The Hong Kong Polytechnic University
School of Design

The Development of Parametric Shape Grammars Integrated with an Interactive Evolutionary System for Supporting Product Design Exploration

Ho Cheong Lee

A thesis submitted in partial fulfilment
of the requirements for the
Degree of Doctor of Philosophy

December 2006

Certificate of Originality

I hereby declare that this thesis is my own work and that, to the best of my knowledge and belief, it reproduces no material previously published or written, nor material that has been accepted for the award of any other degree or diploma, except where due acknowledgment has been made in the text.

____________________________ (Signed)

Ho Cheong, LEE (Name of student)

Abstract

This thesis has developed an interactive system that uses parametric 2D and 3D shape grammars incorporating an evolutionary algorithm for exploring product forms at the early stage of design process. The evolutionary algorithm allows designers to explore product forms by modifying the control parameters of shapes and changing the application of shape grammar rules based on the interaction between designers and the system.

In recent years, shape grammars have been used in Computer-Aided Design (CAD) systems for generating stylistically consistent and novel designs. In this approach to design concept generation, human subjective selections and evaluations are involved in controlling the parametric modification of the shapes and shape grammar rules. System based on this approach is referred to as Interactive Grammar Based Design System (IGBDS). An IGBDS is capable of reproducing large numbers of stylistically consistent designs as well as exploring novel designs.

The design approach based on shape grammars, however, relies on the identification of quality shape grammar rules with sufficient accuracy but in the meantime maintaining a high degree of generality, in order for the system to generate forms with enough variations. The formulation of shape grammar rules useful for different specific design requirements in a computational system is difficult and time consuming in the domain of product form design, due to the complexity in the product surface modelling. This is so because the development of shape grammar rules requires tremendous amount of work in capturing the knowledge from the existing designs. However, most of the existing designs have different form features. It is difficult to find the consensus in generalizing the existing designs into shape grammar rules. In run time, the selection and evaluation processes of the shape parameters and the shape grammar rules are time consuming due to limited speed in rating and interaction between designers and the computer.

To address these problems and to enhance the generative capability of shape grammars in supporting product design, this research focuses on two issues: 1) The development of systematic approach to the formulation of shape grammars combining 2D and 3D forms and 2) Extending the generative capability of the product design support systems which use shape grammars. The aim of these two focuses is to enhance the power of shape grammar based design approach in terms of dealing with complexity of real design applications and to explore the potential of integrating evolutionary methods with shape grammars for increasing diversity in product design. The motivation for such an integration is based on the hypothesis that shape grammar rules modified by the genetic code scripts of an evolutionary

method (such as genetic algorithm) define a new combination of shape features for alternative designs. In this way, traditional shape grammars are extended to an interactive context in which generative and evolutionary computing methods are utilized with better potential of supporting product component design and configuration. Different scenarios in which designers interact with the implemented system in real design applications are studied and evaluated.

The first issue involves an analytical approach to understanding the relations between product design and form complexity, by identifying and analysing the information associated with complex form creation. The research on this issue included the development of methods for the classification of product components, the definition of design spaces for the component configuration, the specification of design constraints and the spatial relationships among components.

The second issue involves the development of a computational system for enhancing the generative capability of IGBDS in the product form exploration, by integrating an evolutionary algorithm with an IGBDS. This integration allows shape grammar rules to be modified at run time. This relied on a careful planning in retrieving the significant elements from the complex information network of shape grammars and encoding all the necessary information into genetic representation. A new representation scheme utilising both the power of genetic programming and shape grammar rule representations is developed, together with the control strategies for manipulating this new representation during an evolutionary design process. With such a method and system, the designers come to interact with the enhanced IGBDS to evaluate the designs and corresponding rules by determining the control parameters and control strategies in the evolutionary cycle. The system provides the visualisation of virtual 3D objects for the designers for their evaluation and selection, and allows them to explore alternative designs with shape grammar rules containing parametric variation potentials, through the integration of Genetic Algorithm (GA), Genetic Programming (GP) and parametric 2D and 3D shape grammars with labels.

The methodology for research carried out in this thesis follows the analytical study of shape grammars and product forms with the development of new computational representations for the integration of evolutionary methods with shape grammars. The implemented system and knowledge base of the shape grammars are tested with realistic design examples. The research involves three major steps: 1) The development of the methods of deriving shape grammar rules in a product design domain, 2) The formulation of genetic representation and control strategies, and 3) The exploration of design and evaluation of the integrated IGBDS with real design examples. With the system developed in this research the designers can explore design with more flexibility in varying the weighting factors, determining the control parameters for the shape grammar rules, selecting appropriate control strategies for

specific design characteristics, controlling order sequences of shape grammar rule application, and exploring form and configuration designs.

For the evaluation of the proposed method, representation and implementation of an enhanced IGBDS, a shape grammar rule base for digital camera design is developed and two prototype systems are implemented using this rule base in order to study the feasibility of proposed system with its supporting components, including the integration with an external 3D solid modelling kernel. The development and experiments of two prototype evolutionary IGBDS illustrated the flexibility in using parametric 2D and 3D shape grammars with labels to explore a wide range of engineering and industrial design problems involving complex and intuitive conception of forms and products. Both implemented systems are interactive, with the first one using a normal evolutionary algorithm, and the second one using genetic programming with parametric shape grammars. The experiments showed that the second prototype is more flexible, while more complex to implement, for the exploration of forms in terms of variety with the parameters within shape grammar rules being changed by the evolutionary algorithms. Based on the analysis and the evaluation of the results achieved in the context of real product design, the thesis concludes on the applicability of shape grammars to real product design and proposes several strategies with which this research can be further advanced for complex form design and visualisation in product design.

Publications and Conferences

Lee, H. C. and Tang, M. X. (2007). Evolving Product Form Designs Using Parametric Shape Grammars Integrated with Genetic Programming. Submitted to *Artificial Intelligence for Engineering Design, Analysis and Manufacturing (AIEDAM)*.

Lee, H. C. and Tang, M. X. (2006). Generating Stylistically Consistent Product Form Designs Using Interactive Evolutionary Parametric Shape Grammars. In *Proceedings of the Seventh International Conference on Computer-Aided Industrial Design and Conceptual Design (CAID&CD '2006)*, pages 374 –379.

Lee, H. C. and Tang, M. X. (2004). Evolutionary Shape Grammars for Product Design. In *Proceedings of the Seventh International Conference on Generative Art*, pages 309-322, Politecnico di Milano University.

Acknowledgements

I would like to thank my supervisor Dr. Ming Xi Tang and my cosupervisor Professor John Frazer for their continuous encouragement, enthusiastic guidance and support throughout the course of my research. They not only introduced me the topic of shape grammars and evolutionary computing, but also strengthen my background at the area of artificial intelligence and product design. Their continuous supervision along with many suggestive comments and consultations are greatly appreciated. I would also like to thank Professor Sharon Poggenpohl for her proofreading and suggestions on improving the presentation of this thesis, and Professor Celestino Soddu and Professor Andrew I-kang Li for their valuable comments and suggestions.

This Ph.D. project was undertaken in the Design Technology Research Centre (DTRC), School of Design, the Hong Kong Polytechnic University. As such, I would like to gratefully acknowledge the generous supports from all the members in the DTRC and School of Design. I would especially like to thank the fellow colleagues and Ph.D. students, Dr. Patrick Janssen, Dr. Zhenyu Gu, Mr. Kwai Hung Chan, Mr. Jian Xun Wang, Miss. Sylvia Tzvetanova, Miss Kathy Lo, Miss Nicole Schadewitz, Miss Weichen Tang and Mr. Wing Chuen Lau for their generous inputs, support and warm friendship. They have made my graduate life wonderful and energetic.

Finally, I owe a special depth of gratitude to my family and my beloved, Khiet Nhi Giang, for their patience, love and moral support, and for the stimulating discussions throughout this research, in my endeavour to undertake this research project.

Table of Contents

Certificate of Originality ii
Abstract iii
Publications and Conferences vi
Acknowledgements vii
Table of Contents viii
List of Figures xiv
List of Tables xvi

Part I Overview 1

Chapter 1 Introduction 2

- 1.1 Product Design 2
  - 1.1.1 Understanding the Process of Product Design 3
  - 1.1.2 Product Design Exploration 4
- 1.2 Shape Grammars 5
- 1.3 Problem Identification 7
  - 1.3.1 Stylistic Consistency 7
  - 1.3.2 Control of Design Process 9
- 1.4 Aims and Scope 11
- 1.5 Research Proposition 12
  - 1.5.1 Theoretic Presuppositions 13
  - 1.5.2 Development of Shape Grammars 13
  - 1.5.3 Integrating Shape Grammars with Evolutionary Algorithms 19
- 1.6 Significance and Potential Benefits 21
  - 1.6.1 Systematic Approach in the Development of Shape Grammar 22
  - 1.6.2 Extending the Generative Capability of Shape Grammar 22
- 1.7 Research Methodology 24
  - 1.7.1 Conceptualization Stage 24
  - 1.7.2 Implementation Stage 25
  - 1.7.3 Evaluation Stage 27
- 1.8 Overview of the Thesis 28

Part II Review of Related Work 31

Chapter 2 Shape Grammars 32

2.1 Introduction 32

2.2 Overview of Shape Grammars 33

2.3 Working Principles of Shape Grammars 33

2.3.1 Formulation of Shape Grammars 33

2.3.2 Representation of Shape Grammars 34

2.3.3 Manipulation of Shape Grammars 36

2.4 Computational Issues 39

2.4.1 Generative Capability of Shape Grammars 39

2.4.2 Emergent Properties of Shape Grammars 39

2.4.3 Classification of Shape Grammars 40

2.4.4 Development of Shape Grammars 41

2.5 Redefining Shape Grammars 49

2.5.1 Shape Grammars for Chinese Lattice 49

2.5.2 Redefining the Shape Grammars 51

2.5.3 Identification of Modifiable Elements 52

2.5.4 System Development 54

2.5.5 System Architecture 54

2.5.6 Implementation Results 56

2.5.7 Enhancing Generative Capability 58

2.6 Application of Shape Grammars to Design 59

2.6.1 Difficulties in Implementation 59

2.6.2 Optimising Shape Grammars 59

2.7 Summary 63

Chapter 3 Evolutionary Algorithms 65

3.1 Introduction 65

3.2 Overview of Evolutionary Algorithms 65

3.3 Working Principles of Evolutionary Algorithms 66

3.3.1 Background of Evolutionary Algorithms 66

3.3.2 Representation of Evolutionary Algorithms 68

3.3.3 Manipulation of Evolutionary Algorithms 68

3.3.4 Effects of the Environment 70
3.3.5 Designer Interaction 71
3.4 Review of System Applications 71
3.4.1 The First Application 71
3.4.2 The Second Application 73
3.5 Computational Issues 76
3.5.1 Design Optimisation 76
3.5.2 Design Exploration 77
3.5.3 Form Creation 77
3.5.4 Product Conceptualisation 78
3.6 Evolutionary Designs 79
3.6.1 Potential Integration Approaches 79
3.6.2 Evolutionary Design Systems 80
3.7 Summary 86

Part III Theoretical Development 88

Chapter 4 The Theoretical Framework 89

4.1 Introduction 89
4.2 Overview of Systematic Approach 90
4.3 Information Network of Shape Grammars 93
4.3.1 Identification of Product Design Characteristics 94
4.3.2 Redefining Specific Shape Grammars 95
4.4 Core Variant Model 95
4.4.1 Core Variant Design Feature Library 97
4.4.2 Standard Association Table 97
4.4.3 Special Association Table 98
4.5 Parametric Shape Grammars 98
4.5.1 Classification of Shape Grammar Rules 98
4.5.2 Construction Shape Grammar Rules 99
4.5.3 Configuration Shape Grammar Rules 100
4.6 Evolutionary Architecture Integration 100
4.7 Summary 102

Part IV Implementation and Analysis 105

Chapter 5 Parametric 2D Shape Grammars 106

5.1 Introduction 106

5.2 Digital Camera Form Design with Shape Grammars 106

5.2.1 Information Network of Shape Grammars 107

5.2.2 Core Variant Model 108

5.2.3 Parametric 2D Shape Grammars of Digital Camera Forms 110

5.2.4 Construction Shape Grammar Rules 110

5.2.5 Configuration Shape Grammar Rules 115

5.3 Evolutionary Architecture Integration 116

5.3.1 Identification of Modifiable Elements (Phenotypes) 116

5.3.2 Generating New Shape Grammar Rules 118

5.3.3 Construction Rule Order Sequences 121

5.3.4 Manipulation of Coded Modifiable Elements (Genotypes) 122

5.3.5 Genetic Algorithm 122

5.3.6 Evaluation Method 123

5.4 System Development 123

5.4.1 System Architecture 123

5.4.2 System Operations 125

5.5 Implementation Results 125

5.5.1 Application of IGBDS 125

5.5.2 Application of IGBDS with Evolutionary Process 127

5.5.3 Examples of Deriving New Rules 129

5.6 Summary 130

Chapter 6 Parametric 3D Shape Grammars 132

6.1 Introduction 132

6.2 Parametric 3D Shape Grammars 133

6.2.1 Limitations of the First Prototype 133

6.2.2 Parametric 3D Shape Grammars for Free Form Design 133

6.2.3 Construction SG Rules for Free Form Generation 135

6.2.4 Construction SG Rules for Component Generation 138

6.2.5 Configuration Shape Grammar Rules 141

6.3 Evolutionary Architecture Integration 143
6.3.1 Overview of the Evolutionary Architecture 144
6.3.2 "GP-GA-SG" Genetic Representation 145
6.3.3 Control Strategies 146
6.3.4 An Example of Control Strategy Application 149
6.3.5 Operation of Control Strategy 151
6.3.6 Genetic Programming 153
6.3.7 Multi-objective Functions 154
6.4 System Development 157
6.4.1 System Architecture 157
6.4.2 System Operations 158
6.5 Implementation Results of the Second Prototype 159
6.5.1 Initial Setting of the System 159
6.5.2 Implementation Results – First Experiment 159
6.5.3 Application of the First Control Strategy 161
6.6 Implementation Results - Second Experiment 162
6.6.1 Initial Setting of the System 162
6.6.2 Implementation Results 163
6.6.3 Evaluation of Implementation Results 164
6.7 Summary 169

Part V Conclusions 171

Chapter 7 Conclusions and Future Work 172

7.1 Revisiting the Objectives 172
7.1.1 Key Problems 172
7.1.2 Integration of Shape Grammars and Evolutionary Algorithms 173
7.2 Revisiting the Research Methodologies 175
7.2.1 Revisiting the Research Proposition 175
7.2.2 Organising Relevant Information 175
7.2.3 Development of Shape Grammars 176
7.2.4 Evolutionary Architecture Integration 179
7.3 Summary of the Key Research Activities 181
7.4 Contributions 183
7.5 Future Work 185

7.5.1 Short Term 185
7.5.2 Long Term 185
7.6 Conclusions 187
Bibliography 189

List of Figures

Figures for Chapter 1

Figure 1.1 The product realization process (Figure is adapted from (Dixon and Poli, 1995))

Figure 1.2 Integrated SG and evolutionary algorithm framework

Figures for Chapter 2

Figure 2.1 Connection between knowledge level and exploration capability level of shape grammars

Figure 2.2 Chinese lattice design: Chengtu, Szechwan, 1825 AD. (A modified version of the original figure shown in (Stiny, 1977) is used here.)

Figure 2.3 Shape grammars for the generation of lattice design (Stiny, 1977)

Figure 2.4 Initial shape for the generation of lattice design (Stiny, 1977)

Figure 2.5 The symmetric properties of Chinese lattice design

Figure 2.6 The first rule of newly derived shape grammar

Figure 2.7 The symmetric rules of newly derived shape grammar

Figure 2.8 The pattern generated with the rule sequence: 10, 40, 40, 30

Figure 2.9 Different ways to access the Inventor API. (Extracted from the help menu of AutoDesk Inventor)

Figure 2.10 The user interface for experimental case study of IGBDS

Figure 2.11 Each labelled shape, Shape A to Shape D, has eight shapes to be selected by designers

Figure 2.12 The help menu for experimental case study of IGBDS

Figure 2.13 The Chinese lattice design shown in figure 2.2 is generated by the system with the rule sequence: 10, 30, 30, 40, 40 and Shape A to Shape D with shape number: 5, 6, 5, 6

Figure 2.14 Pattern generated with the rule sequence: 10, 30, 40, 40, 30 and Shape A to Shape D with shape number: 5, 6, 7, 8

Figure 2.15 Pattern of 3D objects with the rule sequence: 10, 30, 40, 40, 30 and Shape A to Shape D with shape number: 1, 2, 3, 4

Figures for Chapter 3

Figure 3.1 Operation of GA

Figures for Chapter 4

Figure 4.1 Systematic approach in the development of shape grammars for product designs

Figure 4.2 Information network of shape grammars

Figures for Chapter 5

Figure 5.1 Typical arrangement of components configured at the main body

Figure 5.2 Operation of GA

Figure 5.3 Interactive design process utilising shape grammars and evolutionary algorithms

Figure 5.4 The first user interface of the first prototype evolutionary IGBDS
Figure 5.5 The second user interface of the first prototype evolutionary IGBDS
Figure 5.6 The GA has generated the population of the results at 200 generations
Figure 5.7 The two designs with higher scores after over 300 generations
Figure 5.8 Solutions are converged after over 300 generations
Figure 5.9 New design generation
Figure 5.10 A new SG rule is derived for the chosen design

Figures for Chapter 6

Figure 6.1 Generative specification of the class of compact digital camera forms
Figure 6.2 An abstracted Core Variant Model composed of combined NURBS surfaces and components for the generation of the digital camera form design
Figure 6.3 Construction SG rules for free form exterior of main body generation
Figure 6.4 Construction SG rules for component generation
Figure 6.5 Configuration shape grammar rules for assembly
Figure 6.6 Control of sequential generation of the components
Figure 6.7 Genetic representation of the evolutionary algorithm – "GP-GA-SG" interface
Figure 6.8 Parameters and control variables for the form design of a compact digital camera
Figure 6.9 Initial setting of parameters and control variables for a compact digital camera
Figure 6.10 GP crossover – Parent A crossed with parent B to produce the child
Figure 6.11 Modified setting of parameters and control variables by crossover operation for the first control strategy
Figure 6.12 User interface of the evolutionary IGBDS
Figure 6.13 Initial random generation of designs
Figure 6.14 Regular type designs - results obtained from the first generation (top left), 50 generations (top middle), 100 generations (top right), 150 generations (middle right), 200 generations (middle middle), 250 generations (middle left), 300 generations (bottom left) and the back view of the generated design (bottom middle)
Figure 6.15 Results obtained from the first generation (top left), 100 generations (top middle), 200 generations (top right), 300 generations (bottom left), 400 generations (bottom middle) and 500 generations (bottom right)
Figure 6.16 Variation of objective fitness values with different weighting factors and control strategies

List of Tables

Tables for Chapter 2

Table 2.1 Properties of basic elements (Table is redrawn from (Stiny, 2006, p. 164))
Table 2.2 Shapes in algebras (Table is redrawn from (Stiny, 2006, p. 180))
Table 2.3 Some properties of shapes (Table is redrawn from (Stiny, 2006, p. 196))
Table 2.4 More properties of the part relation (Table is redrawn from (Stiny, 2006, p. 196))

Tables for Chapter 5

Table 5.1 Association table for the interrelationships among digital camera components
Table 5.2 SG rules for the exterior of the main body
Table 5.3 SG rules for the zoom lens and swivel device
Table 5.4 SG rules for the flash and viewfinder
Table 5.5 SG rules for the LCD screen
Table 5.6 SG rules for the multi-purpose control button
Table 5.7 SG rules for the shutter and setting buttons
Table 5.8 SG rules for the exterior form styles of the main body and decorative features
Table 5.9 The configuration SG rule
Table 5.10 The configuration constraints
Table 5.11 The phenotype representation for the first prototype evolutionary IGBDS
Table 5.12 Seven ways available for the GA to take action
Table 5.13 Available shapes in a library
Table 5.14 An example of algebra operation to generate four new shapes
Table 5.15 The change of form style of main body by the GA
Table 5.16 An example of a new SG rule is derived
Table 5.17 Construction rule order sequences
Table 5.18 Genotype and phenotype representations for the first prototype system

Tables for Chapter 6

Table 6.1 Parameters and control variables for the first control strategy
Table 6.2 Properties in each SG group
Table 6.3 Initial setting of parameters and control variables for the first control strategy
Table 6.4 Modified setting of parameters and control variables by crossover operation for the first control strategy
Table 6.5a Implementation results of the evolved parameters of SG
Table 6.5b Implementation results of the evolved parameters of SG

Part I Overview

Part one (Overview) consists of an introduction chapter which presents an overview of this thesis.

Chapter 1 (Introduction) gives an introduction of the overall framework, which outlines the whole research background and themes. First the SG based design method is introduced and certain key problems in engineering and product design domain are identified. Then the clearer research objectives are defined. An outline of the research proposition is given based on the main research objectives. Finally, research methodological issues in the development of the integrated SG and evolutionary algorithm framework are discussed.

Chapter 1

Introduction

1.1 Product Design

Product design involves complex activities in which designers, engineers and manufacturers have to cooperate in specifying design problems, developing flexible solutions and utilising the resources for solving design problems. The key indispensable tool in product design is the use of computers in assisting designers, engineers and manufacturers in the design process. Research and developments in the application of Computer-Aided Design (CAD) systems integrated with artificial intelligence (AI) techniques in enhancing the product design process are demanding. Recently, research in exploring shape grammar (SG) approach to product design has received more and more attention by many researchers.

