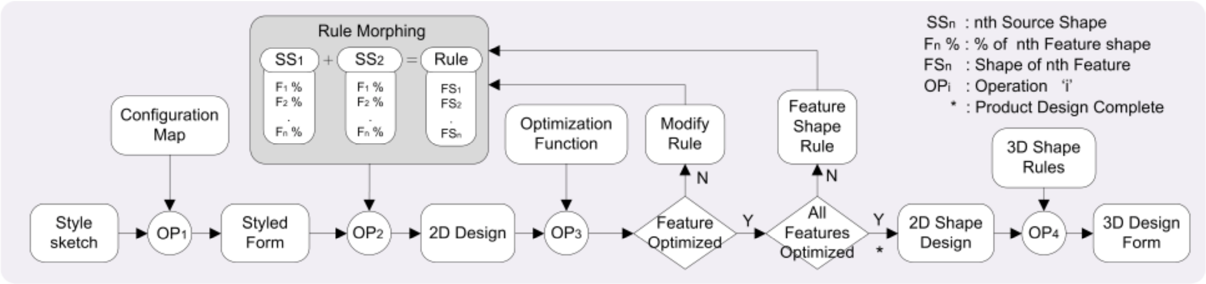

Fig. 1: The framework model.

3.2 Style Sketch

As the designers generally represent the characteristic features of the product form in terms of the sketches, the style sketch represents the designer's intention to achieve the form of the product conforming to certain preferences. These preferences act as the search guidance and delimiter in the design space of the product. In the present framework, the sketch is an input by the user. The sketch is used to modify the skeletal form of the product to incorporate the style elements. Eqn. (4.1) is used to implement the algorithm for this purpose. Style curves and the configuration map generate the style form of the product. This style form of the product is subjected to evaluation and modification before it is ready to move to the next stage.

3.3 Shape Rules and Morphing

This is the heart and brain of the model. The agents are implemented according to the features of the product. Each agent is responsible for a particular feature. The features are designed as per the feature shape rules. The feature shape rules are written by observing the product samples and extracting the characteristic feature elements. The shape rules are written to add/modify these features, when all the features are added using the shape rules, the design is complete.

Each feature should have at least one shape rule to implement it. There could be alternative rules to implement a feature. These rules form the rule set for that feature. If the rules in a rule set have the correspondence of the form, new rules can be generated by morphing the different rules from the set. This technique is very important from the point of view of generating the new designs. Apart from morphing, added flexibility and expansion of the possible shapes can be achieved using (i) the parametric variation of the feature rule shapes, and (ii) Envelope deformation (Fig. 2.). This paradigm of the parametric variation, Envelope deformation and morphing virtually provides a limitless possibility to the shapes and forms for the feature within the boundaries of the shapes rules.

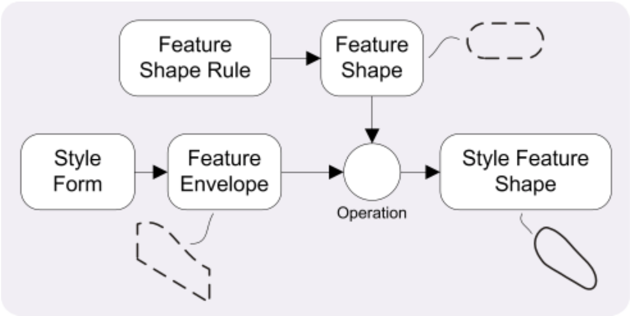

Fig. 2: Envelope deformation.

Fig. 3. shows the enormous generation power of the above schema of using parametric variation and the morphing. If two shapes are used for the schema, and shape one has n variation and shape two has m variations, the parametric variation provide the n + m shapes. If we use the k levels of morphing among the shapes, then the total number of shapes generated *N* is given as:

$$N = n*m*k - 2*n*m + n + m \tag{3.1}$$

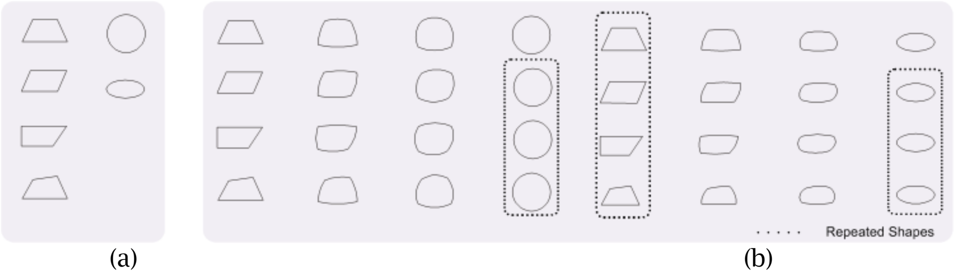

Fig. 3: Design variation through parametric variation and morphing (a) Shape with parametric design (b) Shapes with parametric design and morphing.