This research focuses on the development of a computational framework which integrates two key computational techniques: shape grammars and evolutionary computing for supporting product design activities. The aspects related to how such integration should be developed with a philosophical concept that can be evaluated with product design experimentations are identified in this chapter.

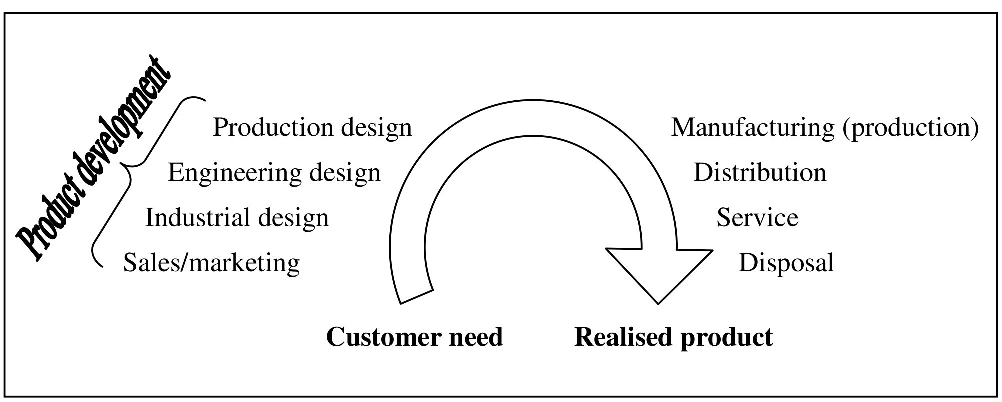


Figure 1.1 The product realization process (Figure is adapted from (Dixon and Poli, 1995))

Before attempting to develop a computational framework for product design exploration, it is necessary to understand the product realization process (Figure 1.1). The product realization process involves activities participated by the employees from various departments in an enterprise during the life cycle of a product.

The means of transformation from a customer need into a realized product are the product realization process. Dixon and Poli (1995) considered the product realization process which involved the whole firm with a complex set of interrelated activities, both cognitive and physical. The means of transformation are therefore referred as the interrelated activities by which new and modified products are conceived, manufactured, delivered to market, serviced, and disposed of. Figure 1.1 illustrates the process with physical activities and decision-making activities (cognitive), including: 1) Product development activities such as sales, marketing, industrial design, engineering design and production design; and 2) Post product development activities such as manufacturing, distribution, service and disposal.

1.1.1 Understanding the Process of Product Design

As shown in Figure 1.1, the first part of product realization process focuses on the physical and decision-making (cognitive) activities for product development such as sales, marketing, industrial design, engineering design and production design. The objective for this part is to support product design exploration from which the customer requirements are transformed into a realized product.

For industrial design, Eggert (2005) described that “Industrial design activities focus on how the new or revised product idea is compatible with the customer’s anatomical limitations and/or aesthetic trends in the market place. Often the industrial design group will prepare an artistic rendering or a physical model that illustrates basic product form, colour, texture, and intended functionality.”

Contrast this definition with IDSA (Industrial Design Society of America), IDSA defined that "Industrial Design (ID) is the professional service of creating and developing concepts and specifications that optimize the function, value and appearance of products and systems for the mutual benefit of both user and manufacturer." Eggert (2005) addressed the issues of how industrial design could improve the aesthetic values of existing products whereas IDSA concerned the issues related to how industrial design could add values to products for benefiting customers and manufacturers.

For engineering design, Dym and Levitt (1991) defined that “Engineering Design is the systematic, intelligent generation and evaluation of specifications for artefacts whose form and function achieve stated objectives and satisfy specified constraints.”

Contrast this definition with Eggert (2005), Eggert (2005) specified that "Engineering design activities result in recommended manufacturing specifications that satisfy the customer's functional performance requirements and manufacturing constraints." Eggert (2005) addressed the issues of how engineering design could achieve manufacturing specifications whereas Dym and Levitt (1991) considered the issues related to how should the process of engineering design be defined.

According to the above definitions, product design can be appropriately described as the determination and specification of the parts of a product and their interrelationships so that the assembly of the parts can perform specific functions with values and attractive exterior appearance. With such a wider scope of product design, the challenging task of product designers is to apply their creativity to synthesize new ideas and utilise their knowledge and skills from various domains such as mathematics, sciences, and manufacturing to determine and foresee how the new designs should behave before the products are created or manufactured.

1.1.2 Product Design Exploration

In the research field of computational design, design is largely seen as an ill structured problem for which exploration is more of a central activity than optimisation. This is due to the fact that design problem is hard to define in the beginning and the problem may evolve as the design process goes on. In such an understanding, it is difficult to develop a system capable of identifying the best solution to a design problem via the mechanism of searching a well defined design space in which design variables and constraints are well defined.

In this thesis, the definition of product design exploration includes those mentioned product development activities except the production design activities. However, the focus of this thesis is on the development of a computational framework for product form design exploration. In the context of this thesis, the term design exploration is used to define an interactive process during which a user or a designer is supported by the system. The system contains certain knowledge about the form of new design and uncertain about the new requirements. As such the system needs to work with the designer in order to apply appropriate shape grammars whilst the designer needs to rely on the system to perform the repeated tasks of generating a large number of different designs based on the strategies and control parameters selected by the designer. In this sense, exploration is seen as the basis for developing new design concepts while optimized designs may be discovered through the interaction in which the system is given a direction by the designer's input. As a result, the designer is given support in computation of new designs based on the knowledge contained by the system.

1.2 Shape Grammars

Shape Grammars are tools used to generate designs by encoding the knowledge implicitly in the shape grammar (SG) rules and applying the rules to design. In SG, both visual expression with shapes and verbal expression with signs or symbols can be unified to describe the complex properties of designs. A set of SG rules is defined by shapes themselves (visual expression) or shapes associated with properties (verbal expression) like labels, colours, costs, weights and etc.

The rules are used to describe the transformation process between shapes. Each rule represents part of the whole transformation process that leads the designs starting from the initial stage to the final stage. The rules are executed in order and somehow recursively either by the designer interactive selection or under the control sequences specified within the rules.

When SG is used to generate two or three dimensional (2D or 3D) forms and the configurations of these generated forms in a particular design domain, the designers are involved to select designs based on different requirements such as aesthetic appeal and determine the values of the parametric control variables. Such system is called Interactive Grammar Based Design System (IGBDS). In recent years, IGBDS is widely adopted in architectural, engineering and product design domain. IGBDS is capable of generating large numbers of alternative designs, stylistically consistent designs and novel designs. A detailed discussion on IGBDS is reported in Chapter 2.

More recently, IGBDS is feasibly extended in business and marketing applications such as capturing brand identities. It is important for companies to create brand new products and sustain original brand identities through product development process. The success of IGBDS relies on the formulation of knowledge encapsulated in the SG rules and the techniques used to explore the SG rules.

Prior to the application of IGBDS, a set of rules should be derived based on theories and practical experiences in designing the objects, or by analysing the existing objects in particular application domains. In industries, the objects are products and belong to engineering and product design domain. The theories in designing the products are developed through research, experiments, creative thinking and innovative ideas while the practical experiences are gained from experts in a particular field of design application. After the rules are derived, the rules can be applied to simulate the real design process of a product in a virtual 2D or 3D space. A comprehensive review specific to designing engineering SG was given by Cagan (2001).

Shape Grammar Based Approach to Product Design

Recently, IGBDS has been applied in product design domain to enhance product design process. For example, Cagan et al. developed the coffeemaker grammar, motorcycle grammar, hood panel grammar and vehicle grammar (Agarwal and Cagan, 1998; Pugliese and Cagan, 2002; McCormack and Cagan, 2002; Orsborn et al., 2006).

The coffeemaker grammar generates novel designs using function labels to maintain proper function-to-form sequences (Agarwal and Cagan, 1998). The coffeemaker grammar has further been developed by incorporating a decision making method in which the grammar rules are associated with cost expressions (Agarwal et al., 1999). With this approach, the designers can make decisions to select appropriate rules by evaluating the generated coffeemakers with costing information during the design process.

The motorcycle grammar was developed to capture brand identity (Pugliese and Cagan, 2002). The hood panel grammar generates novel designs with shape emergent properties (McCormack and Cagan, 2002). The vehicle grammar creates different cross-over vehicles by defining and combining different vehicle classes (Orsborn et al., 2006).

Other examples include a semantic and SG approach to product design developed by Hsiao and Chen (1997). The SG generates new product forms which satisfy to the customers' physical and psychological requirements.

Ang et al. (2006) have applied an evolutionary algorithm to evolve a set of 2D SG rules for the generation of Coca-Cola bottles. The evolved SG rules are executed in sequence and associated with parameters to generate bottles with Coca-Cola styles. The bottles generated fulfil specific volume requirements. The issues of designing branded products are investigated in their approaches. A comprehensive survey which compared the development processes, application areas and interaction features of different SG approaches was given by Chase (2002).

1.3 Problem Identification

Solving design problems are complex in nature that designers have to make decisions on implied information which is not readily available. The implied information refers to inconsistent specifications, over- or under-constrained conditions defined at the beginning of, and during the design process. Simon (1984, 1990) determined that the design problems become "ill-defined" under this situation. Most ill-defined problems have uncertain characteristics in defining the problems, their possible solutions and even the methodologies in obtaining those solutions. Janssen et al. (2002) further explained that the nature of ill-defined design problems is unstructured and the solutions are in a vast multidimensional design space.

In this thesis, two major design problems related to product design exploration are addressed: the first major problem relates to issue like balance between stylistic consideration (Eg. stylistic consistency) and technical innovation (or stylistic innovation), and the second major problem relates to the issue like the control of product design exploration under multi-dimensional requirements.

1.3.1 Stylistic Consistency

Due to fierce market competition in the industry, the companies face challenges in launching new products to the market periodically. The launched products may not necessarily be major technological break-through products, but should at least have new features which add values to the existing products. For example, the new features of the products may be designed as a whole series in order to reflect a new particular style.

Product designers try to balance stylistic consistency and stylistic innovation for the new designs. The management of style is one of the critical issues in product design exploration. The difficulties are arisen from maintaining the brand image while introducing new design features to the products. If a new product form has been radically modified, then it may not be coincided with the brand image from customer perceptions.

Stylistic consistency is a typical "ill-defined" design problem since style does not have a universal definition. For the purpose of this work, the "style" of a product may be interpreted as design features or characteristics which look attractive and perceivable or recognisable by customers. In general, the key distinctive characteristics of a product are the disposition of components, the types of components used and the boundary constraints of the component form.

The use of a shape grammar to generate products with consistent styles is a definite advantage since the shape grammar encodes the details of a particular product style into shape grammar rules. However, the use of a shape grammar to manipulate the modification of product style is a difficult task. When an existing shape grammar is defined for a particular style, the designers can select different set of rules from each branch of subset of rules. The resulting product design can therefore be generated with certain design characteristics of a particular style. If designers want to develop a shape grammar to generate designs with a particular style, or to modify an existing shape grammar with new style, three critical issues have to be addressed.

The first issue relates to the definition and creation of product style. There are questions like: how to define a style? What elements constitute to a particular style? Is there any concern about historical and cultural background when defining a product style? What are the views of customers, sales, manufacturers, designers for the particular product style? Besides, encoding a "style" into the shape grammar rules is subjective to the shape grammar developer or designers. The shape grammar developer and designers cooperate to analyse the existing products and convert the distinctive characteristics into the rules in accordance to their knowledge and experience.

The second issue is concerned with the cultivation of a standard for maintaining the stylistic consistency. A product style represents the image of a brand identity when companies promote their products with certain key identifiable characteristics. Usually, a family of products with a particular style is delivered for particular marketing customers. For example, the strategies of a company are to lunch a series of mobile phones targeted on teenagers. The family set of products has a good interface and can be changed with different colour outlooks. Maintaining the stylistic consistency among a family set of products can be applied as a company strategy in promoting the brand image.

The third issue is related to the modification of product style under the constraints of avoiding the distortion of the brand image for new product development. A style is maintained by means of converging the languages defined by the shape grammar and the language of stylistically correct designs. Conflicts exist when designers try to explore designs by changing the languages defined by the shape grammar. Since the style of the generated results is interpreted subjectively by the designers and customers, this introduces difficulties to designers. For a simple case like the regular type style, it can easily be identified from the results. For a complex case like cultural and brand identity style, it is hard to determine the structural organization of components and the variation of forms from the results. As a result, designers will have difficulties in balancing the effects of visual change of products among the brand image and customer perceptions of quality, service, and the intangible associations.

This thesis addresses the issues related to the formulation of shape grammars for a particular style, and the modification of shape grammars for new defined style. The focus is on the development of control strategies which addresses the technical parts of the formulation and modification of shape grammars for generating stylistic consistent or new designs, rather than addressing all the above mentioned issues. Suggestions on the future improvement of the formulation of shape grammars are provided in the final chapter of this thesis for generating designs with other concerned issues like cultural background.

1.3.2 Control of Design Process

As described earlier in section 1.3, the nature of solving design problems is complex. Apart from the problem of stylistic consistency specified in section 1.3.1, this section specifies the second major product design exploration problem addressed in this thesis. The second major problem relates to the issues of the control of product design exploration under multi-dimensional requirements. In order to support to realise the importance of this problem, various approaches in research of design paradigms in solving design problems are reviewed first.

The literature outlines a broad scope for solving design problems. These views range from designating design problems as search problems (Kanal and Kumar, 1988), to exploring alternative possible solutions (Janssen, 2004; Frazer, 2002). These suggest controversial views of solving design problem activities both in simplifying and complicating the design tasks. Specifying appropriate design problems with right kind of abstractions and correctness and proper interpretation by designers also leads to diverse the scope of design problems (Dorst, 2006).

From the view of design as a searching process, Kanal and Kumar (1988) simplified the design problems as search problems in optimising the solutions. The improvement of designs is achieved by searching among a selection of some well known and near optimal solutions. This kind of searching metaphors aims to elucidate the design process.

From the view of design as an exploration process, Janssen et al. (2002), on the contrary, has criticized the searching metaphors for solving design problems: "However, they (the searching metaphors) do not accurately reflect the reality of the design process and thereby actually result in further confounding the issue" (Janssen, 2004).

Frazer (2002) has identified such confounded issues: "This is why it is misleading to talk of design as a problem solving activity—it is better defined as a problem finding activity. This has been very frustrating for those trying to assist the design process with computer based problem solving techniques. By the time the problem has been defined, it has been solved. Indeed the solution is often the very definition of the problem" (Frazer, 2002).

From the view of design as a co-evolution process, Dorst and Cross (2001) have described an empirical study to analyze and describe the design process as a "co-evolution" of the design problem and the design solution.

These varying views contribute to the ad hoc nature of various approaches in research of design paradigms in solving design problems. Since the real problem is not well defined at the beginning of the design process, solutions cannot be well known in advanced. The solutions can be progressively found when the real problem is being continuously discovered and refined during the design process. Furthermore, there are no absolute evaluation methods to completely validate the solutions. The evaluation of the solutions is depended on the designers to determine whether the design problem is sufficiently described (Ozkaya and Akin, 2006).

Following the reviews, it can be seen that solving and specifying design problems will not be easily obtained by designers in a straight forward manner. Besides, the design requirements are multi-dimensional. This means that the designs have to be evaluated against a broad array of requirements. Often some requirements can be evaluated physically like weight whereas some can only be evaluated cognitively like designer's preference. As a result, the uncontrolled design process with multi-dimensional requirements reduces the chance in successfully solving and specifying design problems. The successful chance of conquering design problems can be raised up higher if there are methodologies provided to monitor the design process.

Shape grammar (SG) is an excellent tool in monitoring the design process for generating designs. However, there are two fundamental issues to be addressed before the potential advantages of shape grammar can be fully utilised. The first issue deals with how to approach the formulation of a shape grammar whereas the second with how to approach the modification of an existing shape grammar.

For the first issue, the basic premise in developing an Interactive Grammar Based Design System (IGBDS) is to systematically derive a set of SG rules to generate designs which fulfil specific requirements. Either theories or empirical skills from experts are difficult to identify in developing the SG. It is a time consuming process to derive theoretical design concepts by means of research. Also, the practical skills of experts are qualitative in nature and therefore hard to quantify for computation. Therefore, most of the SG rules are derived through analysis of existing designs.

However, some existing designs are stylistically inconsistent and therefore have different features, which in turn have discrete geometric attributes regarding spatial geometry. If these stylistically inconsistent designs are required for analysis, this will introduce configuration conflicts and increase the complexity of the SG. Besides, some features of the existing designs have different evaluation values from different standpoints. For example, a product designed with round fillet features around its exterior is expensive but has better aesthetic visual appeal. As a result, objectives to find the consensus of how features should be encoded in the SG rules from the existing designs blur.

For the second issue, a specific SG is limited to generating designs within a confined design solution space. It is based on the assumption that the specific SG is tailored for one particular design problem. If there are variations on the design requirements, the specific tailored SG cannot generate satisfactory solutions. From the generative points of view, one may argue that the specific SG can generate infinite designs by recursively applying the SG rules themselves using maximal representation. In this case, the specific SG can generate emergent solutions which are not confined in a specific design solution space. Theoretically, one may get emergent solutions if the specific SG generates an infinite number of designs. Practically, the chances to get emergent solutions may not easily appear if the specific SG does not change to adapt for the new design requirements. As a result, there is a necessity to modify the specific SG for new and more generic design problems.

1.4 Aims and Scope

To address the above issues, two key objectives relate to deriving information rich and high quality of SG and enhance the generative capability of SG for product design problems. First, a systematic approach to organise all the important information for the development of SG is established. This approach aims at deriving a complex information network of SG which models the complex interrelationships among shapes and all related attributes. The information network of SG is an extension to the original SG formalism. It should allow for modelling the complex interrelationships among shapes and all related attributes, and be able to compute certain kinds of designs more easily or expressively than with a standard SG.

In this information network of SG, a core variant model is built to capture all the relevant information of design objects such as attributes, constraints on attributes, spatial relations among shapes, rule order sequences and etc. The core variant model is constructed based on the analysis of the SG which generates the existing designs. The modifiable elements of the SG are identified from the analysis. These modifiable elements form the key parts of the core variant model. By modifying the modifiable elements of the core variant model, new SG representation and rules can be derived for more generic designs. With this information network of SG, the generative capability of a specific grammar is extended which can generate more generic designs.

Second, the existing SG rules in an IGBDS should be automatically modified when there are new design requirements. An evolutionary approach is incorporated to an IGBDS to evolve the existing SG rules in a controlled manner. A control mechanism is developed to resolve configuration conflicts and fulfil various kinds of constraints during the generation process. The key issues in this approach include:

- utilizing the power of genetic and SG representation in the integrated SG and evolutionary algorithm framework,
- introducing control strategies to evolve a set of stylistically inconsistent designs which are gradually modified into stylistically consistent designs,
- resolving configuration conflicts for different features by constraining the maximum boundaries of features and embedding collision avoidance criteria in objective functions,
- adopting weighting methods in determining the evaluation values of designs,
- developing multi-objective functions for the evaluation from many different perspectives, and
- allowing the designers to alter the existing sets of SG rules by modifying the control parameters of objective functions and selecting appropriated control strategies.

1.5 Research Proposition

Two key perspectives related to the development of SG thoroughly influence the development of the integrated SG and evolutionary algorithm framework. The theoretic presuppositions based on these two perspectives are first stated in section 1.5.1. Further interpretation of these two perspectives forms the core of this research proposition and is discussed in detail in the section 1.5.2 and 1.5.3.

1.5.1 Theoretic Presuppositions

The conceptual formulation and methodology for the development of this framework is influenced by two key perspectives:

A) Systematic Approach in the Development of SG

From the theoretical points of view, shapes can be calculated by SG rules for the exploration of designs. The SG rules calculate shapes in accordance to the knowledge of design implicitly embedded in the SG rules. Understanding the use of SG rules in exploring designs is a way of understanding the design process. It is of the same importance to understand the reasons and methods in deriving the SG rules. Without the clear understanding of why and how to derive the SG rules, the applicability of the SG rules is not fully utilised but hindered. The development of an analytical approach is useful in deriving the SG rules with the understanding of the relations between product design and form complexity. The analytical approach should be capable of identifying and analysing the information associated with complex form creation. Based on the analysis, an information network of SG can be established for the classification of product components, the definition of design spaces for the component configuration, the specification of design constraints and the spatial relationships among components. In engineering and product design domain, this analytical approach is critical to the development of successful SG based design applications.

B) Extending the Generative Capability of SG

Engineering and product design processes can be simulated by the application of SG. To explore designs to fulfil new requirements or constraints, the product design processes should probably be amended. This in turn will lead to amending the corresponding SG rules in the SG based design applications. Investigation on all related issues in the modification of SG rules results in exploiting the potential generative capability of the existing SG rules. This exploitation process directly influences the generative capability of IGBDS. Consequently, the importance and necessity in enhancing the generative capability of IGBDS are emphasised in the exploration of designs.

1.5.2 Development of Shape Grammars

With the understanding of complexity in solving the two major design problems introduced in section 1.3.1 and 1.3.2, this section explains the strategy to solve these problems. Since the view of design exploration affects the strategy, the view this thesis takes should first be clarified. In this thesis, the exploration of designs is not only categorised as a problem solving activity but also a problem finding activity.