This is a huge number compared to the sum of the variations. For example, Fig. 3(a) shows two shapes with 4 and 2 variations respectively by employing parametric design methodology. The total shapes available in this case are 6 shapes, whereas, with the 4 levels of morphing, as shown in Fig. 3(b), 22 shapes are generated. As we can see, the generation of these shapes is governed by Eqn. (3.1).

3.4 Optimization of the Shapes

The solution to the design problem is guided and controlled by the constraints described in the optimization function. This function represents the constraints, which limits the search space. It may be based on satisfying the functional, ergonomic or aesthetic requirement. Apart from the geometric preference function and constraints, the optimization module may also incorporate the other mechanisms to modify the control parameters, which may be shapes, size, color, material, texture, etc. In this framework, we are limiting to the consideration of the shape control only, to optimize the design. Here, the style curve conformance to the feature envelope is the preference function with suitable boundary constraints. Eqn. (4.1) presents the optimization function used for this purpose.

3.5 3D Form

Every design has to be realized in the 3D form. The 3D rules, in conjunction with the configuration map and features shape, serve to convert the 2D design to the 3D form. They may also incorporate further dress-up treatments, like the filleting, chamfering, surface finish, texturing or any other post processing to the 3D form. Presently, we are limiting the model to the 3D form generation using the basic 3D operation like extrude, fillet and chamfer only.

4 IMPLEMENTATION

The presented framework is being implemented for the generation of design concepts for the industrial products. In order to test the paradigm, an example of a mobile phone is used as the test case. Initial trials show that the model can be used to enhance the designers ease and expand his/her capability in the generation and evaluation of design alternatives. The details of the implementation and a few results are presented below.

4.1 Analysis of the Product Shape Features

Various models of the mobile phone belonging to a common category are collected and analyzed, to identify the prominent and common features and their form (Fig. 4.). The analysis results in the enumeration of various features and their compositional relationship [12-13]. The form of the mobile phone can be decomposed in various features, as shown in the feature configuration map. This Configuration map serves as the basis for the design of agent hierarchy. Fig. 5 shows the agent hierarchy for the mobile phone. The Manager agent divides the tasks to different feature agents and communicates with them. The feature agents carry out the required tasks of the shape generation/ modification of the respective feature.

Fig. 4: The mobile samples and their configuration map.

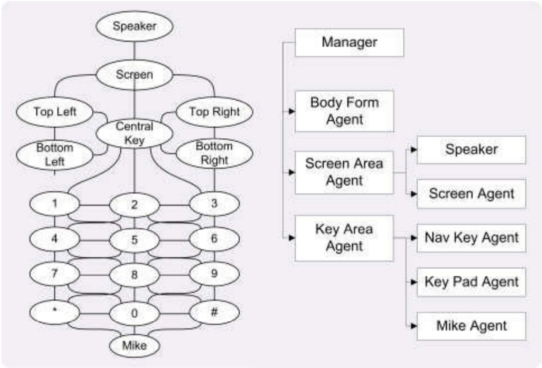

Fig. 5: The configuration graph and the agent hierarchy for the configuration map in Fig.4.

The complete product model is represented by a set of features and their interrelationship. This relationship can be represented as a feature configuration graph. Fig. 5 shows the configuration graph for the mobile phone. The nodes in the graph represent the features and the edges represent the properties shared among the features. Though in this example case, this shared property means the shape boundary, but it can mean any property relating the features. Each shared property will determine the construction of the specific configuration graph. The configuration graph plays a very important role in resolving the property specific conflicts among the features. Like in this case, the feature configuration map helps in determining the relative position and boundaries of various features.

4.2 Style Curve Mapping

The style curves provided by the designers are the guide to the designer's intention. These curves are used to deform the configuration map to achieve the style form of the configuration map. This is done by mapping some of the nodes of the configuration map by the user to the corresponding nodes of the sketch. At least three nodes are required per sketch curve to capture the shape of the sketched style curve. There is no need to map all the nodes, but larger the number of nodes mapped, the better will be the conformance to the style. Fig. 6. shows the correspondence between the style curve and the configuration map. With the knowledge of the sketch nodes and the reference nodes in the configuration map, it is possible to match all the nodes of the configuration map to the sketched style curve by minimizing the difference between the nodes marked by the designers and the corresponding nodes of the configuration map.

We can define a minimization function as per following scheme:

Let

$P = \{p_1, p_2, ..., p_k\}$ be the position of the reference nodes in the configuration map

$Q = \{q_1, q_2, ..., q_k\}$ be the position of the deformed nodes in the configuration map

$R = \{r_1, r_2, ..., r_m\}; m \le k$ be the position of the nodes modified by the designer

$S = \{s_1, s_2, ..., s_m\}$ be the position of corresponding set of the nodes in the configuration map

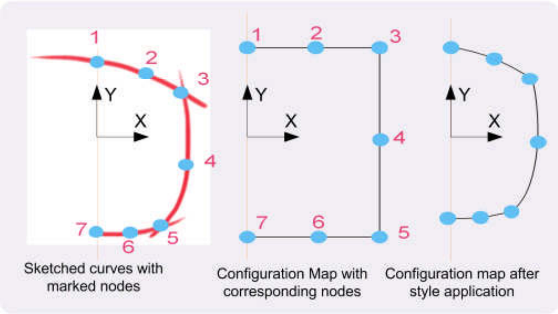

Fig. 6: The node correspondence between the style curve and the configuration map.

In order to achieve the styles position of the nodes in configuration map, following function needs to be optimized,

$$F = \arg\min\left[c_1\sum_{i=1}^{m}\left\|R_i - S_i\right\| + c_2\sum_{j=1}^{k}\left\|P_j - Q_j\right\|\right] \tag{4.1}$$

The first part of the function is the cumulative difference between the marked nodes. The second part of the function is the cumulative difference between the original and the deformed nodes. c_1 and c_2 are the weights. These weights are put in the function in order to normalize the relative order of the two terms.

The above function is subjected to various constraints. Like the nodes on the line of symmetry have the constraint as (x = 0). For all other nodes, (x = -x) is the constraint. There could be a few inequality constraints also. They are defined to maintain the valid design of the product. These are like the nodes belonging to the microphone of the mobile should be below the nodes defining the buttons etc. The configuration graph is used to establish these constraints.

4.3 Style Morphing

Morphing can also be used to generate design by operating with the complete model form. It is based on the operation with complete data set of the model, rather than on individual feature. In order to achieve the morphing among the data sets, they need to have the topological correspondence. It means that all the forms taking part in the morphing operation should have the same number of vertices and edge connectivity. A similar methodology is discussed by Kara et al [8].

Suppose there are m models with correspondence

Let U be the vector representing the base template or 'style-less' model,

and $V_i; i = 1,\ldots,m$ be the vectors representing various models with style,

It is possible to separate the style characteristics of the models as

Let $\Delta V_i = U - V_i : i = 1,\ldots,m$ be the style vectors representing the style characteristics

We can generate new style vectors by morphing among these styles characteristics as,

$$\Delta V_{new} = \sum_{i=1}^{m} w_i \Delta V_i \;\; ; \quad \sum_{i=1}^{m} w_i = 1 \tag{4.2}$$

w_i is the weight of the i^{th} style vector.

Using the morphed style vector, a new model U_{new} may be represented as

$$U_{new} = U + \Delta V_{new} \tag{4.3}$$

4.4 Multi-agent System and the Design Process

Multiagent system (MAS) enables the application of the shape rules automatically or interactively. The relationship between the MAS and the design space is mapped in the agent hierarchy, which mimics the design team with members have the task specific knowledge base and are responsible for the specific task. Here, the knowledge base is presented in the form of the shape rules for the features, but in the future, more complex behavior may be modeled on the agents. MAS directly maps the configuration map and the configuration graph is the part of the MAS knowledge base.

There is an important shift from the reported work on MAS for the form design [11], as the shape rules are part of the agent knowledge base rather than the reported arrangement in above reference, where the agents refer to a common repertoire of shape rules. The main advantage of the present scheme is that each feature is executed by the specific agent, and its knowledge base may be modified as the more complex shapes are needed to implement that feature, without violating or referring to other shape rules. This is important because of the fact that the features of the industrial products keep growing in the level of complexity.

The designer's sketch containing the style information is an input to the manager agent. These curves modify the configuration map to generate the style configuration map. This style configuration map is used by the manager to create a design template for the product. This template and other user information, such as the overall size and the aspect ratio of the product, act as input to generate the base design of the product. The base design contains the envelopes for various features. The manager also creates a stack of labels, which contains the information about each feature to be added to the design. This stack is created using the configuration graph. The manager also generates feature specific preferences/ optimization functions and the appropriate constraints for each feature or group of features. These are based on either the user input, or are automatically generated from the overall objective function. Here, we are using a simple preference function which maintains the minimum gap among the feature curves set by the designer. The preference function may be very complex containing sub-functions to achieve multi-objective optimization.