These two activities link with each other in an interactive design environment during the design process.

Following the above view, three parts can be itemised from the overall strategy. The first part of the strategy describes the role of professionals in various departments of an enterprise, and the computational framework. The second part of the strategy applies a scenario to describing how the professionals and computational framework can cooperate in an interactive design environment. The third part of the strategy addresses the issues of the construction of the computational framework.

A) Interactive Design Environment

An interactive design environment is created for the design activities to be taken by the professionals and the system. Within the environment, designers can communicate and coordinate different professionals in various departments of the enterprise.

The design activities addressed in this thesis belong to product design domain, and are referred as multi-dimensional creation. This means the creation of a product has many different aspects applying to different design stages of the product.

For this research, it has three main tasks: 1) The development of an interactive design environment suitable for product design activities; 2) The creation of a system for generating designs in such environment; and 3) Evaluation of the design examples using the system to validate the methodology and outcome. Within the design environment employing shape grammars, the following roles of professionals and the system are clarified.

A1) Role of Professionals in Various Departments

Professionals in various departments of the enterprise like sales, marketing, industrial design, engineering design, production design, manufacturing, distribution, service and disposal are identified in section 1.1. Professionals give their advices, concerns and specifications to the shape grammar developer and designers. For example, marketing manager discusses the customers' behaviour and trend obtained from a market research with the shape grammar developer and designers. Another example, manufacturing engineers give advices to the shape grammar developer and designers about the regulations, standards and guidelines for manufacturing. In the context of this thesis, such professionals are not directly identified as a part of the system environment since the system mainly supports exploration of initial design concept and forms.

A2) Role of Shape Grammar Developer

Shape grammar developer listens to the professionals, converts their needs and knowledge of designs into different set of shape grammar rules. Besides, a shape grammar developer organises the rules and prepares a starting grammar. This starting grammar works as a knowledge base for supporting design activities.

The tasks of preparing a shape grammar include converting the knowledge of design from each expertise into shape grammar rules, analysing the existing designs, identifying the interrelationships among various features, suggesting multiple ways to describe the features, converting those features into modifiable elements of rules, encoding the knowledge of the existing designs into different sets of rules and determining several ways to apply the rules for a topologically diverse set of solutions.

In addition, the shape grammar developer has to revise the existing shape grammars upon the receiving comments from the professionals including designers and engineers during design process.

A3) Role of Designer

The role of a designer is to specify the design problems upon discussing with various professionals, make decisions to select appropriate problem solving strategies, develop new or refine existing problem solving strategies, evaluate the generated designs by the system, and reflect any unsolved problems to the corresponding professionals and shape grammar developer.

A4) Role of Computational Framework

The role of the computational framework is to create a system which adopts an integration approach of two key computational techniques: shape grammars and evolutionary computing for supporting product design activities. The system evolves the shape grammar rules to generate the designs. The system does all the complicated procedures and calculations to construct the complex 3D models for visualisation. This reduces the designers' time to do the complicated modelling tasks to construct a large number of 3D models from scratch. Also, this allows the designers to concentrate their efforts on performing higher level of design tasks such as evaluation of designs and making decisions.

B) Supporting Design Activities in an Interactive Design Environment

After distinguishing the role of different professionals and the computational framework, this part outlines a possible scenario of using the system in design practice.

In general, the designer can follow a common scenario to use the system to explore designs. For instance, the shape grammar developer prepares a starting grammar, the system evolves grammars that produce designs, the designer evaluates the designs, then the system takes account of the evaluations, and so on, until the designer is satisfied.

After the development of a starting grammar, the second step of constructing the environment is to specify the procedures to be taken by the designers:

- The designers analyse the design requirements arisen from the professionals in various departments;
- The designers convert the abstract design requirements into different evaluation criteria. Some criteria can be numerically evaluated whereas some can only be subjectively evaluated by the designers;
- The designers input those criteria into the computational system;
- The designers run the system.

Upon receiving the instructions by the system, the third step of constructing the environment is to specify the procedures to be taken by the system:

- The computational system starts to run in accordance to the input criteria;
- The computational system generates the results;
- The computational system supports the evaluation of the results by providing numerical analysis and visualisation of complex 3D models.

After generating the results, the fourth step of constructing the environment is to specify the procedures which the designers take response to the generated results:

- The designers evaluate the generated results;
- The designers identify all the conflicts arisen from the requirements and the results;
- The designers report those conflicts to the relevant professionals;
- The designers make decision to refine the requirements and run the system.

The whole procedure repeats from any stage until the results are satisfied to the designers. If the generated results are not satisfactory, it may be caused by the wrong definition of the requirements. In this case, it is determined by the designers that the generated results are satisfied to some criteria while giving up some conflicting criteria. Designers should report those conflicting criteria to the relevant professionals and ask the professionals to consider modifying the requirements if necessary. In any stage, new requirements may come into existence. Designers have to take response and actions accordingly.

Based on this process, it can be seen that the design activities in each stage of design process influence the final results. The understanding of the effects on decision making at each design stage is therefore crucial to the success or failure of product design. Another significant advantage which can be seen from the process is that designers continuously evaluate the designs, select and modify the objective functions during the design process. Since the generated designs are continuously monitored interactively by the designers during the design process, the designs can be improved and refined after many generations. At this stage, the designers can have more choices to select the interesting designs which are generated in accordance to the designers' expectations.

As a result, the design problems are refined from an unclear specification of requirements, through the continuous evaluation and improvement of the solutions, to a more clear specification of requirements. This provides a higher chance for the results generated by the system being more suitable to fit the expectations of the designers than the starting stage.

C) Computational Framework

As stated in section 1.3.1 and 1.3.2, the problem of the balance between stylistic consistency and stylistic innovation, and the problem of the uncontrolled design process with multi-dimensional requirements are discussed in details respectively. The research direction for constructing a computational framework to solve these two problems can be focused on the fundamental issue of how to establish a shape grammar base for a product. The derivation of SG rules relies on applying appropriate methodologies to encode the knowledge of existing designs into SG rules.

To tackle the problems that arise from the analysis of existing designs, a systematic approach in the development of SG is developed. The basic premises of the approach are summarised as follows:

- to classify the components of existing designs from different perspectives: Using digital camera form design as an example, the flash with rectangular shape is stylistically inconsistent with the flash with oval shape. From a geometric point of view, the flash with rectangular shape should be classified in one group different from the flash with oval shape. However, from the functional point of view, these two flashes can be classified into the one group since both perform the same functions.

- to define the maximum and minimum boundary of geometry for each component: The components can have different discrete geometric attributes in the spatial geometry but the value of each attribute will not exceed its own maximum or minimum boundary of geometry.

- to specify the constraints of spatial relationships between any two components: A configuration plan is set up based on the spatial relationships between any two components. Each component is allocated in the spatial geometry in accordance to the configuration plan. The position of a component is justified by the constraints of the spatial relationships among the components.

- to identify evaluation criteria from different perspectives: The evaluation criteria of existing designs can be sought out from the designers, customers and manufacturers in advance. Then the designs can be evaluated with respect to individual criterion. For example, the smaller shell volume of the exterior of a digital camera uses less material to manufacture which in turn reduces the manufacturing costs. However, the designers may not be visually attracted to the design. As a result, the evaluation value of the shell volume is high whereas the artificial selection value is low. The designs can also be evaluated through the combination of these two criteria using a weighting method. By adjusting the weighting factors of each evaluation criterion, the designers can determine which evaluation criterion is significant. Based on the evaluation results, the designers can determine which features should be encoded in the SG rules.

This approach symmetrically analyses the existing designs and organises the analysed results in several ways for further processing in the development of SG. For example, the form features of products in particular design applications are abstracted from existing designs. These form features representing different components are categorised according to their spatial relationships among components based on geometric locations in the assembly. The form features are categorised into different groups. Parametric 2D and 3D SG rules with labels are established for the generation of form features and their corresponding geometric locations in the assembly. A completed methodology related to the above issues in the development of SG will be described in chapters 4, 5 and 6.

1.5.3 Integrating Shape Grammars with Evolutionary Algorithms

In order to exploit the potential generative capability of the existing SG rules, this thesis explores a computational framework which incorporates an evolutionary algorithm into an IGBDS in product design domain. The evolutionary IGBDS enhances the generative capability of the existing SG by using an evolutionary algorithm as an adaptation mechanism for evolving alternative SG rules. The alternative SG rules evolve to generate designs that fulfil new requirements and constraints. The adaptation mechanism monitors the evolutionary process by objective function and control strategies which direct the evolving SG rules to obtain specific design characteristics.

Two prototype evolutionary IGBDS using genetic algorithm (GA) and genetic programming (GP) as the core evolutionary algorithms are developed to manipulate the SG rules. Both prototype systems share the same computational framework. The differences between the first and second prototype include the representation issues in utilising the power of genetic and SG representations. The genetic representation of the first prototype system facilitates the GA to modify the shapes in the parametric two dimensional (2D) SG rules with labels by Boolean operations of new shapes. The genetic representation of the second prototype system facilitates the GP in ease of manipulation of the parametric three dimensional (3D) SG rules with labels under a control environment. Another difference between the two system prototypes is the manipulation issues in evolving the SG rules. The adaptation mechanism of the first prototype system facilitates the GA to monitor the evolutionary process by artificial selections. Whereas the adaptation mechanism of the second prototype system facilitates the GP to monitor the evolutionary process by multi-objective functions which include artificial selections, geometric evaluation functions and volume estimations as well as specific control strategies.

The conceptual framework consists of two key elements: 1) SG as the knowledge for design and 2) Evolutionary algorithm as the generative and adaptation mechanisms, as illustrated in Figure 1.2. For the first element, the development of SG is described in the previous section. For the second element, using the second prototype system as an example, GP creates computer programs to modify the SG rules. The evolving programs are directly represented in the chromosome as trees. The evolving programs consist of terminals and functions which are predefined at the beginning of the evolution. The evolving programs are assigned to different control strategies. In the control strategies, the effects of the terminals and functions are controlled during evolution.

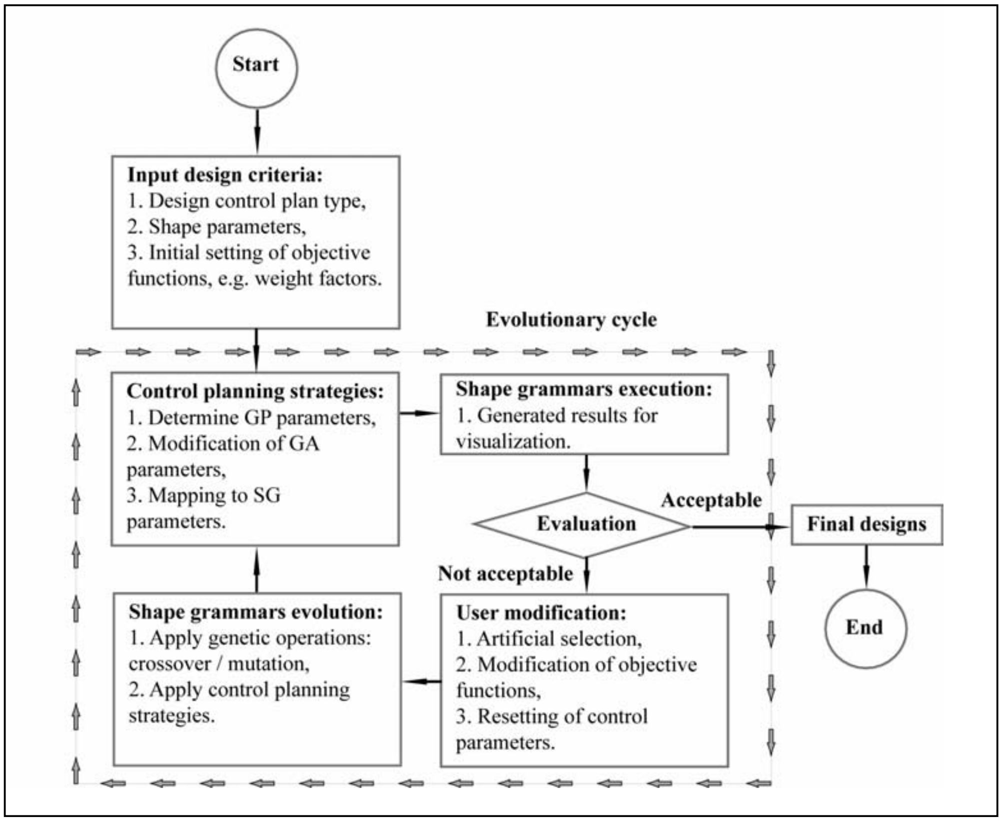


Figure 1.2 Integrated SG and evolutionary algorithm framework

At the beginning of running the system, the designers first input a set of design criteria by specifying design control plan types, shape parameters and initial setting of objective functions, e.g. weight factors. It then enters the evolutionary cycle. Based on the designers' inputs, the selected control strategy determines the GP parameters.

The GP parameters then modify the GA parameters which in turn map to the SG parameters. The SG implementation module then generates the actual design shapes based on the SG parameters. The actual design shapes are evaluated by the evaluation module.

If the results are not satisfactory, the designers can intuitively select the better designs, modify the objective functions and/or reset the control parameters. Genetic operations such as crossover and mutations will then be used to evolve the SG in accordance with the control strategies. Another evolutionary cycle starts and repeats until the satisfactory results emerge or maximum generations are reached.

The evolutionary IGBDS provides a new paradigm of SG based design in which the potential generative capability of SG rules is exploited. Control strategies are developed incorporating objective functions to facilitate the evaluation of designs. Whereas objective functions focus on general requirements, control strategies focus on specific requirements in an evolutionary system.

The implementation of the integrated SG and evolutionary algorithm framework involved several technical problems which include:

1) integration issues: One of the key issues in integrating a highly detailed SG and evolutionary computing system is that the random modification properties of evolutionary computing and the capturing of style properties of SG are in conflict. The random modification of product form designs removes the style of the product. More conflicts will occur if one combines different SG rules to derive new shapes,

2) representation issues: A new representation scheme should be developed in order to utilise the power of genetic and SG representations in encoding the shape features of the products, and

3) manipulation issues: A new adaptation mechanism which combines objective functions and control strategies should be developed in order to monitor the evolutionary process of the SG rules.

The solutions to these technical problems related to the above issues in the exploration of designs using this framework will be described in chapters 5 and 6.

1.6 Significance and Potential Benefits

Applications of SG in engineering and product design domain initially are developed to automate the design process supported with expert knowledge obtained from experts in design, manufacturing and all related areas. Most of the successful SG applications developed depend on useful and high quality SG rules specific to the applications. A systematic approach in deriving the SG rules from the analysis of existing designs has yet to be developed as difficulties arise from knowledge acquisition methods in both theory and practical experience in designing.

The analytical approach of deriving the SG rules must therefore be able to strategically organise the completed design information of existing designs from different perspectives. A complex information network of SG can then be established. The complex information network of SG, however, may not fully represent the knowledge of designs from the original design concepts. To a certain extent, the complex information network of SG can reveal part of the knowledge from original design concepts.

There are still many critical issues to be tackled in modelling the design information in a complex information network of SG. The complex information network of SG is useful in the derivation of high quality and useful SG rules for specific design applications. Potential benefits from this approach include the enhancement of the generative capability of SG. More alternative designs can be generated from the evolutionary IGBDS than the traditional IGBDS. A more detailed description of the contributions from this thesis to the field of SG approaches to design is discussed as follows:

1.6.1 Systematic Approach in the Development of Shape Grammar

The systematic approach developed in this research improves the capability of modelling the design information of existing designs. This systematic approach contributes to the enhancement of the development process of SG. With the support of classification methods of components, specification methods in defining design spaces for components and specifying constraints of spatial relationships among components, and identification methods of evaluation criteria, the design information for existing designs can be strategically modelled into a complex information network of SG from different perspectives.

The systematic approach also defines a new way to encode the SG rules in terms of individual component construction and its position in the overall configuration of a product. Together with all the relevant information such as functional descriptions, geometric properties, spatial relationships, constraints and evaluation criteria modelled in the complex information network of SG, the SG rules can be derived to process information related to individual component construction and configuration design. The designers can determine the parameters of SG rules and order sequences of rule executions in component and configuration design based on the complex information network of SG.

1.6.2 Extending the Generative Capability of Shape Grammar

The development of the evolutionary IGBDS utilises the power of genetic and SG representations. The representation of SG depicts the modification process in terms of shapes specified in the left and right sides of SG rules. The representation of SG can be viewed as abstracted knowledge representation since SG captures the knowledge of designing by means of shape modification processes specified in the SG rules. Encoding the SG rules into genetic representation allows the evolutionary process to perform on the SG rules. The genetic representation of SG rules can therefore assist the modification of SG rules to adapt new design requirements. As a result, exploitation of potential generative capability of the SG rules can be achieved. This also implies that the complexity of the design information is growing during the evolutionary design process.

The development and experiments of two prototype evolutionary IGBDS verify the potential benefits of the integrated SG and evolutionary algorithm approach to designs. The two prototype evolutionary IGBDS illustrate the flexibility in using parametric 2D and 3D SG with labels to explore a wide range of engineering and industrial design problems. These design problems involve complex and intuitive conception of form generation.

As a whole, this research demonstrates a systematic approach in deriving the SG rules in which the SG rules are interrelated to form a complex information network of SG. The generative capability of the SG rules can be further exploited by the evolutionary algorithms in order to fulfil new design requirements and constraints. The designers can justify different evaluation methods by varying the weighting factors, determining the control parameters for the SG rules, selecting appropriate control strategies for specific design characteristics, controlling order sequences of rule executions and exploring form and configuration designs, in engineering and product design domains.

With the framework developed in this research, it is possible to improve the development process of SG based design applications. The analysis process decomposes the complex composite design characters of designs by recognising essential and identifiable design characters. By grouping the design characteristics of the designs from different perspectives into SG rules, richer information can be represented by the SG rules. Both the shape grammar developer and the designers can have more understanding of the design process by considering the designs from different perspectives. More opportunities are opened for the designers to participate in the determination of setting control parameters for the SG rules by the evaluation of the designs from different perspectives.

For example, in the case of the form design of a product composed of components which are configured in specific geometric positions on the product, different sets of SG rules can be derived based on the stylistic design characteristics and spatial relationships among components. The SG rules can be further encoded as genetic representation for the evolutionary algorithm to evolve alternative SG rules.

The designers can apply the existing sets of SG rules to generate designs for good visual appeal or ergonomic requirements. Or the designers can explore new designs using the evolved SG rules to fulfil new requirements. Either way the designers can justify ways to evaluate the designs from different perspectives at each stage of design process. The benefit of this approach allows the designers to realise more issues regarding the design problem from different perspectives than tackling the design problems from a single viewpoint.

Furthermore, the integral visualisation process makes transparent the various alternative viewpoint or weighting results. The framework developed in this research has been implemented with two prototype systems. Experimental results obtained from the evaluation of this platform and the platform itself provide a solid background foundation for further research in the field.

1.7 Research Methodology

Since this research can be divided into two parts: 1) Systematic approach in the development of SG and 2) Extending the generative capability of SG, a research paradigm tailored for effective justification on the mutual effects on each part of the research outcome is made. The research paradigm is composed of three key stages: conceptualization, implementation and evaluation.

The conceptualization stage includes:

1) Problem identification and clarification, and

2) Theoretic modelling, hypothesis and presumed solutions.

The implementation stage includes two prototype systems for both parts of the research methodology. The evaluation stage includes different testing plans to systematically evaluate the designs generated by the two prototype systems. Each of the above stages is further discussed in detail in the following sections (Section 1.7.1 to 1.7.3).

1.7.1 Conceptualization Stage

A) Problem Identification and Clarification

This project starts from the fundamental issues concerned with the development of SG and reveals the problems in deriving the SG rules. The causes for the problems have been carefully identified from different viewpoints. The development process of SG has been analysed in detail in order to develop understanding the nature of the difficulty in IGBDS.

B) Theoretic Modelling, Hypothesis and Presumed Solutions

Since the modelling of design information of existing designs, the methods of encoding the design information into SG rules and the ways to derive the SG rules are key issues for the development of successful SG based design applications, a systematic approach to derive SG rules is established. The systematic approach specifies strategic analysis methods of designs from different perspectives. Classification, specification, and identification methods are involved to analyse the designs and the analysed results are organised to build a complex information network of SG.

Another key issue of this research emphasises the importance and necessity of enhancing the generative capability of IGBDS in the exploration of designs. By integrating an evolutionary algorithm to the IGBDS, SG rules are modified to adapt to new design requirements during the evolutionary process. This relies on careful planning to retrieve significant elements from the complex information network of SG and encode all the necessary information into genetic representation.

1.7.2 Implementation Stage

The implementation stage includes system realization for two implemented systems: two prototype systems are based on the theoretic strategies identified in the framework development at the conceptual stage.

A) Experimental Case Study

Prior to building the two prototype systems, an experimental case study to redefine a Chinese lattice SG for generic pattern designs has been reviewed. This arrangement serves two purposes: one is to serve as an example in illustrating the application of a general non-parametric 2D SG for pattern designs; and the second is to demonstrate how a specific SG can be modified for more generic design applications. The experience gained from this experimental case study is helpful in developing the completed proposed systematic approach for the construction of the two prototype systems. The evaluation of the implementation results obtained from this experimental study establishes an understanding of the critical issues when applying the systematic approach in the development of SG. Lack of understanding of such critical issues is fatal to the logical formulation of a solid foundation for the construction of the two prototype systems.