Once the overall size, envelope, constraints and preference functions are established for all features, the manager calls the agents based on the element of the label stack. This comprises the foundation for the feature addition to the base design. Now, the top element of the label stack is referred to invoke the appropriate agent. The base design, feature specific preference function, constraints and other relevant information are the input to the agent. Based on this information, the agent applies the shape rule to add the feature to the base design, or further evaluates the feature curves for the possible violations of the preference function, like crossing the boundary set by the feature envelope, violating the minimum gap among the feature boundaries, or the minimum/ maximum size of the feature. If so happens, the agent may try to achieve the optimum size/shape by appropriate rectification methods like scaling the size of the envelope, etc. If the optimum shape of the feature is not achieved in the set number of iterations, the incomplete design is referred back to the manager with the appropriate message. The manager, then generates an appropriate warning/ error message and seeks the user interaction to

- (i) Relax the preference function, or increase the number of iterations to complete the design of current feature, or
- (ii) Implement the next feature, or
- (iii) Halt the design process.

If the appropriate shape rule is successfully implemented and the feature curves are added, the modified design is sent back to the manager with appropriate message. The manager then, marks the feature as 'added' and calls the next element on the label stack to invoke the next agent. Similar process is repeated till all the features are added to the design and optimized. On the completion of this process, the 2D design is ready. The 2D design is then sent to the 3D agent with the appropriate information to achieve the 3D form of the product. Currently this step is achieved manually.

4.5 The Design Operations

Though there are various iterative loops of operations at different stages, the framework follows a sequential mode of operation for the generation of designs. The general process of design generation is depicted in Fig. 7.

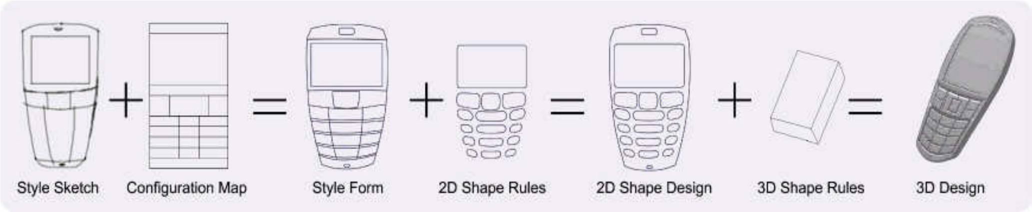

Fig. 7: The design pipeline.

The style sketch is read into the program and the skeletal structure provided by the configuration map is modified to map the style curve. This can be done using the style mapping model described in the previous sections, or interactively by the designer. The style form is then fed to the agent module, where the manager determines the form, size, shape and location of various features. It also creates the feature envelopes. Then, the manager calls the respective agents to add the features to various envelopes based on the shape grammar rules. Here again, the process can be done automatically or interactively to determine the deformation of the shapes added by the agents.

The new shapes for the feature rule can also be created by using the shape morphing. At this stage the shape of the features are optimized using the given optimization function. In the present case, the optimization function used is to maximize the minimum distance among the feature envelopes. A similar process as described above is used to achieve the valid layout of the feature shapes. The process of adding the feature shapes continues till all the features in the configuration graphs are added and optimized. At this stage the 2D form of the product is complete.

The 2D model is sent to the 3D module where various feature rules like extrude, bevel, fillet, chamfer, etc. are used by the manager to give the 3D form to the design. After the 3D form is complete, the model can be textured to incorporate the visual details.

As stated earlier, the framework can be used to design the product models in two modes of operation, interactive mode or automatic mode.

The interactive mode facilitates the user interaction at various stages of the design process. Whereas the automatic mode generates the design based on the range of deviation allowed in the style of the product and in the optimization function. These values are user defined and can be controlled to limit the search of the design space or the number of iterations executed by the model. The new designs can be generated using the morphing of the separated style vectors Eqn. (4.2.) applied to the base of template form of the model Eqn. (4.3). An interactive interface to implement the above scheme is given in Fig. 8.