The experimental study focuses on the development of an IGBDS for pattern designs. One of the Chinese lattice designs is chosen as an example for analysis. The original grammar rules particular for this example developed by Stiny (1977) are first introduced. With reference to these grammar rules, a new set of grammar rules are developed based on symmetric properties. The new grammar rules are designed to accommodate the preference of designers. The system is developed using the Visual Basic Application (VBA) supplied within the AutoDesk Inventor environment. Since Inventor allows software developer to use the "Application Programming Interface" (API) to customise the applications, the system can be potentially used as a plug-in product to AutoDesk Inventor platform.

B) Two Prototype Systems

After the experimental case study has been reviewed with regard to critical issues in developing the SG, the understanding established in the experiments carried out in this experimental case study forms a solid foundation for the construction of the two prototype systems. Based on the understanding established, theoretical analysis and experiences gained from applying part of the systematic approach in the experimental study, the formulation of the framework using evolutionary algorithms as adaptation mechanisms in the exploitation of the generative capability of SG rules is established. Two prototype systems incorporating an adaptation capacity into an IGBDS are then developed. The two prototype systems focus on both objectives: 1) developing the SG and 2) exploration of designs with integration of evolutionary algorithms and the SG.

B1) The First Prototype

The most widely used evolutionary algorithm adopted by researchers in engineering and product design areas of research is genetic algorithm (GA), which demonstrates the principle of evolution and survival of the fittest (Holland, 1975; Davis, 1991). The GA is adopted as the core evolutionary architecture in the construction of the first prototype system to evolve the SG rules in the exploration of new designs.

The first prototype system focuses on the development of parametric 2D shape grammars with labels enhanced by evolutionary computing for supporting new product form designs. The forms of products, with digital cameras as examples, are analysed to derive shape features in the form of SG rules. The rules are then encoded as the code scripts of a GA in order to generate new SG rules. The results generated by the GA define a new combination of shape features for alternative designs. In this way, traditional SG is extended to an interactive context in which generative and evolutionary computing methods are combined. Both product component design as well as product configuration are supported in this framework. The prototype system is developed as a stand-alone application using Visual Basic and linked in the OpenGL programming environment.

B2) The Second Prototype

Since the first prototype system employs parametric 2D shape grammars with labels to generate 3D shapes, the modelling capability of free form shapes is limited. Another limitation of the first prototype system is that only artificial selection is adopted as the evaluation criterion. In order to further improve the performance of the framework, the second prototype system is developed with parametric 3D shape grammars with labels for more powerful capability in dealing with sophisticated free form generation. Other improvements include the introduction of control strategies to monitor the evolutionary process under specific requirements. Together with the introduction of multi-objective functions to facilitate the evaluation from different perspectives, the evolving SG rules can flexibly adapt to generate designs for new requirements.

The second prototype system focuses on the development of parametric 3D shape grammars with labels enhanced by evolutionary computing with a control mechanism for supporting new product form designs. With such an evolutionary IGBDS, a set of existing products in a particular product design domain is first analysed to derive shape features in the form of SG rules. The SG rules are created with 3D labelled shapes. All the SG rules are then encoded as the code scripts of genetic representations for the generation of alternative SG rules. In order to systematically evaluate the designs and the corresponding SG rules, control strategies named "GP-GA-SG" are developed to facilitate the evaluation of the generated results. Both product component designs as well as product configurations are supported in this framework. A prototype system for this framework has been implemented with examples of generating a particular type of product form design. The implemented evolutionary IGBDS uses genetic programming (GP) to evolve desirable designs and the corresponding SG rules. The second prototype system has been developed using Visual C++ and ACIS 3D modelling kernel, and tested with different experiments.

1.7.3 Evaluation Stage

Experiments have been conducted to study the flexibility of the framework in tackling different design problems related to the product design domain. The theoretical propositions of the systematic approach in the development of SG and extension of the generative capability of the SG have been carefully verified empirically by the two prototype systems at implementation level. The results obtained from the two prototype systems are further analysed numerically to determine the performances of the implemented evolutionary IGBDS.

1.8 Overview of the Thesis

This thesis is divided into five parts. Part one (Overview) consists of an introduction chapter which presents an overview of this thesis.

- Chapter 1 (Introduction) gives an introduction of the overall framework, which outlines the whole research background and themes. First the SG based design method is introduced and certain key problems in engineering and product design domains are identified. Then the clearer research objectives are defined. An outline of the research proposition is given based on the main research objectives. Finally, research methodological issues in the development of the integrated SG and evolutionary algorithm framework are discussed.

Part two (Review of Related Work) reviews work related to this research, and consists of two chapters that discuss the main areas of research upon which the integrated SG and evolutionary algorithm framework is based.

- Chapter 2 (Shape Grammars) introduces SG based design approaches to generate designs in different design domains. Both theoretical development and practical applications of SG are discussed. In particular, a number of integrated evolutionary algorithms and SG based design systems are described.

- Chapter 3 (Evolutionary Algorithms) provides an overview of evolutionary computation. A review of related research and recent development of evolutionary designs in various design domains, and in particular engineering and product design domain is given.

Part three (Theoretical Development) presents the theoretical development of the integrated SG and evolutionary algorithm framework using the systematic approach, and consists of one chapter.

- Chapter 4 (The Theoretical Framework) describes the key procedures of the systematic approach in developing the evolutionary IGBDS with two key elements: the parametric SG and the evolutionary architecture. This chapter provides an overview of the systematic approach in deriving the SG rules. The key procedures of the systematic approach such as 1) The construction of an information network of SG, 2) The construction of a Core Variant Model, 3) The development of parametric SG and 4) The construction of an evolutionary architecture are described. The detailed implementation of the systematic approach is illustrated with two prototype systems which will be described in the next two chapters.

Part four (Implementation and Analysis) presents a complete development process of the two prototype systems using the systematic approach. This includes the theoretical development, implementation and analysis of the integrated SG and evolutionary algorithm framework, and consists of two chapters:

- Chapter 5 (Parametric 2D Shape Grammars) describes the first prototype evolutionary IGBDS which employs genetic algorithm (GA) as the core evolutionary architecture. First, the systematic approach in supporting the development process of a parametric 2D SG for digital camera form design is described. The parametric 2D SG is used in the construction of an IGBDS. Second, the construction of an evolutionary architecture used for the integration to the IGBDS is described. This is the key feature of this system to apply the evolutionary techniques for evolving new SG rules. Third, the methodologies of the formulation of the new SG rules using the evolutionary techniques are described in detail. Fourth, the system development of the first prototype system is described. In the first prototype system, the new SG rules are derived based on the systematic approach and the artificial selection evaluation method, and used for the exploration of new designs. Finally, the implementation results of the first prototype system for digital camera form designs are presented. Interactions among the designers and the system are also described.

- Chapter 6 (Parametric 3D Shape Grammars) describes the second prototype evolutionary IGBDS which employs genetic programming (GP) as the core evolutionary architecture. The second prototype system consists of parametric 3D SG with labels which are enhanced by evolutionary computing with control mechanisms for supporting new product form designs. New key features such as new genetic representation and control strategies have been developed in the second prototype system for technological advancement in deriving SG. This second prototype system strategically applies the control strategies and multi-objective functions to control the evolving SG rules for the generation of new designs with particularly desired design characteristics.

  First, certain problems and weaknesses associated with the first prototype system are highlighted. In order to solve these problems, this second prototype system extends the generative capability of the first prototype system by introducing the parametric 3D SG for free form design generation.

  Second, the development of the evolutionary architecture for the second prototype system is described. In this evolutionary architecture, new genetic representation of the evolutionary algorithm called "GP-GA-SG" interface of phenotypes and genotypes is developed to utilise the power of genetic and SG representation, and is described. Also, the methodology for deriving the control strategies to manipulate the new genetic representation, and the evaluation methods of designs using multi-objective functions are described.

Third, the system development of the second prototype system is described. The key features of this system are highlighted and include the technological details of the implementation for the newly developed genetic representation and control strategies. With such an evolutionary IGBDS, both product component designs as well as product configuration designs are supported.

Fourth, the implementation results of the first experiment using the second prototype system for digital camera form design are presented. Also, the evaluation of the implementation results is presented, and the application of the first control strategy is demonstrated in the first experiment.

Finally, the implementation results of the second experiment and the evaluation of the implementation results are presented. The evaluation of the generated designs depicts a clear picture of the complex effects produced by the multi-objective functions and control strategies in modifying the SG rules. Certain key features of the newly developed genetic representation and control strategies are highlighted and tested in the second experiment.

Part five (Conclusions) consists of one concluding chapter which presents conclusions and future work.

- Chapter 7 (Conclusions and Future Work) presents the findings from this research and identifies the key contributions made from the findings. Possible directions for future research in this area and several related issues are briefly discussed.

Part II Review of Related Work

Part two (Review of Related Work) reviews the work related to this research. This part consists of two chapters that discuss the main areas of research upon which the integrated SG and evolutionary algorithm framework is based.

- Chapter 2 (Shape Grammars) introduces SG based design approaches to generating designs in different design domains. Both theoretical development and practical applications of SG are discussed. In particular, a number of integrated evolutionary algorithms and SG based design systems are described.

- Chapter 3 (Evolutionary Algorithms) provides an overview of evolutionary computation including review of related research and the recent development of evolutionary design in engineering and product design domain.

Chapter 2

Shape Grammars

2.1 Introduction

This chapter reviews the working principles, computational issues, basic technical mechanisms, applications and all the related properties and issues of SG. The study of using SG to design has a long history over three decades starting with the introduction of SG formalism described by Stiny and Gips (1972). A more comprehensive illustration of the theoretical formalism of SG was presented in the Stiny's paper (Stiny, 1980a). Recently, the discussions of the formal process of design using SG and the clear illustrations on how to embrace ambiguity, and build on its expressive power in generating an infinite complexity of designs are presented in his recent book (Stiny, 2006).

This chapter consists of five main sections:

- In section 2.2, an overview of SG in supporting the design activities is described.

- In section 2.3, the working principles of SG and the basic technical mechanisms of SG are discussed.

- In section 2.4, the computational issues in the development of SG and the SG properties are discussed.

- In section 2.5, the methodology in converting a specific SG to a more generic SG for the generation of generic designs is discussed.

- In section 2.6, the historical development and the current state-of-art for the development of SG applications are reviewed.

2.2 Overview of Shape Grammars

SG is one of the artificial intelligence techniques used extensively by researchers in representing, reasoning and generating designs. The widespread use of grammatical approaches to design began with the introduction of SG by Stiny (1980a). Stiny (1980a) developed the definitions for SG formalism. In the definitions, the production of new shapes is achieved by executing a set of rules which govern the transformation process of shapes. The new shapes that exhibit certain types of emergent behaviour that are not explicitly created, are generated by combining and substituting shapes under the guidance of the transition rules (Stiny, 1994).

Historical Development of Shape Grammars

The initial formulation of SG was first introduced by Stiny and Gips (1972) in which the configurations of shapes were generated by representing abstract paintings and sculptures. Further refinement and generalisation of the SG approach to design was conducted by Gips (1975) and by Stiny (1975, 1980a). The theoretical foundations of SG have been discussed through a comprehensive overview by Knight (1994a). A comprehensive review more specific to designing engineering SG was given by Cagan (2001). Another comprehensive survey which compared the development processes, application areas and interaction features of different SG approaches was given by Chase (2002). More details related to the historical development of SG and the current state-of-art in developing SG are discussed in Section 2.6.

2.3 Working Principles of Shape Grammars

The working principles of SG can be depicted by three main activities: 1) Formulation, 2) Representation, and 3) Manipulation of SG.

2.3.1 Formulation of Shape Grammars

SG rules are developed to capture the knowledge about how practical design methods can be applied to manipulate and combine various physical components. Using SG based approach in a design application the first step is to acquire the knowledge of designing either for a particular design application or generic applications. The second step is to formulate a suitable representation for the expression of design knowledge. The creation of the knowledge bases in terms of SG can be achieved by three major ways: 1) Developing design theories, 2) Capturing what successful designers know or their practical experiences in designing the objects, or 3) Analysing the existing objects in particular application domains. These knowledge bases of SG compose the facts and SG rules. Whenever there are facts of the specific design situation which match the facts in the knowledge bases of SG, SG will apply automated reasoning procedures to these facts and execute the corresponding SG rules to solve the design problems. Usually, the facts are the shapes, labels, parametric values and other related attributes defined for the specific SG rules.

2.3.2 Representation of Shape Grammars

There are many possible ways to describe a design with certain representation methods. How a design is represented in computational formats can be interpreted by a particular computational methodology. This is a critical computational issue. In SG, the designs are represented as the shapes and their attributes which can be interpreted by a SG. A SG consists of a vocabulary of shape elements, a set of production (or transition) rules and an initial shape. Since the designs are generated by the SG rules, the representation of the SG rules is also a critical computational issue. The SG representation is formulated for the expression of the captured design knowledge, which is composed of shapes, labels, spatial relationships among shapes and other attributes.

Basically, referring to Stiny's definition of shapes, shapes are made of basic elements which are of zero (point), one (line), two (plane) or three dimensions (solid) (Stiny, 2006). Common basic elements like labels, lines, planes and solids are used in the description together with the linear relationships of coordinate geometry and spatial relations among shapes. Other basic elements like curves, curved surfaces or curved solids can be used to extend this repertoire of the vocabulary of shapes.

Table 2.1 summarises the key properties of basic elements using maximal representation (Stiny, 2006). The table shows the basic elements and their corresponding attributes like dimensions, boundaries, contents and embedding properties. The basic elements include point, line, plane and solid. The corresponding dimensions of such basic elements are of 0, 1, 2, and 3 respectively. With maximal representation, points can not be divided into smaller elements. For those elements with dimensions higher than 0 such as line, planes, and solids, they can be decomposed into many discrete elements like line segments, triangles, and tetrahedrons. A common boundary element such as a point, a line (edge), or a plane (face) is formed between any two of these successive elements.

Basic element	Dimension	Boundary	Content	Embedding
Point	0	none	none	identity
Line	1	two points	length	partial order
Plane	2	three or more lines	area	partial order
Solid	3	four or more planes	volume	partial order

Table 2.1 Properties of basic elements (Table is redrawn from (Stiny, 2006, p. 164))

Algebra U_{ij}	U_{i0}	U_{i1}	U_{i2}	U_{i3}
U_{0j}	U_{00}	U_{01}	U_{02}	U_{03}
U_{1j}		U_{11}	U_{12}	U_{13}
U_{2j}			U_{22}	U_{23}
U_{3j}				U_{33}

Table 2.2 Shapes in algebras (Table is redrawn from (Stiny, 2006, p. 180))

Algebra	Basic elements	Boundary shapes	Number of parts
U_{0j}	points	none	finite
U_{1j}	lines	U_{0j}	indefinite
U_{2j}	planes	U_{1j}	indefinite
U_{3j}	solids	U_{2j}	indefinite

Table 2.3 Some properties of shapes (Table is redrawn from (Stiny, 2006, p. 196))

U_{ij}	Every shape has a distinct nonempty part – there is no smallest shape	Every shape is a distinct part of another shape – there is no largest shape
$0 = i = j$	no	no
$0 = i < j$	no	yes
$0 < i \leq j$	yes	yes

Table 2.4 More properties of the part relation (Table is redrawn from (Stiny, 2006, p. 196))

Table 2.2 summarises the algebras used in the maximal representation (Stiny, 2006). The table shows the algebras of the shapes which are defined by three main elements. The first element is the shapes themselves which are created with the basic elements. The second element is the part relationships among shapes which include the Boolean operations. The third element is the Euclidean transformations. Usually, three-dimensional geometry is sufficient to represent designs and enumerate the algebras for shapes.

Table 2.3 and 2.4 summarise the properties of basic elements in relation to the algebras for shapes using maximal representation (Stiny, 2006). The tables organise the algebras for shapes with respect to two numerical indices i and j.

The algebras for shapes are defined in accordance to Stiny's definition: "In an algebra U_{ij}, the left index i determines the dimension of the basic elements, and the right index j is the dimension in which those basic elements are combined in shapes and in which the transformations are defined. … Evidently, i is greater than or equal to zero, and j is greater than or equal to i." (Stiny, 2006, p.181).

Further development which builds upon earlier work in the area of representing design parts and their descriptions using well-defined algebraic representations for shapes (Stiny, 1994) has been made by Earl (1999). The advantage of this enhanced representation scheme is representational flexibility. No explicit structure and no differentiated subparts are required on their own shapes that make up the design. Final product properties and behaviour can be reflected by the shapes themselves. The shapes can be structured according to the kind of rules that will be applied to them. The relationships among compositions of parts of a design represent a structure of design descriptions. The closure properties of these descriptions and some of the formal tools can be used in constructing design descriptions.

2.3.3 Manipulation of Shape Grammars

A) Embedding Working Device

In SG, the embedded working device allows designers to interpret the shapes differently using maximal representation. This shows another property of SG in which shapes are ambiguous without predefined basic elements. The condition of embedding one basic element in another occurs whenever every basic element contacts these two elements at the same instance. If the embedding is restricted to individual symbols, patterns, structures, elements or shapes, the shapes are defined with identities. Embedding devices treat the shapes as individual symbols for all the longest lines of shapes except the elements in sets. Embedding can be classified into three types: 1) Reflexive, 2) Antisymmetric, and 3) Transitive (Stiny, 2006, p.167).

B) Substitution Mechanism

The syntax of the language, i.e., the SG, defines the transformation methods for the elements in the vocabulary of shapes to be combined. Constraints can be applied to the relationships among the elements in the transformation process. The production (or transition) rules transform the initial shape into new shapes through these four main processes. The transformation methods specify that the execution of a SG rule goes through four main processes: 1) Recognition, 2) Transformation, 3) Matching, and 4) Replacement processes. Since the matching process controls the first two processes, the matching process is more complex than other processes. The transformation methods can be viewed as a substitution mechanism, which starts with an initial shape and repeatedly substitutes the parts of this shape with new parts in generating new designs. The substitution mechanism performs the key operations on the SG rules which involve four main operations: 1) Recognition, 2) Transformation, 3) Matching, and 4) Replacement operations on the geometry of the individual shapes. Usually, the transition SG rules are manually selected and applied by human designers, rather than indiscriminately applied, through the 2D or 3D substitution mechanism.

C) Transformation Process

The transition rules can be divided into two parts: the left-hand side shapes and the right-hand side shapes, in which the shapes are specified in the vocabulary for shapes. The left-hand side shapes refer to the antecedent 2D or 3D arrangements of the shapes specified in the vocabulary whereas the right-hand side shapes the consequent arrangements of these shapes. In the 3D case, the verification process in matching between a particular rule and the current form tends to be much more complex than in the 2D case. In comparing the left-hand side shapes of a particular rule to the shapes in the current form, the arrangements of both shapes would be the same provided that one arrangement is a rotation, translation, scaled and/or reflection version of the other arrangement. Knight (1994a) and Flake (1998) have described such common types of transformations as Euclidean transformations or affine transformations. Similar to the verification process in matching, performing these transformation processes of the shapes for the 2D case is feasible and much more simply than in the 3D case.

SG is used to describe such transformation processes of shapes starting from an initial shape, to the ending shapes which are determined either by the designers or computers, or to whenever the SG rules can be applied to modify the evolving shapes. In other words, the design objects are evolving when there are SG rules continuously applied to modify the design objects. The matching operation controls the recognition and transformation operations in which searching is performed by recognising and transforming the shapes of the left hand side of the SG rule which match the shapes in the current form. After the matching process, the identified shapes in the current form would be cut out and replaced with the transformed shapes of the right hand side of the SG rule.

D) Mathematical Illustration

For the recognition process, the key operation is to search any part of or the whole Target shape T which contains similar shapes of the left hand shape L. Suppose that there exists any shape C which could contain basic elements such as points, line, planes, solids, labels, weights, etc, and specify how these elements are combined. For example, a SG rule is specified with the specification: $A \rightarrow B$ which means that the shape $t(B)$ is substituted for the shape $t(A)$ in C whenever $t(A) \leq C$. The transformation shape $t(A)$ refers to a transformation t applies to shape A . The specification of $t(A) \leq C$ means that the transformation shape $t(A)$ is part of (embedded in) C . After the application of the SG rule, the resulting shape is specified by $(C - t(A)) + t(B)$. The shape $t(A)$ is recognised in and cut away from the shape C , and this missing part is replaced with the transformation shape $t(B)$.

E) Schemas

An extension to the formulation of SG rules with the specification $A \rightarrow B$ applies schemas. Schemas are like SG rules with the addition of variables, assignments, and predicates. Suppose that there exists any shape C which could contain basic elements such as points, line, planes, solids, labels, weights, etc, and specify how these elements are combined. For example, a SG schema is specified with the specification: $x \rightarrow y$ which means that x and y are variables which take shapes as parameters. A predicate may be applied to restrict an assignment g such that the rule is defined in accordance to the predicate: $g(x) \rightarrow g(y)$. In this way, the rule is defined flexibly such that the parameters of the variables x and y are simply shapes with rules with certain prescribed properties. For example, $y = x$ defines identities for the free form shapes x and y ; $y = t(x)$ defines y as a transformation t of x .

F) Recursion Mechanism

The significant advantage of using schema is that the rule can be applied recursively whenever there are a part like x or $t(x)$. The substitution mechanism substitutes the shape $t(g(x))$ in C with $t(g(y))$ whenever $t(g(x)) \leq C$. The transformation shape $t(g(x))$ refers to an assignment g and a transformation t apply to shape x . The specification of $t(g(x)) \leq C$ means that the transformation shape $t(g(x))$ is part of (embedded in) C . After the application of the SG rule, the resulting shape is specified by $(C - t(g(x))) + t(g(y))$. The shape $t(g(x))$ is recognised in and cut away from the shape C, and this missing part is replaced with the transformation shape $t(g(y))$. The new shapes are evolved by recursively applying the transition rules to their sub shapes using the schemas. A final shape emerges from the new generated shapes if it satisfies the design requirements.

2.4 Computational Issues

2.4.1 Generative Capability of Shape Grammars

The generative capability of SG depends on the usefulness and quality of SG which in turn depends on various factors. These factors include the richness of knowledge captured in the SG, and, when, how, in what conditions and for what applications to apply the SG. For example, a simple SG is useful and of high quality when it is applied to the architecture domain but is useless or of poor quality in the product design domain. Also, the knowledge captured in SG is limited by knowledge acquisition methods in several ways. Considering the most common knowledge acquisition method is to analyse existing designs, only typical samples of the existing designs are selected as representative cases. This small scale of sampling does not cover all the special design characteristics from all the existing designs. Besides products are rapidly evolving to suit market trends and customers' needs on a day-to-day basis. Consequently, the knowledge base established by the analysis of the existing designs can never be complete and accurate.

Therefore, this thesis proposes a systematic approach and a computational framework for systematically deriving SG with knowledge obtained from the analysis of existing designs as well as designers' experience and creative capability. This will be introduced in later chapters (Chapter 4, 5 and 6).

2.4.2 Emergent Properties of Shape Grammars

A SG consists of a number of properties which are distinct from other artificial intelligence (AI) techniques with regard to having non-deterministic, ambiguous, embedded and emergent properties. The emergence of shapes is one of the fundamental properties of SG, which reveals the creativity defined by the calculation of shapes, using the right types of SG rules, and recursion and embedding technical devices of SG. The emergent properties of SG are therefore based on the types of SG rules used, and the recursion and embedding technical devices of SG which in turn relate to or are somehow dependent on the ways in when, how and what to apply with the SG rules. This section discusses the emergent properties of SG.

Concept of Sub-Shape

The emergent properties of SG involve the formulation of a sub shape concept such that one shape may be a sub shape of another. Take an example of a "X" shape which is composed of two lines with the same length. There are left-hand side shapes of a SG rule which specify the antecedent arrangement of "X" shapes formed by two lines with different length. In SG, the rule can be executed even if the antecedent arrangement of the “X” shape is not a Euclidean transformation of the current form. This is achieved by the conceptual formulation of the sub shape working principle (Stiny, 1980a). Whenever one arrangement of shapes, no matter whether they belong to 2D or 3D, line or curved solid, is completely contained within another arrangement of shapes, the former can be regarded as a sub shape within the latter. In this case, a transition SG rule can be executed flexibly whenever the matching conditions are fulfilled by either identical arrangements of shapes, or sub shapes of some arrangement of shapes. This special arrangement of shapes lies in the sub shape concept forms as part of the emergent properties of SG which exhibits certain types of emergent behaviour (Stiny, 1994). Since this sub shape working mechanism is extremely complex in implementation due to the complex task of matching rules to forms, most of the implementations are limited in preliminary stage and the true power of SG is not fully realised in real applications. For practical applications, researchers tend to use set grammars which are a sub set of SG for real applications in engineering and product design domains.

2.4.3 Classification of Shape Grammars

SG can be classified into different types in accordance to their corresponding points of views. For instance, there are six types of SG: basic grammars, nondeterministic (ND) basic grammars, sequential grammars, additive grammars, deterministic grammars, and unrestricted grammars classified in accordance to the generative capability of SG (Knight, 1999a). Other types of SG can be classified in accordance to different application domains. For example in product design domain, two types of SG: 1) Construction and 2) Configuration SG can be derived to model the generation process of components and their configurations respectively. The SG can also be classified in accordance to the technical point of view, for instance non-parametric and parametric SG.

If non-deterministic SG rules are used to generate the design objects for a particular application, then the generated design objects do not have predefined hierarchically organised list structures. The design objects so generated can be creative when applying the non-deterministic SG rules in various sequences under alternative transformations. By using the recursion mechanism and embedding technical device, the non-deterministic SG can generate emergent shapes by calculation with the shapes of the design objects. For IGBDS, the designers are allowed to interpret and select appropriate non-deterministic SG rules to modify the design objects. Without designer participation in the system, any appropriate non-deterministic SG rule can be applied to modify the design objects at the current state. In this case, the design objects are free to be modified at any time without any predefined order sequences for the execution of the SG rules. What has gone on related to the application of SG before the current state does not need to be accounted for.

2.4.4 Development of Shape Grammars

Eight fundamental issues related to developing new SG as addressed by Cagan (2001) are discussed. The discussion outlines all these key issues by elaborating related research works from different aspects. In this way, the guidelines of formulating a new systematic approach in developing new SG can be established in this thesis. The new SG developed by the proposed systematic approach will have certain characteristics that exceed and improve the standard SG approach.

A) Simple Grammar versus Knowledge Intensive Grammar

Regarding the first item of this issue: simple grammar, using the basic and specific Chinese lattice SG developed by Stiny as an example, the SG only consists of a few rules which can describe all legal specific Chinese lattice designs (Stiny, 1977). A simple non-parametric SG, which will be presented in the next section (section 2.5), is a modification version of a more specific Chinese lattice SG, to support more generic pattern design tasks. The purpose of choosing this example is to illustrate how the proposed systematic approach can be developed to create simple SG as well as to improve the generative capability of the simple SG.

In creating simple SG, the standard approach in constructing the SG can be adopted in the first place. For instance, useful information of SG for Chinese lattice design can be obtained from literature or through field studies on the remaining designs of ancient Chinese lattice. All the relevant information of SG for lattice design includes:

- basic definitions of the construction methodology in creating Chinese lattices,
- the information obtained by reverse engineering several Chinese lattice designs and viewing their form and function or stylistic characteristics, and
- the information obtained by studying the Chinese lattice literatures, understanding what are the factors leading to different types of designs which have been generated and the state-of-art of methodologies and technologies in creating the Chinese lattice designs.

An information network of SG is proposed based on all the relevant information obtained from the above procedures. The information network is designed to assist in the construction of a specific SG. Once the specific SG is developed, the second stage of the proposed systematic approach can take place. The second stage aims to figure out all the possibilities in extending the generative capabilities of the specific SG. This stage is analogous to the current practice of designing in industries, to figure out new methods and resources to explore new designs to fulfil new design requirements. In this case, the specific SG can be extended to generate more generic designs, which can fulfil new design requirements and go beyond the existing designs under current consideration.

It is necessary to analyse how to modify the specific Chinese lattice SG which can fulfil new design requirements. A methodology to redefine the specific SG to a more generic SG is proposed based on the identification of modifiable elements of the specific SG, and the modification, classification and reorganisation of the identified modifiable elements in creating a new SG. When redefining the specific SG, the relative trade-off between various costs (e.g., the effectiveness of the newly derived generic SG in generating the specific designs may be reduced) and benefits (e.g., extending the confined solution space for generating more generic designs) caused by such changes have to be examined and evaluated. The newly derived SG may overcome some of the difficulties in which situations cannot be properly handled by the old specific SG. For instance, the newly derived SG can make previously unfeasible solutions to the design possible, such as exploring the diversity of new designs by maintaining some of the original design characteristics and incrementally adding new elements. As a whole, the capability of the proposed systematic approach is reflected by the very nature of its power in continuously updating the SG to suit new design requirements.

Regarding the second item of this issue: knowledge-intensive grammar, the proposed systematic approach uses classification methods for component definition, specification methods for defining design spaces to gather information about components, constraints, spatial relationships among components, and evaluation criteria. All the relevant information is extracted from the existing designs. After gaining all the relevant design information, the next step is to strategically model such information into a complex information network of SG from different perspectives.

Similar to the creation of simple SG, the proposed information network of SG can then be used by the proposed systematic approach to assist the construction of specific knowledge-intensive SG. For instance, in the digital camera design example, the SG rules reveal (which will be introduced in the chapters 5 and 6) are knowledge-intensive SG. An evolutionary architecture is adopted to evolve the new SG in order to generate more new designs with certain design characteristics, which can fulfil new design requirements and support creativity in design.

As a result, the proposed systematic approach can be used to derive a knowledge intensive SG, for example the digital camera SG, in order to generate the feasible and functional designs. The proposed systematic approach can also be used to derive simple, more primitive grammars, for instance the pattern design SG, which can generate topologically valid but not necessarily or completely feasible or functional solutions.

B) Generation versus Search

Regarding the first item of this issue: generation, from theoretical points of view, the SG rules can generate an infinite complexity of designs. Most of the SG rules described in the literature on design computation and related areas are carefully selected and have specific knowledge embedded in the SG rules. This can help the illustration of the applications of both simple and knowledge-intensive SG in generating feasible and emergent designs. However, without employing useful SG rules to generate designs, the designs generated can only be feasible in some cases or situations but the chances to best meet certain design criteria are relatively small. The SG rules must therefore be evaluated and selected for particular application domains.

If the SG rules are designed to be specific to a problem, then the system is limited to a certain range of designs. If the SG rules are designed to be generic, then the system does not have the specific kind of knowledge for designers to benefit from. Exploring potentially useful rules is a more fundamental issue than exploring emergent designs from recursive application of a set of well established SG rules. The proposed systematic approach helps to convert the specific SG to more generic and useful SG by identifying and modifying the modifiable elements of the specific SG rules. In this way, a balance between how much knowledge has to be kept in the specific SG and the amount of pure exploration that can occur by modification of the specific SG rules can be determined.

Regarding the second item of this issue, i.e., search, a different interpretation of the meaning used by Cagan (2001) is adopted. The meaning of search as described by Cagan (2001) is first stated:

"…In the search mode, the grammar is used to search design space for designs of certain characteristics and performance; in particular, directed search techniques such as shape annealing seek out optimal solutions among the many feasible designs within the language of the grammar."

The proposed systematic approach developed in this research employs the evolutionary algorithms as alternative search techniques to the shape annealing techniques in which new SG vocabularies are added to modify the SG. New SG rules can then be derived to explore new designs rather than searching optimal solutions among the many feasible designs within the language of the SG.

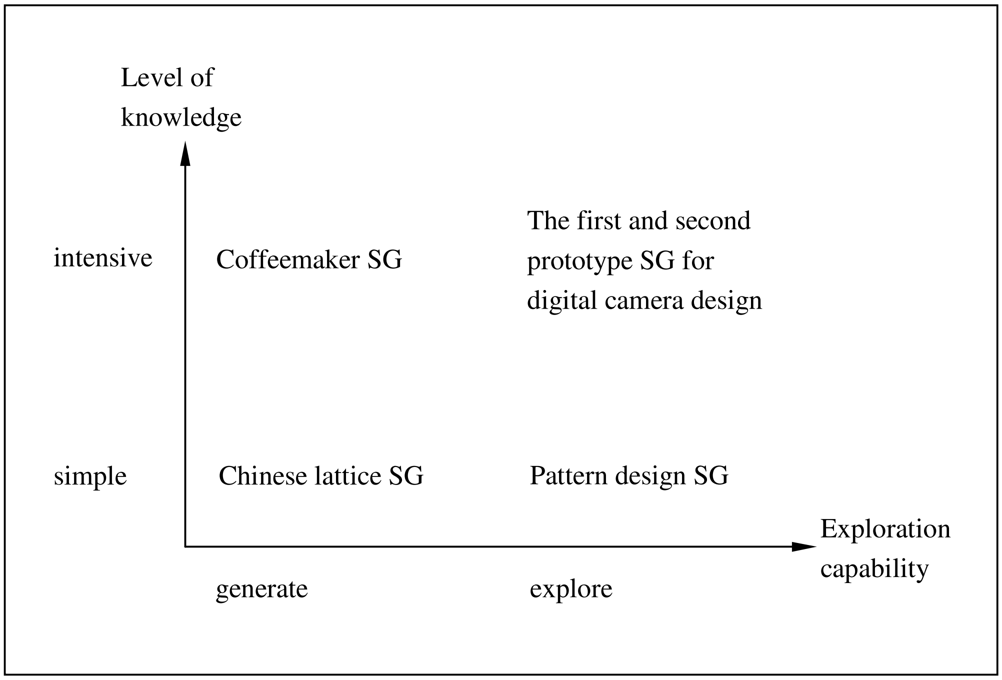


Figure 2.1 Connection between knowledge level and exploration capability level of shape grammars

In this way, the design space is expanded by the application of the newly derived SG rules but at the same time it is optimised. The evolving SG rules are used instead of fixed SG to explore designs of certain characteristics and performance. The concepts of searching from the fixed design space generated by fixed SG are extended to exploring and optimising designs by the evolving SG. As a result, potentially useful SG rules can be generated. Subsequent exploration of those potentially useful SG rules to generate more complex designs can be performed.

The interconnected relationships between these two issues, that of simple versus knowledge intensive and generation versus exploration, are shown in Figure 2.1, which plots the amount of exploration capability versus the level of knowledge. The pattern design SG (introduced in section 2.5), redefining the specific Chinese lattice SG with new modifiable elements and order sequences of rules for more generic pattern designs, appears in the lower right. The Chinese lattice SG developed by Stiny (1977), because of its relative simplicity and being specific only to one particular application, lies in the lower left. The coffeemaker SG developed by Agarwal and Cagan (1998), is defined as a knowledge-intensive SG which can generate complicated designs without optimisation search, appears in the upper left. Two digital camera SG which will be introduced in the coming chapters (Chapter 5 and 6) for integration with an evolutionary framework, are positioned to the upper right. The proposed systematic approach introduces the integration of evolutionary algorithms to the specific SG in order to increase their generative capabilities.

Knowledge of the specific SG tends to improve the efficiency in using the specific SG to design. However, the knowledge intensive SG has pitfalls: restriction of the amount of exploration done and novelty within the designs generated. In developing the evolutionary architectures, control strategies have to be considered to utilise the advantages of knowledge built into the specific SG and the exploration capabilities provided by evolutionary algorithms.

C) Independence versus Coupling of Form and Function

Regarding the first item of this issue: independence, for industrial products like a coffeemaker, some components are functionally independent to the others. The coffeemaker SG can be derived from the functional decomposition of its different components (Agarwal and Cagan, 1998).

Regarding the second item of this issue, i.e., coupling of form and function, some component forms are freely to be modified but some are tightly coupled with functions. For the first case, the SG rules can be grouped from a geometric perspective whereas for the second case from a functional perspective. As a result, the existing designs can be analysed and classified into the groups from different perspectives like geometric and functional perspectives.

D) Symbolic-atomic Grammar versus Emergent Grammar

Regarding the first item of this issue, i.e., symbolic-atomic grammar, the pattern design SG introduced in the next section (section 2.5) is developed using atomic elements for the configuration of the elements of the pattern designs. Although the pattern design SG derived by the proposed systematic approach does not have the pure "SG emergent" properties using maximal representation, the proposed systematic approach have alternative solutions. The alternative solutions are by means of reconstructing a specific Chinese lattice SG from different perspectives and using different construction methods. In this way, the proposed systematic approach can be used to derive potentially useful SG rules for more generic applications.

Without a set of potentially useful SG rules, the ambiguity and lack of structural properties of shapes hinder their power in SG to generate designs. Imagine searching an almost infinite number of designs generated by irrelevant SG rules to get some emergent designs, which are meaningful and of interests to designers. This is time consuming or even fails. One of the key issues leading to the success of applying SG in real world applications is to identify potentially useful and high quality SG rules. The importance of these rules was therefore emphasised and addressed in the previous discussion.

Regarding the second item of this issue, i.e., emergent grammar, the emergence of designs will not occur when only relying on the magic power of SG. The primary condition in making the emergent properties powerful tools of SG is to ensure that potentially useful SG rules are employed within the SG. In the development of the first prototype system for digital camera design, an attempt to derive new and useful SG rules for new shapes is implemented. Although it is not the pure "SG emergent" shapes generated by the newly derived SG rules using the maximal representation, the proposed systematic approach opens up opportunities to derive useful SG rules. Once the potentially useful SG rules are derived, subsequent exploration of those rules to generate more complex or emergent designs can be performed in future research.

E) Fixed Grammar versus Parametric Grammar

Regarding the first item of this issue, i.e., fixed grammar, due to its limitation in flexibly modifying shapes, it can only be used in some applications which require fewer varieties of design generation, such as the pattern design application. The pattern design SG can perform the combinatoric enumeration of symbolic-atomic elements. On the other hand, regarding the second item of this issue: parametric grammar, since the parametric SG has more flexibility in modifying the shapes, they can be used in many different applications. Infinite variations within a class of stylistically consistent product form designs can be concisely represented by the parametric SG. It is that nature of parametric SG that makes them most appealing for applications to intricate in industrial and engineering domains. For example, the SG rules for digital camera design (which will be introduced in the chapters 5 and 6) are parametric SG rules developed for industrial and engineering applications.

F) Type of Algebra Best for Modelling a Given Application

F1) The $V_{02} \times V_{12}$ Coffeemaker SG

For industrial and engineering design applications, V_{02} SG can be used to define discrete functional components at the first instance. It follows with applying V_{12} SG to concisely describe the significant transformation of the shapes of the functional components in the two dimensional planes. For example, the coffeemaker SG has the powerful expressive properties in describing the spatial relations of shapes of the functional components which have many constraints and tightly coupled to other components (Agarwal and Cagan, 1998).

F2) The V_{02} Pattern Design SG

For generic pattern design applications, V_{02} SG use labels to represent elements and manipulate those elements in the two dimensional planes. Labels will be necessary to represent the V_{02} SG which are derived for the combinatoric enumeration of symbolic-atomic elements. More sophisticated pattern design applications can be developed if higher dimensional SG is employed.

F3) The $V_{02} \times V_{12}$ Digital Camera SG

For product design applications, the use of $V_{02} \times V_{12}$ SG is sufficient to describe 2-1/2 dimensional products. For example, the digital camera $V_{02} \times V_{12}$ SG rules, which will be introduced in the coming chapters for the first prototype system, are derived to generate the two dimensional profiles of the components. A complete three dimensional representation of the digital camera can then be produced by different engineering methods such as coiling, extrusion, lofting, revolving and sweeping of the generated two dimensional profiles. In this way, the digital camera SG first generates the two dimensional profiles of each component. The two dimensional profiles of each component are then extruded with different thicknesses to form the preliminary three dimensional representation of the components. The digital camera SG further applies the Boolean operations on the preliminary three dimensional shapes of the components in order to generate more complex geometric features of the components such as the features of the Lens.

F4) The $V_{02} \times V_{12} \times V_{03} \times V_{13}$ Digital Camera SG

Although the $V_{02} \times V_{12}$ SG developed in the first prototype system is capable of generating most of the standard digital camera form designs, it does not support the free form modelling of the products in three dimensions. The $V_{02} \times V_{12}$ SG can only work in planes with simple abstraction into the third dimension by using the engineering methods for example, the extrusion of two dimensional profiles. Since the third dimension is important to many engineering artefacts, the maturity of the two dimensional SG developed in the first prototype system continues to integrate with $V_{03} \times V_{13}$ SG in the second prototype system. Obviously, the development of higher dimensional SG in algebras of label (V_{ij}) and weight (W_{ij}) has significant advantages in handling complex design activities; however, with such SG like the V_{33} SG come extremely complexity in implementation.

G) Evaluation with Shape Grammar Applications

In an attempt to derive the new SG rules in the first prototype system for digital camera designs, the evaluation of the newly derived SG rules are evaluated indirectly by the designers. The generated designs are first evaluated by the designers during the evolutionary process and then the correspondingly newly derived SG rules are evaluated accordingly. In the first prototype system, only one evaluation criterion is used to indicate the overall performance. The overall performance is determined after all the SG rules have been executed.

In the second prototype system for digital camera designs, multi-objective functions are used to evaluate the generated designs. The newly derived SG rules can be evaluated indirectly by artificial selection and objective functions from different perspectives. In addition to the application of multi-objective functions in evaluating the SG rules, control strategies are used to control the modification of shapes as well as evaluate the corresponding SG rules.

H) Routine versus Creative Design

Stiny has commented on the desire to eliminate ambiguous shape properties. Stiny picked a Ph.D. project as an example - “I (Stiny) recently saw the sign “No Ambiguity” in an office for Ph.D. students in artificial intelligence (AI) who were working on “design rationale” – in particular, on a sketch-recognition language complete with a vocabulary of predefined shapes and a syntax for combining them.” (Stiny, 2006, p. 305). With the issue of “No Ambiguity” addressed in this example, Stiny continued with his comments - “It seems that ambiguity is something to stop in AI. But this seldom if ever happens with signs, and can’t be expected for designs. In fact, it misses what drawing and sketching are for. Meaning is closed off in advance to anything new. There’s no reason for reason. My retrospective account of meaning in terms of topologies provides a workable alternative in an open ended process.” (Stiny, 2006, p. 305).