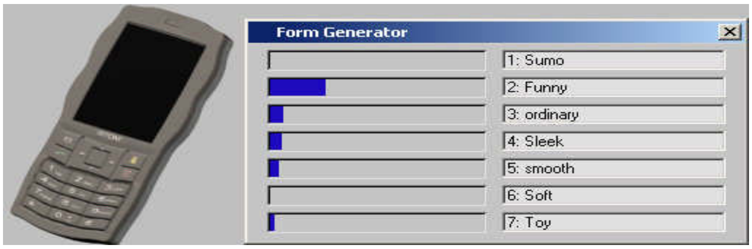

Fig. 8: The interactive design of a mobile phone with various style elements.

Various morphing percentage for the model in Fig. 8 are: Sumo - 0%, Funny - 24%, Ordinary – 8%; Sleek – 7%, Smooth – 6%, Soft – 0% and Toy – 3%. The template or base vector's share is 52%.

5 RESULTS

Fig. 9 presents some of the designs generated using the proposed framework. These shapes are generated in 2D. Then the 3D form generation is done manually.

Fig. 9: Some of the designs generated using the proposed framework.

6 CONCLUSIONS

In this paper, the software agents map the product configuration and the style sketch constraints. These agents search the design space provided by the model to arrive at the acceptable designs in conformance to the style preference. The agent based paradigm in best suited to simulate the design team environment. It is flexible, expandable and facilitates the optimum design with growth in knowledge and complexity of the product configuration.

The framework can be run in various modes, namely the interactive mode with user interaction and continuous evaluation by the user, or the autonomous mode where the designs are generated within a range of the prescribed parameter variation and/or conforming to a style sketch with varying degree of conformance along with the optimization function to minimize or maximize certain characteristics of the features. The framework exhibits the generative and creative potential far beyond the capability of the human designer, and may aptly bridge the gap between the creative/generative phase and the detailed designs offered by the present age CAD systems.

The current state of implementation is a proof of the concept. A lot more needs to be done to make the model efficient in terms of the style curve interpretation, shape rules, optimization function and its relational model with feature parameters. This is viewed as the future direction of the current research.

7 REFERENCES

[1] Chen, K.; Owen C. L.: Form language and style description, Design Studies, 18(3), 1997, 249-274.

[2] Chen, S. E.; Parent, R. E.: Shape averaging and its applications to industrial design, IEEE Computer Graphics and Applications 9(1), 1989, 47–54.

[3] Hao, Q.; Shen, W.; Zhang, Z.; Park; S. W.; Lee, J. K.: Agent-based collaborative product design engineering: an industrial case study, Computers in Industry, 57(1), 2006, 26–38.

[4] Hsiao, S. W.; Chen, C. H.: A semantic and shape grammar based approach for product design, Design Studies, 18(3), 1997, 275–296.

[5] Hsiao, S.W.; Liu, M. C.: A morphing method for shape generation and image prediction in product design, Design Studies, 23(5), 2002, 533–556.

[6] Hsiao, S. W.; Wang H. P.: Applying the semantic transformation method to product form design, Design Studies, 19(3), 1998, 309–330.

[7] Jennings, N.; Sycara, K.; Wooldridge, M.: A roadmap of agent research and development, Autonomous Agents and Multi agent Systems, 1(1), 1998.

[8] Kara, L. B.; Shimada, K.: Supporting early styling design of automobiles using sketch-based 3D shape construction, Computer-Aided Design and Applications, CAD Solutions, 2008, 867-876.

[9] Kavakli, M.; Scrivener, S. A. R.; Ball, L. J.: Structure in idea sketching behaviour, Design Studies, 19(4), 1998, 485– 517.
[10] Madhusudan, T.: An agent-based approach for coordinating product design workflows, Computers in Industry, 56, 2005, 235–259.
[11] Orsborn, S.; Cagan, J.: Multiagent shape grammar implementation: automatically generating form concepts according to a preference function, Journal of Mechanical design, ASME, Vol. 131, 121007-1, Dec. 2009.
[12] Orsborn, S.; Cagan, J.; Boatwright, P.: A methodology for creating a statistically derived shape grammar composed of non-obvious shape chunks, Res Eng Design, 18, 2008, 181–196.
[13] Orsborn, S.;, Cagan, J.; Boatwright, P.: Automating the Creation of Shape Grammar Rules, Gero, J. S. and Goel, A. K. (eds.), Design Computing and Cognition, 2008.
[14] Pahl, G.; Beitz, W. : Engineering Design, A Systematic Approach, Springer Verlag, 1988.
[15] Schon, D.; Wiggins, G.: Kinds of seeing and their functions in designing, Design Studies 13(2), 1992, 135–156.
[16] Stiny, G.: Introduction to Shape and Shape Grammars, Environment and Planning B, 7, 1980.
[17] Weiss, G.: Multiagent Systems, MIT Press, 2000.