Stiny further picked another example from scholar Gerald Jay Sussman. Stiny quoted what Gerald Jay Sussman said in Sussman’s seminar on March 17, 2005 that “The key idea is the development of engineering “languages” that allow us to separate concerns in design. Such languages provide ways of expressing modularity and isolation between modules. They provide means of composition that allow the construction of compound systems from independently-specified and implemented parts. They allow characterization of both structure and function, and how function is determined by and implemented in terms of structure. They provide black-box abstractions that allow one to specify the behaviour of a composition independently of the implementation.” (Stiny, 2006, p. 401). Based on this example, Stiny continued to make his comments on the issue of “No Ambiguity” more clearly - “Computer scientists like to divide things into independent units to make problems combinatorial. Language is vocabulary and syntax, and an engineering language uses both to fix structure and function. Compositionality precedes design – what a compound system does depends on its constituent modules (parts), what they do separately, and how they’re put together. The shape grammarist agrees that this is a dangerous idea – before you know it, design is impossible.” (Stiny, 2006, p. 401).

Compared with the proposed systematic approach developed in this research, a first glance seems to systematically organise all the predefined information and a hierarchical structure or syntax for combining the information. The first step in the proposed systematic approach is similar to the development of engineering “languages” as described above. However, there are two main strategies in making the proposed systematic approach to link the power of SG harmoniously together.

1. The first strategy of the proposed systematic approach is the same as what standard SG approach does, to define a set of specific SG rules for designs. But the differences are to systematically organise all the relevant information of SG, provide ways of expressing modularity of SG rules and isolate modules of SG rules for different design requirements.

2. The second strategy of the proposed systematic approach is to allow room for the change of abstraction levels of the specifications of the modules of SG rules. In this case, potential use of maximal representation of SG can be allowed for the generation of "SG" emergent shapes. The emergent shapes are generated with ambiguity and emergent properties of SG using maximal representation. The research to apply maximal representation in product design domain requires further research and is not included in the scopes of this thesis.

2.5 Redefining Shape Grammars

This section describes the methodology to convert a specific SG to a more generic SG so as to generate more generic designs. The system development and implementation results of developing non-parametric 2D SG for generic pattern design are presented. An example of Chinese lattice design is chosen for the illustration of a small part of the proposed systematic approach. The experience gained from this experimental case study is helpful in developing the completed proposed systematic approach for the construction of the two prototype systems. The foci of this experimental case study are on the issues of identifying elements of a specific SG from perspectives other than the original one and modifying the identified elements to derive a new SG for more generic designs. The key features of this demonstration include the relevant technological details of the implementation of the IGBDS.

2.5.1 Shape Grammars for Chinese Lattice

The Chinese lattice design shown in figure 2.2 is first introduced as an example to illustrate the application of SG in generating pattern designs. As documented by Stiny (Stiny, 1977), the SG specified in figure 2.3 and 2.4 generate the lattice shown in figure 2.2. Since the SG is only specific for one type of Chinese lattice design, it limits the generative capability of SG. A new approach is adopted to define a new SG based on the symmetric property. The SG developed by Stiny is used as a reference for the development of the new SG for more generic pattern designs.

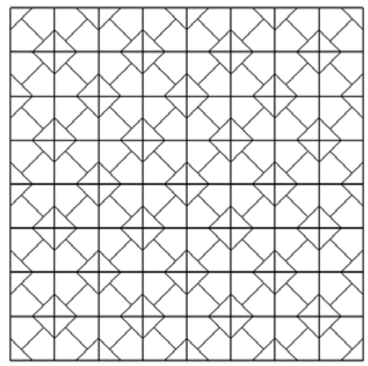


Figure 2.2 Chinese lattice design: Chengtu, Szechwan, 1825 AD. (A modified version of the original figure shown in (Stiny, 1977) is used here.)

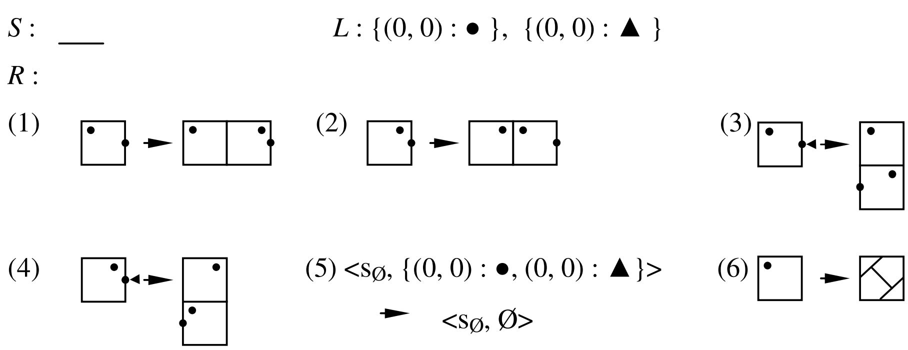


Figure 2.3 Shape grammars for the generation of lattice design (Stiny, 1977)

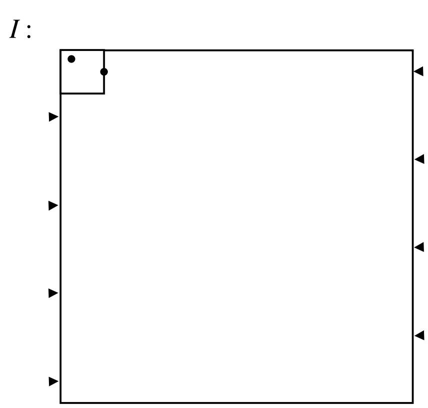


Figure 2.4 Initial shape for the generation of lattice design (Stiny, 1977)

As shown in figure 2.3, *S* refers to a finite set of shapes, in this case it only contains a single straight line. *L* refers to a finite set of unordered sets of labelled points. *R* refers to a finite set of shape rules. As shown in figure 2.4, *I* refers to the initial shape. The SG starts at the initial shape *I* and recursively applies the shape rules in the set *R* to generate a shape. The termination of the shape generation process occurs when no shape rule in the set *R* can be applied.

2.5.2 Redefining the Shape Grammars

As mentioned in the previous section, the Chinese lattice SG is limited for the generation of one specific type of Chinese lattice design. In order to enhance the generative capability of the SG, a flexibility study on the modification of a specific type of SG to a generic type of SG is performed. The study involves the analysis of design characteristics of the design such as its symmetric property. Figure 2.5 illustrates the symmetric property of Chinese lattice design.

Each element in the Chinese lattice design is denoted as label ‘A’ and ‘B’. Each element with label ‘A’ has a neighbouring element ‘B’ and vice versa. When considering two consecutive elements together, the same result is applied for another two neighbouring elements and so for more elements. This makes up the implication of symmetric property of the Chinese lattice design. Therefore, a new set of rules for generic pattern designs can be constructed based on the symmetric property.

A	B	A	B	A	B	A	B
B	A	B	A	B	A	B	A
A	B	A	B	A	B	A	B
B	A	B	A	B	A	B	A
A	B	A	B	A	B	A	B
B	A	B	A	B	A	B	A
A	B	A	B	A	B	A	B
B	A	B	A	B	A	B	A

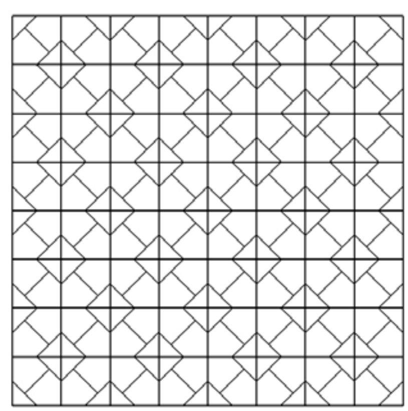


Figure 2.5 The symmetric properties of Chinese lattice design

Using the new set of rules as a tool for investigation in both classical and new patterns gives the designers an advantage to explore different patterns easily and allows them to implement a new sequence of rules for specific designs. In order to define a new set of SG rules to be more generic, both the vocabulary of shapes, rule sequences and rules themselves have to be opened up for the designers to redefine interactively. These redefined shapes and rules allow the designers to accommodate their preferences for the creation of new patterns.

2.5.3 Identification of Modifiable Elements

The reconstruction process modifies three major modifiable components of the specific SG that influence the final result of the pattern designs. These components are the vocabulary of shapes, spatial relations and rule sequences. A reference of the effects of modification is documented in Knight's papers (Knight, 1999a, b).

A) The First SG Component: Vocabulary of Shapes

In the original SG, there is an initial shape as shown in figure 2.4. In the newly derived grammar, a label 'A' is added to the initial shape. The first rule specifies that whenever there is a match of labelled shape with label 'A', four distinct shapes with labels 'A, B, C, D' will be replaced as shown in figure 2.6. The first rule (Rule number 10) aims to introduce more distinct shapes for pattern designs.

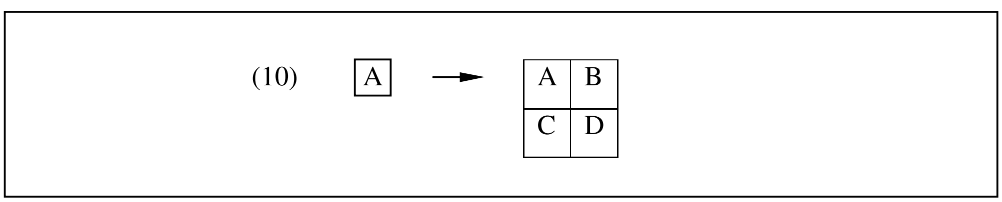


Figure 2.6 The first rule of newly derived shape grammar

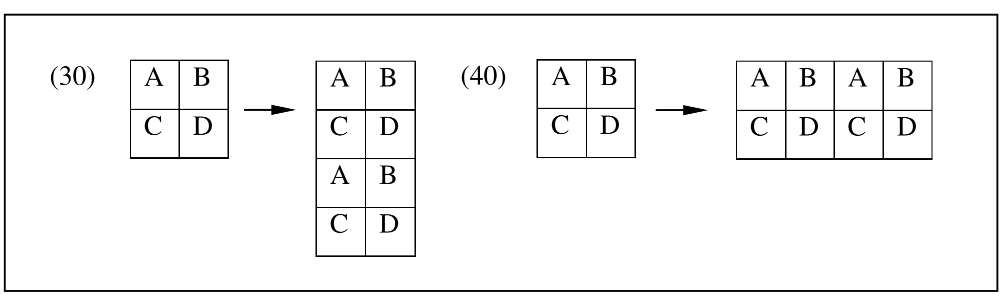


Figure 2.7 The symmetric rules of newly derived shape grammar

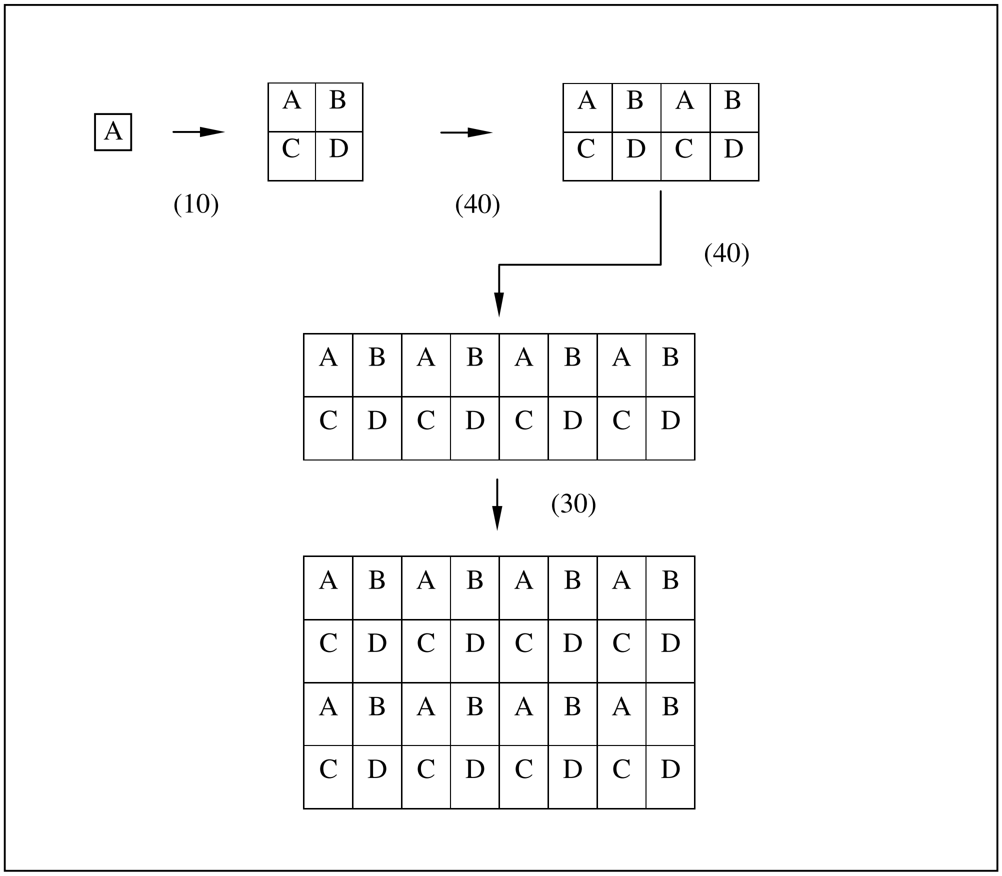


Figure 2.8 The pattern generated with the rule sequence: 10, 40, 40, 30

B) The Second SG Component: Spatial Relations

The spatial relations of symmetric property are considered in the formulation of newly derived rules. Figure 2.7 shows the symmetric rules of newly derived SG. When applying the Symmetric rules (Rule number 30 and 40) to the shapes, the number of elements generated can double in each generation. Compared with the original rules which sequentially apply to each element, only one new element can be generated in each generation. The Symmetric rules can therefore generate elements with the growth of 2^n for n generations. The aim of deriving the symmetric rules is to generate alternative pattern designs.

C) The Third SG Component: Rule Sequences

The aim of modifying the rule execution sequences is to generate alternative configurations of pattern designs. Figure 2.8 shows an example of the pattern generated with the rule sequence: 10, 40, 40, 30.

2.5.4 System Development

This experimental case study for generic pattern design is developed using the Visual Basic Application (VBA) supplied within the AutoDesk Inventor environment. Since Inventor allows software developer to use the "Application Programming Interface" (API) to customise the applications, the system can be potentially used as a plug-in product to Inventor. Also, the programming environment can allow different languages to be used such as Java, C++ and VB through an Automation interface. Figure 2.9 illustrates different approaches in accessing the Inventor API.

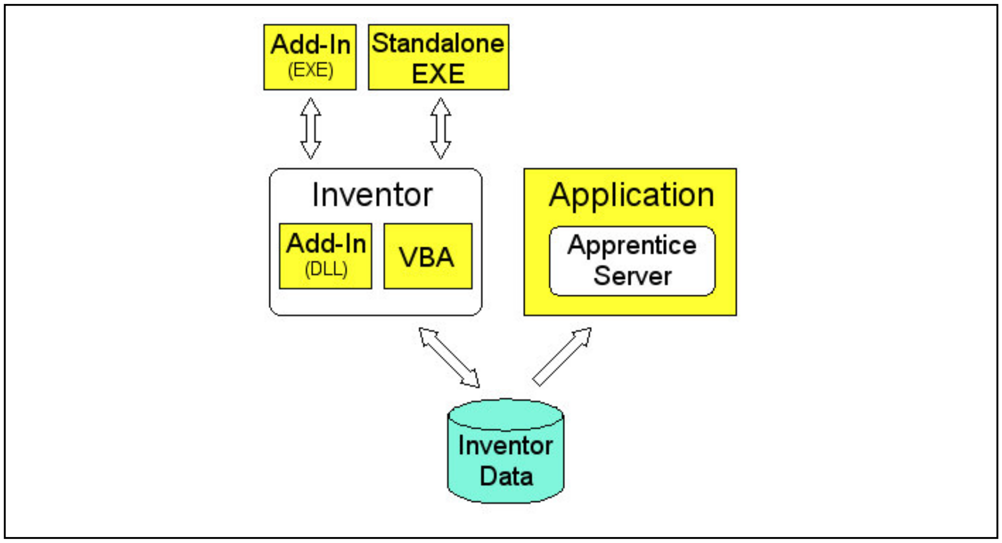


Figure 2.9 Different ways to access the Inventor API. (Extracted from the help menu of AutoDesk Inventor)

2.5.5 System Architecture

The input interface is composed of two parts: Designer preference and Evolution (Figure 2.10). The first part: Designer Preference lists the available shapes for the designers to select. Each labelled shape, Shape A to Shape D, has eight shapes to be selected by designers as shown in figure 2.11. The rules are listed item by item when the designers click the "Next" button in the "List of Rules" section.

For the second part: Evolution, the system can only simulate the random generation of patterns. Since the focus in this experimental case study is on the analysis of the modifiable elements of SG, the evolutionary techniques can be integrated with the IGBDS at a later stage. Finally, a help menu is provided to support the designers for the manipulation of the system (Figure 2.12).

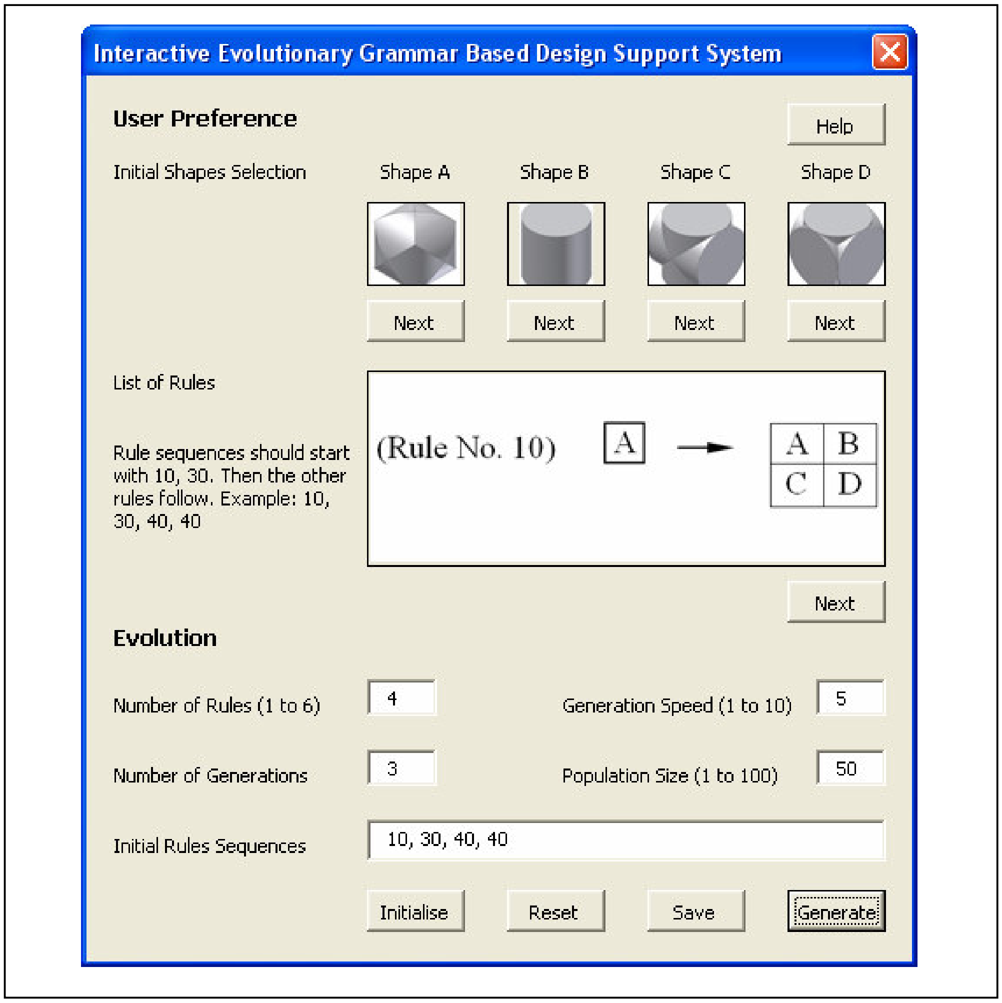


Figure 2.10 The user interface for experimental case study of IGBDS

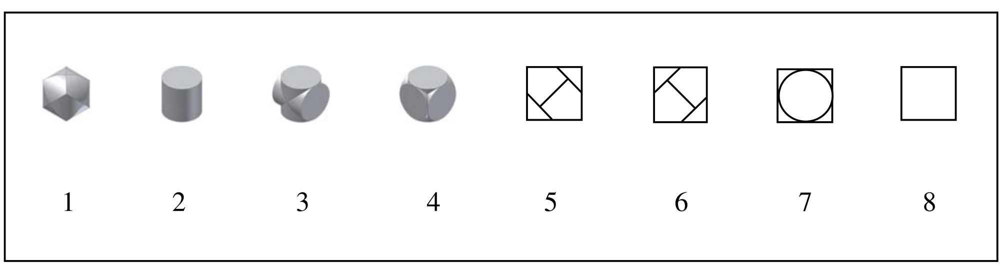


Figure 2.11 Each labelled shape, Shape A to Shape D, has eight shapes to be selected by designers

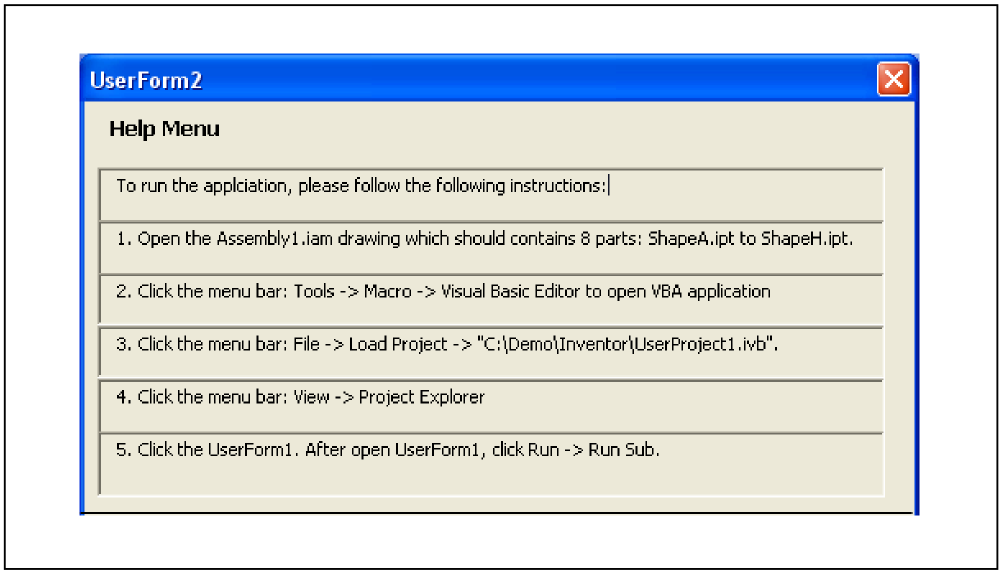


Figure 2.12 The help menu for experimental case study of IGBDS

2.5.6 Implementation Results

The original Chinese lattice design shown in figure 2.2 is generated by the system with the rule sequence: 10, 30, 30, 40, 40 and Shape A to Shape D with shape number: 5, 6, 5, 6 (Figure 2.13).

With the IGBDS, the original SG developed for the Chinese lattice design has been modified to generate more generic pattern designs. Figure 2.14 shows another pattern with the rule sequence: 10, 30, 40, 40, 30 and Shape A to Shape D with shape number: 5, 6, 7, 8. Figure 2.15 shows a pattern of 3D objects with the rule sequence: 10, 30, 40, 40, 30 and Shape A to Shape D with shape number: 1, 2, 3, 4.

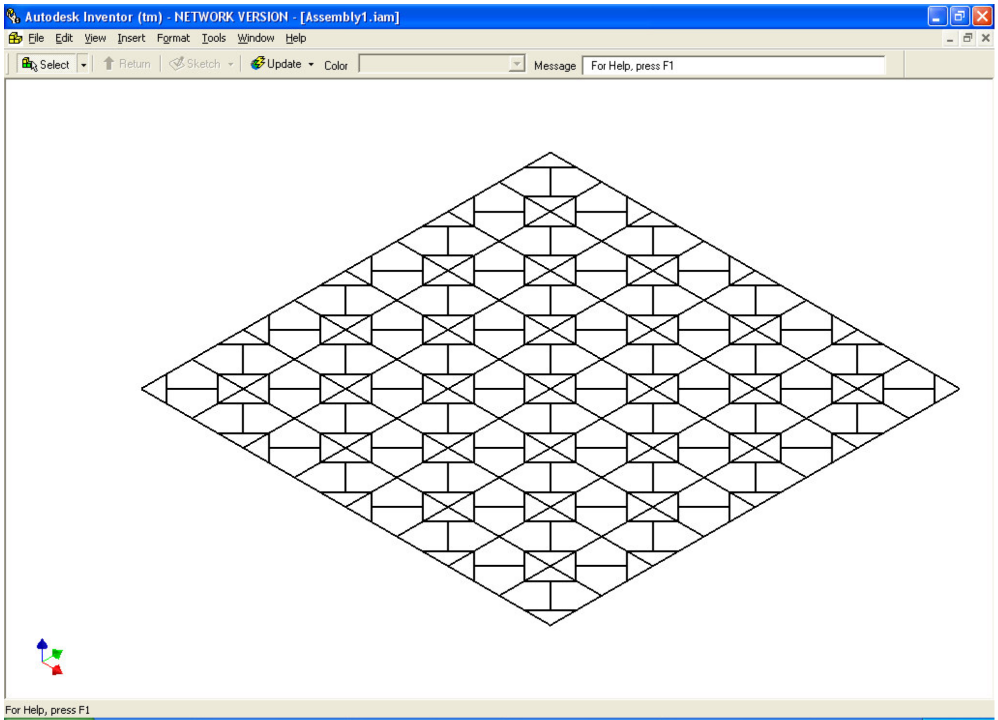


Figure 2.13 The Chinese lattice design shown in figure 2.2 is generated by the system with the rule sequence: 10, 30, 30, 40, 40 and Shape A to Shape D with shape number: 5, 6, 5, 6

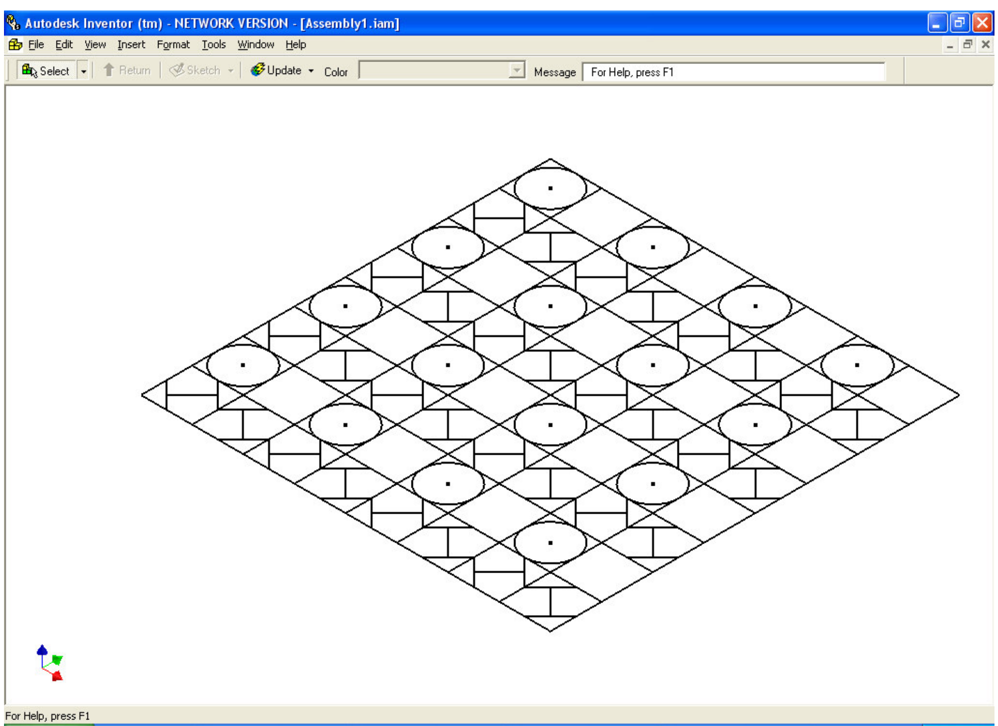


Figure 2.14 Pattern generated with the rule sequence: 10, 30, 40, 40, 30 and Shape A to Shape D with shape number: 5, 6, 7, 8

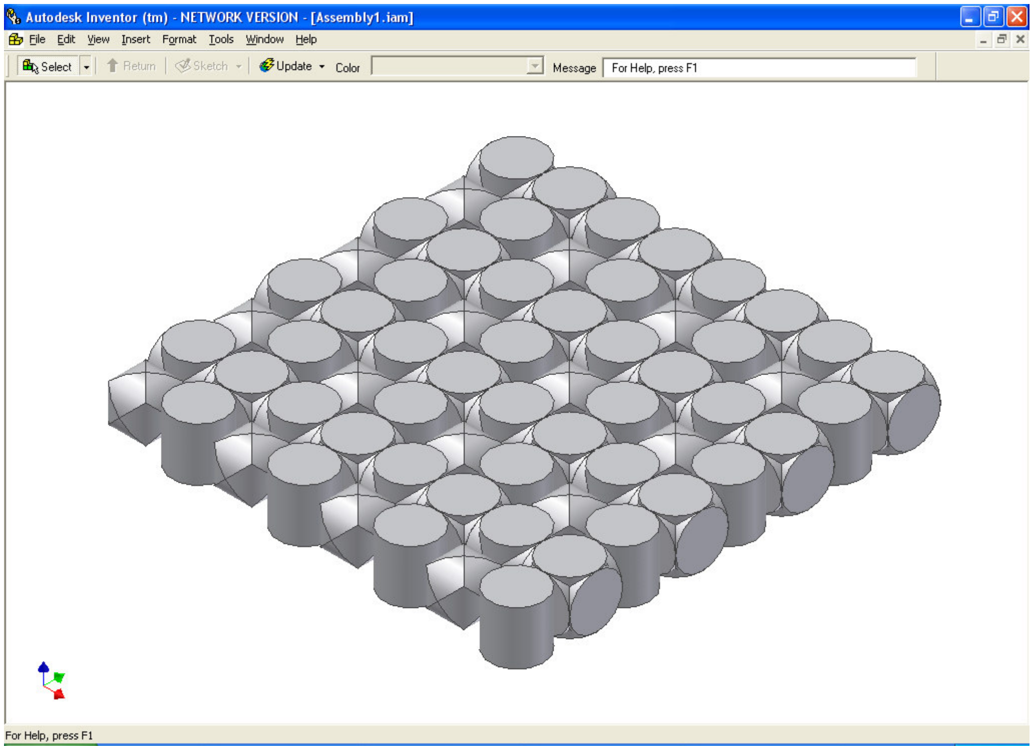


Figure 2.15 Pattern of 3D objects with the rule sequence: 10, 30, 40, 40, 30 and Shape A to Shape D with shape number: 1, 2, 3, 4

2.5.7 Enhancing Generative Capability

This section has described the system development of an IGBDS in a case study. Especially, the key features including the relevant technological details of the implementation for the system have been described. The operation process of running the system has been demonstrated. The experience gained from this experimental case study is helpful in developing the completed proposed systematic approach for the construction of the two prototype systems. This case study demonstrates the methodology in converting a specific SG to a more generic SG that can generate more generic designs. This is achieved by identifying the modifiable elements of a specific SG and modifying the modifiable elements to form new SG rules. An example of converting the Chinese lattice SG to a more generic pattern designs was chosen for demonstration of the proposed systematic approach.

Alternative approaches could be applied for more dynamic interpretations of Chinese Lattice. For example, Soddu et al. have applied a lot of possible different dynamic codes to better represent the character and the style (Soddu, 1994; Soddu and Colabella, 1997). This type of approach is described with more details in section 3.5.3 and 3.6.2.

2.6 Application of Shape Grammars to Design

First introduced in Stiny's seminal work and demonstrated in architectural design domain (Stiny, 1980a, b), SG rules captured styles of designs, generated stylistically consistent designs and novel designs. Since then, many SG based systems have been implemented in architectural design, visual arts, engineering and industrial design domains. All these systems involve: 1) encoding the knowledge of design process into SG by analysing existing sets of designs so as to reproduce these designs and 2) exploration of new designs from the stylistically consistent languages.

For example, Li developed a parametric SG for the enhancement of understanding styles in architecture (Li, 2004). The SG rules were developed in accordance to the analysis of a twelfth-century Chinese building manual: the Yingzao fashi. To understand a style of a particular architecture, the designers interpreted which designs generated by the SG were stylistically correct. A standard of stylistic correctness could be developed by refining the SG to eliminate the generation of stylistically incorrect designs and produce only stylistically correct designs. Other examples related to engineering and product design include those introduced in section 1.2.

2.6.1 Difficulties in Implementation

Apart from the discussions of the powerful capability and advantages of SG in purely theoretical terms, there are many difficulties in creating implementations of SG which have been highlighted by Knight (1999a, b) and by Gips (1999). There are researchers addressing computer implementation issues of SG, such as Ulrich Flemming, who suggested that it was necessary to construct a robust implementation of a parameterised SG interpreter that allows for the graphical definition of parameterised SG rules. Practically, Flemming's students found that difficulties appeared when attempting to create such an implementation in determining parametric rule application (Chien et al., 1998).

2.6.2 Optimising Shape Grammars

One of the key objectives in this thesis is to enhance the generative capability of IGBDS by integration with an evolutionary framework to an IGBDS. There are other related research works which have been developed using different approaches with other optimisation techniques. In order to understand the historical development and the current state of art in this area, this section reviews such applications in different domains. The critical issues of implementing SG in each application are highlighted.

A) Architectural Design Domain

Results are limited to a fixed grammar with the research moving in the direction of learning process from both parsing and generating expressions. One possible way is to establish a hypothesis based on a theory of shape-evolution. An attempt was made to provide possible architectural grammars with conventions for configurative mechanisms, to parse historical shape-evolution. The shape-evolution can be an architectural or a vernacular building style. New rules and rule-sets can be evolved in an environment of already given rule-sets and procedures of evaluation. The process of the evolution was studied as the simulation of knowledge acquisition. Gero (1992) and Gero et al. (1994) presented some results in this direction.

Attempts have been made by Rosenman et al., in order to extend the generative capability of SG by integrating the evolutionary algorithms with grammar based design system (Rosenman, 1996a, 2000; Rosenman and Gero, 1999). Two-dimensional orthogonal plan SG rules for buildings have been derived and encoded as genetic representation. The genetic representation includes genotype which encodes the selection and application of a set of SG growth rules. The evolutionary algorithms evolve the genetic representation of the SG rules which make small modifications to an existing plan in order to generate new plans.

With an example of designing a facade, Gero and Ding (1997) have applied the integrated SG and evolutionary algorithm approach to explore style emergence in architectural designs. The work was subsequently refined to show that the style can be captured from a language model using genetic representation (Ding and Gero, 2001). Using such approaches would seem to create rules through a learning mechanism such as a GA. However, there were no mechanisms to modify the existing form elements. Therefore, the rules were only altered to generate designs within a confined design space. This research targets this point to extend the design space by allowing both the form elements and configuration to be modified in a controlled manner.

Another application in architectural design domain has been conducted by Çağdaş (1996) who employed a SG model to integrate a depth-first search method and a SG for the design of row-houses. The generative capability of grammar and the reasoning capabilities of knowledge-based systems are utilised in guiding the generation of design solutions from a high-level abstraction to a low-level abstraction.

While CAD systems are becoming more and more sophisticated, the direction for research is not limited only to technical aspects but also to other issues such as cultural and environmental concerns. Sourav and Michael (1996) presented the Network and GA models to compute shape grammars. This approach led to the exploitation of aspects of knowledge representation and directed search within the Network. They demonstrated that the use of an Augmented Transition Network (ATN)-frame was able to simulate the idea of space-between. Real instances of non-bisymmetric Palladio villas via a shape grammar were generated in the background of cultural expressions.

B) Urban Planning Design Domain

In urban planning design domain, Duarte et al. (2006) developed a parametric SG for urban planning. The SG captured the knowledge of creating some features of the existing urban fabric. A large amount of work was put into historical analysis and fieldwork for the derivation of useful SG rules. This project is going to integrate genetic algorithms for the generation of novel urban and housing configurations that are more sustainable and energy efficient.

C) Structural Engineering Design Domain

In structural engineering design domain, Shea and Cagan (1999) used shape annealing, a combination of SG formalism and simulated annealing, to design structures. The concept of search process in simulated annealing was borrowed from physical processes whereas evolutionary algorithms emerge from biology. More details of this approach can be referenced to the Shea's research works (Shea, 1997, 2001, 2002, 2004).

In this generative structural design system called eiForm, a set of SG growth rules are controlled by the simulated annealing technique to add, replace and modify structural members in order to generate new structures. The special design characteristic of structures is the planar topology of structural members. Focusing on special design characteristics of structures, the SG rules are developed to generate space frame structures with different topologies of structural members. The SG rules generate the 3D structures indirectly in which the 2D structures are essentially generated and then projected onto a 3D surface, such as a hemisphere or pyramid. In this system, an initial form is defined to represent the initial design for a frame structure. A set of transition rules are also defined to perform the substitution operations which add, remove and modify structural members. The SG rules are developed using the most commonly used approach by analysing existing classes of design.

Particular examples like traditional geodesic patterns have been constructed using the SG. The key operations of the SG can be classified into two parts: 1) iteratively apply the SG rules and 2) continuously optimise the generating designs. For the first part, different space-frame structures can be generated through iteratively applying the SG rules. For the second part, the SG was combined with constraint satisfaction mechanisms, performance evaluation software and a simulated annealing optimisation algorithm to continuously optimise the generated designs.

Apart from this particular example, a wide range of other examples of space frame structures have been constructed using this system. To illustrate the generative capability of SG, Shea (1997) has demonstrated that the system was capable in generating three space frame roof structures for an octagonal air plane hanger with walls that vary in height. Each design was evaluated against different evaluation criteria with the first design targeted to optimise for pure efficiency, the second and the third to achieve good visual appeal through an aesthetic measure based on visual uniformity and the golden proportion respectively.

Another example using this system was in generating truss structures (Shea and Cagan, 1999), the topologies were classified as one group whereas the independent members were classified as another. Two groups were influenced by each other to complete a whole design. Thus the topology of the truss affected the sections of independent members or vice versa.

D) Mechanical Engineering Design Domain

In mechanical engineering design domain, earlier attempts in merging grammars with optimisation techniques have been achieved by Schmidt and Cagan (1998), aiming to direct grammatical generation by design goals. These attempts have led to success in the generation of optimal mechanical systems. Other examples include the generation of optimsed process plans for machining designs done by Brown and Cagan (1997). The process plans were defined by a language of machinable parts which were derived by Brown et al. (1995).

E) Product Design Domain

As described in section 1.2, Ang et al. (2006) have applied an evolutionary algorithm to evolve a set of 2D SG rules for the generation of Coca-Cola bottles.

2.7 Summary

This chapter has described the working principles, computational issues, basic technical mechanisms, applications and all the related properties and issues of SG. The key points are as follows:

- The working principles of SG have been introduced with the discussion of the formulation, representation and manipulation of SG. The representation of SG has been described and discussed. A SG system consists of vocabularies which are made and represented by shapes. The shapes are formed with basic elements of dimensions zero, one, two and three which are points, lines, planes and solids respectively. The basic elements of shapes can be represented by maximal representation. The use of maximal representation of shapes is a fundamental requirement which leads to the creation of a special property of SG: ambiguity of shapes. The representation of SG rules not only relies on the vocabularies of SG, but also on the spatial relations of shapes specified in the left and right sides of the SG rules.

- The manipulation of SG has been described. The SG rules are applied to modify the shapes of the design objects. The operations of the SG rules have been described which include the recognition and replacement of shapes of the design objects. The schemas used to derive the SG rules have been discussed.

- A thorough description and discussion on the basic technical mechanisms has been performed to illustrate how a SG system works in design. The basic technical mechanisms consist of recursion and embedding working devices. Recursion mechanism of SG and the details of the order of executions of SG rules have been discussed. The embedding working device of SG which provides capability in both interpreting and changing the shapes has been discussed.

- The computational issues in the development of SG have been discussed. SG properties such as non-deterministic, ambiguity, embedding and emergent properties have been identified and discussed. The generative capability and classification of SG have been discussed. The critical computational issues related to the development of SG have also been discussed. The ambiguity of shapes has also been highlighted in the discussion with the computational issues of SG.

- The methodology in converting a specific SG to a more generic SG for the generation of generic designs has been discussed. The system development and implementation results of developing non-parametric 2D SG for generic pattern design have been presented. This is achieved by identifying the modifiable elements of a specific SG and modifying the modifiable elements to form new SG rules. An example of Chinese lattice design has been chosen for the illustration of a small part of the proposed systematic approach. The experience gained from this experimental case study is helpful in developing the completed proposed systematic approach for the construction of the two prototype systems. The foci for this experimental case study are on the issues of identifying elements of a specific SG from perspectives other than the original one and modifying the identified elements to derive a new SG for more generic designs. The key features of this demonstration include the relevant technological details of the implementation of the IGBDS.

- The historical development and the current state-of-art for the development of many applications in different application domains using SG have been reviewed. The difficulties in the implementation of SG have also been discussed. One of the key objectives in this thesis is to enhance the generative capability of SG by integration with an evolutionary framework. Other related research works which have been developed using different approaches and applied in various design domains such as A) Architectural Design, B) Urban Planning Design, C) Structural Engineering Design, D) Mechanical Engineering Design, and E) Product Design, have been reviewed. The critical issues of implementing SG in each application have been highlighted.

In summary, shape grammars are useful representation and reasoning methods for generating designs. Current research in this area involves a wide range of issues in geometric reasoning, knowledge representation, and information organization. Various applications have demonstrated that shape grammar remains a fundamental research area with great potential to support design applications. However, none of the existing systems integrated evolutionary computing techniques for evolving SG rules that include 3D representations, with integration to 3D solid modelling techniques. In the coming chapters, generative and evolutionary computation in design will be reviewed before the development of a new computational framework that integrates shape grammars with evolutionary computing techniques is introduced and evaluated.

Chapter 3

Evolutionary Algorithms

3.1 Introduction

This chapter reviews the working principles, computational issues, basic technical mechanisms, applications and all the related properties and issues of evolutionary algorithms. This chapter consists of five main sections:

- In section 3.2, researches on different types of generative and evolutionary design techniques are described.

- In section 3.3, the working principles of evolutionary algorithms are first introduced with the discussion of the background of evolutionary algorithms.

- In section 3.4, two system applications have been reviewed to analyse the related researches on the application of integrated SG and evolutionary algorithm approaches which are the key subject to be investigated in this research.

- In section 3.5, the computational issues in the development of evolutionary algorithms are discussed.

- In section 3.6, the applications of evolutionary algorithms in different design application domains are described and discussed in detail.

3.2 Overview of Evolutionary Algorithms

Generative and evolutionary design techniques have become active research topics in recent years. Central in the generative and evolutionary design techniques is the study of generative methods and testing of the designs generated by these methods (Simon, 1969). These techniques are developed based on inspiration from natural evolution. Four main types of evolutionary algorithms were developed: evolutionary strategies (Rechenberg, 1973), evolutionary programming (Fogel, 1963), genetic algorithms (GA) (Holland, 1975) and genetic programming (Koza, 1992) of which GA were widely used.

In this review, the focus is on the application of genetic algorithm. There are four main steps to apply a classical GA to solve design problems. First, representations of genotypes and phenotypes for the specific design problem are defined. Second, suitable GA for the manipulation of the representations is designed. Third, selection criteria for the evaluation of design objects are formulated. Fourth, factors of the environment and designer interaction that affect the performance of the design process when applying these techniques to the design domain are considered.

Since the classical GA can only solve simple design problems, variations of the classical GA have been developed to solve difficult design problems. All related topics have been deeply studied by researchers so as to define primitives, rules, constraints, evaluation criteria and environments more appropriately for the best use of these modified GA.

Modified GA requires few assumptions about the design domain. With this advantage, these modified GA can easily be integrated within a knowledge based CAD system to support optimisation tasks and exploration of abstract concepts. The process of using a traditional CAD system in detailing a single artefact is then enhanced to explore alternative designs. Consequently, these powerful GA will be more widely used as the core computation architecture of future design systems and environments that support design activities.

3.3 Working Principles of Evolutionary Algorithms

3.3.1 Background of Evolutionary Algorithms

A) Adaptation from Nature

The notion of the evolutionary algorithm is an adaptation of natural evolution. Three main ingredients are used to describe the natural evolutionary process: selection, transmission and variation (Universal Darwinism) (Dawkins, 1983). During evolution, the interplay between selection, transmission and variation can be observed through the embryogenic process which describes the development of the phenotype from the genotype. First, those organisms whose phenotype is fit to the current environment will be chosen to reproduce by selection. Second, transmission starts copying of the genetic material from two parents which are the selected organisms. Third, the genetic material will then be combined and transferred to the offspring. During this copying process, errors may be generated if variation occurs. Therefore, the evolutionary process can construct more fit individuals by replacing the genes of a whole species that are not suited to the current environment with more suitable genes.

B) Evolutionary Computation

The evolutionary computation algorithm simulates this process by selecting more successful designs from a population of alternative designs for reproduction. Due to historical reasons, four main types of evolutionary algorithms were developed as described at the beginning of section 3.2. Further appraisal of literature concerning the working principle of the most commonly used type of evolutionary algorithms is described in the coming sections. The application of GA in handling design optimisation and design exploration tasks are also described in the coming sections.

C) Terminologies

Some of the terminologies are frequently used in a GA to formulate a design problem. Genes refer to the coded parameters stored in the population. The values taken by genes are termed alleles. A string is used to represent a collection of genes in one individual of the population. This string is held internally by a GA and referred to as chromosome. The term genotype refers to a collection of the entire coded parameter set of an individual. The term phenotype refers to the solution in the new population that the genes define. GA manipulates the alleles of genes which are then decoded to give parameter values of the phenotypes.

D) Difficulties in Implementation

The main difficulties of applying the generative and evolutionary design techniques are mainly classified into five issues: justifying the right use of genotype and phenotype representation for the specific design problems, designing competent genetic algorithms for the manipulation of the representation (Goldberg, 2002), formulating the selection criteria for the evaluation of both the quantifiable and unquantifiable design objects, considering the factors when establishing the outer and inner environments, analysing designer and computation roles for design of designer interaction environment. However, the success of applying such computation techniques is to analyse each part of the difficulties for specific design problems in detail instead of blindly applying any of the existing techniques to solve the problems. Without a measure of the complexity of the specific design tasks and a thorough understanding of the critical issues when applying the generative and evolutionary design techniques to solve the design problems, the failure rate is higher.

3.3.2 Representation of Evolutionary Algorithms

A) Phenotypes

The phenotype representation describes all permissible designs that can be generated by an evolutionary design system. It enumerates the design-space for evolutionary search by a GA. There are different aspects of the representation issues that need to be addressed which depend on the usage of the evolutionary design system.

B) Genotypes

The genotype representation defines the arrangement of chromosomes manipulated by a GA. A genotype can consist of different arrangements such as a single chromosome, a number of chromosomes, or even a number of pairs of chromosomes (Paton, 1994). Similarly, chromosomes can be constructed with different settings of alleles such as lists of rigidly ordered alleles, unordered sets, or hierarchically structured groups of alleles (Goldberg, 1991).

The role of genotypes is simply to provide the search space for a particular design problem. The search space is composed of alleles of genes which are the coded parameter values of the phenotypes. The genotypes are created and their alleles are modified by generic operators such as mutation and crossover.

3.3.3 Manipulation of Evolutionary Algorithms

The genetic algorithm plays the major role in the evolutionary design system. It performs three main functions: 1) Modifying alleles within chromosomes using genetic operators, 2) Decoding genotypes to produce phenotypes, and 3) Evaluating the phenotypes to identify the fittest designs.

A) Operation of GA

At the beginning of running the evolutionary design system, a simple GA generates an initial population of solutions with random values. A main loop then begins at this stage. Each solution is then evaluated and assigned a fitness value by a fitness function. Based on the score obtained from each solution, the solution with a higher score will be selectively copied to a temporary area termed ‘mating pool’. Two of the solutions are randomly selected as parents from this ‘mating pool’. These two parents generate two offspring by random crossover and mutation operators. These two offspring replace the parents of the population. The crossover and mutation processes repeat to generate offspring until every parent of the old population is replaced, a new population with fitter solutions is established (Holland, 1975).

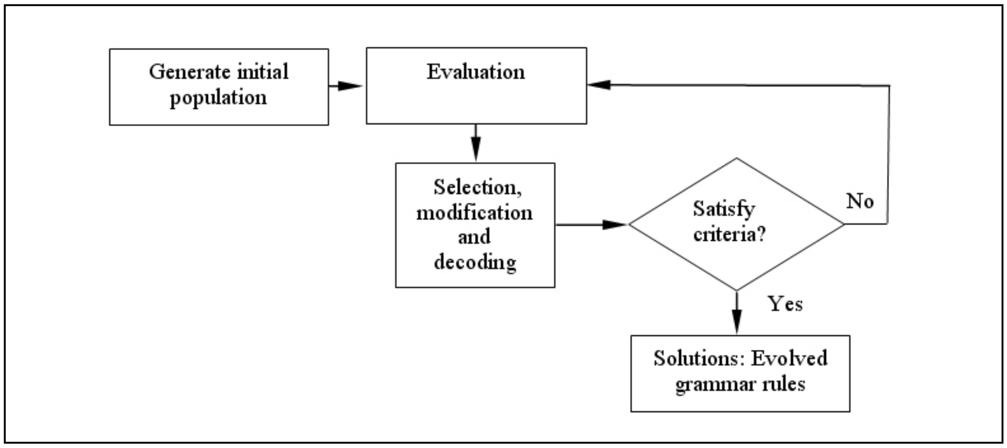


Figure 3.1 Operation of GA

The GA repeats the evaluation and reproduction processes for a specified number of generations, or the GA will stop if a satisfactory solution emerges. Figure 3.1 shows the operation of a simple GA.

B) Selection

The probability of an individual being selected is determined by a selection scheme. The selected individuals are used for producing offspring by genetic operators. Fitter individuals have higher probabilities of being selected. Three types of selection schemes are commonly used: roulette wheel selection, rank-based selection and tournament selection (Yao, 1999).

The algorithm of roulette wheel selection is illustrated. The parents are selected from population P(i) based on their fitness. Let the fitness values of n individuals be f1, f2,…, fn . Then the selection probability for individual i is

$$p_i = \frac{f_i}{\sum_{j=1}^{n} f_j}.$$

The selection probability is calculated directly from an individual's fitness values. If there are less fit individuals and more unfit individuals in the population, these fit individuals will dominate the whole population. This problem can be solved by using fitness scaling methods (Goldberg, 1989).

C) Evaluation

During evolution, designs are evaluated by the objective functions. These objective functions analyse and calculate the fitness values for each phenotype. All objective functions must be adequately and correctly specified as they guide the evolution of designs. Deficiencies of the evolved designs are found if the objective functions are incorrectly specified.

3.3.4 Effects of the Environment

This section introduces the characteristics of the environment that affect the evolutionary design system. The theoretic approach of a generative process which is a simulation of a meta-environment is described (Janssen et al., 2002). The role of the environment is to describe a pre-existing universe for the operation of an evolutionary system. The environment specifies the conditions that affect the design to be grown from an encoded code script to a decoded design (Janssen et al., 2002). In Janssen's approach, the universe consists of two types of environments, the meta-environment and the general environment. The meta-environment is non-evolvable and can be classified as a subset of the environment. It can be defined by the laws and meta-representations which are analogues to physical laws and molecular chemistry in the natural world respectively. The meta-environment remains unaffected throughout the evolutionary process.

The main difference in Janssen's approach to building an evolutionary design system for an optimisation system is that "complex representational schemas" are used to incorporate with the generative process that takes an encoded code script and expands this script to a decoded design. The aim of Janssen's approach is to explore the possibilities to open up the non-evolvable meta-environments into evolvable meta-environments. This approach allows a much wider variety of designs to be evolved compared with the optimisation system in which only a few parameter values can evolve. This requires a more abstract and generalised meta-environment and the support of a robust and flexible generative algorithm. This meta-environment has another name called 'meta-representational schema' which utilises different specialised representations or 'specialised design schema'.

Environments play a major role in influencing the whole generative process and particularly the selection of individuals. The external influences from the outside world, or the meaning of the general environment in Janssen's approach, have effects on the developmental process. All aspects related to these effects are referred to as epigenetic factors. Some researchers prefer to ignore the epigenetic factors when designing an evolutionary design system (Bentley, 1999a). This kind of evolutionary system only considers the genetic factors, the epigenetic factors are disregarded. In such a scenario, the general environment has no effects on the developmental process. This developmental process is started when embryogenesis occurs. Although the environment of the outside world has impact on the selection process, the difficulties existed in the implementation stage. Some solutions are proposed to conquer the most critical and foreseeable problems. But the crucial issue is the unforeseeable areas which only a small part of the science of nature has been scientifically discovered. Much effort is required to develop the research to construct a valuable symbiotic interaction between the general environment and design proposals.

3.3.5 Designer Interaction

A significant advantage can be achieved if the design process can be enhanced by utilising the power of computation and designer's knowledge. Designer interaction has a major role in developing the evolutionary design system. This allows an emergent environment to be born that design can gradually build with the support of evolutionary computation techniques and the designer's knowledge.

3.4 Review of System Applications

One of the key features developed in this proposed computational framework is to study the feasibility in applying the integrated SG and evolutionary algorithm approaches to product designs. Two system applications are reviewed to analyse the critical issues in related researches using similar approaches.

3.4.1 The First Application

System application one shows the use of a network approach for representation and evolution of SG (Kundu and Hellgardt, 1996). The approach is briefly reviewed and analysed. The limitations of the approach are discussed and suggestions are made. Also, the methodologies used in this approach which can be referenced to the study of the integrated SG and evolutionary algorithm framework are discussed. Finally a conclusion on critical issues is drawn.

A) Network Approach

A fixed grammar used in routine design tasks limits expressions or ability of the designers. With the introduction of an evolutionary mechanism to grammars, the grammars do not only produce designs to suit a peculiar design requirement but also capture domain specific knowledge.

Kundu and Hellgardt proposed to use a network model to represent SG which were evolved by an evolutionary mechanism (Kundu and Hellgardt, 1996). A genetic algorithm (GA) is adopted to evolve an augmented transition network (ATN)-frame. ATN was developed by Woods (1970, 1973) and was widely used to describe grammars for natural language understanding and question answering systems (Woods 1970, 1973). The framework is established based on the context of cultural expression and applications in architecture. This approach exhibits the 'learning' and 'reasoning' capability through the interaction of the system with the task environment.

The GA is used to evolve the arcs of the ATN. New arcs will be generated and some existing arcs will perish through the evolution process. The arcs of the network are designated with two numbers which are the evaluation values of the rules. These numbers are put into matrix notations. The corresponding values of each of these matrix entries are converted into binary coding and serially concatenated to form the genotype strings. The evolution process will apply the genetic operators to these strings and therefore new arcs will be produced. Consequently, this approach can be used to evolve grammar rules.

B) Elaboration

There are limitations in adopting this approach to a design task. Suggestions to solve the problems are made and the implications to adapt this approach in product design domain are discussed.

B1) Limitations of the Application

The grammar is limited to generate shape configurations which can only be quantifiably evaluated. It will be better to incorporate a visual control or designer's evaluation method to the framework for those hard coded evaluation functions. These functions include aesthetic standards in the designs of a specific field such as architectural and environmental domains.

B2) Performance of the Framework

While the domain specific knowledge is claimed to be 'an implicit quality embedded in the rules' by the 'Pittsburgh Approach', the quality of applying this knowledge to a design task is directly and subjectively determined by the quality of fitness functions. A question of how to specify good quality fitness functions which will particularly suit the proposed framework is not clearly revealed. As a result, the performance of the framework will be affected by the poor quality of the knowledge gained. It will be better to build up an analysis of the fitness functions in relation to different criteria when applying the rules.

B3) Implications

This framework demonstrates an insight regarding the possibility to evolve a SG. A further study in adapting the infrastructure of the framework can be carried out. The focus will be targeted to the consideration of a specific product design with assembly constraints. All the solutions suggested for the problems can be investigated to construct a more robust approach to apply an evolutionary mechanism to SG in the domain of product design.

C) Critical Issues

Despite the approach showing a generic approach to compute SG with an architecture design example “Palladio Villas”, what would be the representation change if this framework is adapted to other design domains liked product design. In the product design domain, the rules are constructed in accordance to functional requirements. Usually, most of the rules are ordered sequentially with constraints specified (Agarwal and Cagan, 1998; Agarwal et al., 1999; McCormack and Cagan, 2002; Pugliese and Cagan, 2002).

Although designers are given the flexibility of choosing alternative rules in different stages when determining which rules to apply, each stage is strictly followed in an organized structure. This structure is well organised as a guidance plan for designers to choose the rules. Since the structure of the rules does not allow modification without careful planning, this will greatly limit the application of evolutionary mechanism. The representation scheme of rules will require a lot of modification to make the rules more flexible for the evolutionary mechanism to become active and manipulate, particularly in the domain of product design.

3.4.2 The Second Application

System application two shows the use of an evolutionary system to evolve a dynamic template of grammar rules for the creation of architectural designs (Schnier and Gero, 1996). The review focuses on the methodologies used in research rather than the research context itself. An analysis is then performed on the possibility to apply the research methodologies to other domains such as product design. A number of limitations to the approach are revealed and potential solutions are proposed. A conclusion on critical issues is drawn based on implications of this research approach.

A) Learning Genetic Representations

SG rules have been done extensively by hand to capture the rules of designs from design examples and then produce designs by these rules (Chase, 1989). This involves a lot of effort in research about the designs and the design process. An attempt has been made to automate the process (Mackenzie, 1989). It reveals that a large amount of high level knowledge is required to realise the example designs. Another approach concentrates to evolve a higher level representation of form based on self-improving codes of a shape grammar (Schnier and Gero, 1996). This approach does not need much information as it turns a knowledge-lean representation into a knowledge-rich representation.

An evolutionary system was built based on this bottom-up approach where complex shapes are created by assembling smaller sub-parts (Gero and Schnier, 1995). During evolution, individuals are created that resemble the given example designs as closely as possible. Each individual is evaluated by the fitness function which determines the type and the relevance of description that belong to the given example designs. New evolved genes are created based on the successful combinations of low level genes. The coding of these genes will gradually contain more information about the example applications. By using both the original basic genes and the evolved genes to produce solutions, these solutions can adapt to the new design requirement and are biased to the given example solutions.

The design of Frank Lloyd Wright's prairie houses was chosen as an example to illustrate this approach (Gero and Schnier, 1995). The analysis of the style of Frank Lloyd Wright's prairie houses indicates that SG can represent both the common procedures and the common features used for designing the houses (Chan 1992, 1995). The common procedures are embedded in the rules and the rule sequences that are possible, and the common features that appear in the shapes which are manipulated by the rules. The shape rules taken for the design of Frank Lloyd Wright's prairie houses are based on the analysis done by Koning and Eizenberg (1981). The first 34 rules are used which specify 2-dimensional layouts, with a developed basic layout, organized into function zones, and some detailing.

The main idea is to establish different function zones such as service and living space around the fireplace. The basic coding used in Gero and Schnier (1995) are not enough to capture information about the functional organization. A new coding method is introduced to integrate information about the semantics of the shapes. Semantic information can therefore be attached to the outlines by adding a set of lines of different types. The information including the different semantics or function of the rooms is captured in the basic coding. The line types in the basic coding are evaluated by fitness functions with two criteria. The first criterion specifies that any individual produced has to fit the design both in line types and in shape. The number of line types in the basic coding does not have a limit, it can be larger than numbers used in the design example. The second criterion specifies the interpretation and depends on the way the line types are used. These two criteria can be mixed in use. Colour grammars can be referenced when the coding of line types uses different colours, the difference is the colours do not have any semantic value attached in a colour grammar (Knight, 1994b).

The learning process is by means of evolving the basic genetic representation to a more complex genetic representation which is closer to the given design examples. The basic genes are replaced by evolved genes if these evolved genes match part of the design features of the design examples.

The genetic coding consists of two parts, the first defines the primitives and the second part defines the attribute values. A total of four genes are used with the first gene defining the types of primitives, i.e. either a line or a turn, and the remaining three genes defining the five line types.

The fitness calculation is based on the percentage of similarity between the phenotype representation of an individual as matched to the given design example. To avoid the problem of convergence, in that particular evolved individuals can dominate the population, a 'niching' effect is created. This 'niching' effect is achieved by assigning higher fitness values to the few other individuals that describe aspects of the case.

The evolved representation is used to create new floor plans. The new floor plans are influenced by the fitness function that evaluated the designs. The topological constraints have to be specified explicitly in the fitness function as the evolving coding does not support the automation of constraint satisfaction. The 'Pareto optimisation' is used to solve the problem of multiple fitness problems (Radford and Gero, 1988). In order to prevent the problem of convergence mentioned previously, 'niching' is used (Horn and Nafpliotis, 1993). Successful results are found for the different settings of the fitness functions.

Finally, since the methodology does not include the topologies for the designs, the topologies of the design can be added by increasing the numbers of line types with meaning. To achieve this goal, a checking of the correct context is required for the use of line types in the designs. This approach learns design knowledge from design examples and opens up the possibilities in case-based design.

B) Elaboration

This approach does not clearly show the differences among the evolution of genes and the evolution of the designs or evolution of both. It will be better to illustrate with more examples to distinguish the differences among them. For example, if using the evolved genes to designs, what should the fitness functions be defined for genes or for designs.

Limitations of Application

This approach limits evolving genes to the production of houses in the style of Frank Lloyd Wright. It seems that the evolved genes can be further categorised in terms of their function purposes. The methodologies can be modified into four steps. First, vectors can be evolved to form more complex genes. Second, if successful genes are found during evolution, reduce the number of genes connected to a particular design to form evolved genes. Third, the evolved genes are categorised into different groups in accordance to their functions. Finally, the evolved genes will be used to generate new designs to fulfil new function requirements.

C) Critical Issues

This approach opens up the possibility for learning from design examples using an integrated SG and evolutionary algorithm approach to design in the architectural field. An investigation can be done on the application of evolving shape grammars to design in the product design field. This approach can be referenced for evolving simple two dimensional sketches which can be extracted to form three dimension objects in a computer aided design system. However, the design examples taken should be selected such that these designs can be represented by the basic coding or evolved coding genes.

3.5 Computational Issues

3.5.1 Design Optimisation

Evolutionary programming and GA have been widely used to solve numerical function and combinatorial optimisation problems. For the numerical problems, both constrained (Michalewicz and Schoenauer, 1996; Kim and Myung, 1997) and unconstrained (Yao and Liu, 1996; Yao and Liu, 1997) numerical function optimisation problems have been solved by evolutionary algorithms. Also, research on multi-objective optimisation by evolutionary algorithms has been carried out (Fonseca and Fleming, 1995, 1998).

Combinatorial optimisation problems, such as the travelling salesman problem (Grefenstette et al., 1985; Fogel, 1988; Yao, 1993), transportation problem (Vignaux and Michalewicz, 1991; Michalewicz, 1992), switchbox routing in integrated circuits (Lienig and Thulasiraman, 1995), cutting stock problem (Hinterding and Khan, 1995; Liang et al., 1998), lecture room assignment problem (Luan and Yao, 1994), etc have been tackled by evolutionary algorithms. Compared with more traditional approaches, the results generated by evolutionary algorithms are quite competitive.

In addition, a few research works on optimisation relevant to industrial applications are introduced here. For engineering and operation research, the GA applications include the control of a GA pipeline in steady-state and transient conditions (Goldberg and Kuo, 1987; Goldberg, 1989). Other applications such as communications network link size optimisation is handled by using a GA plus advanced operators (Coombs and Davis, 1987; Davis and Coombs, 1989). This hybrid GA is developed for optimisation with both an integer representation and several problem-specific operators to choose suitable link capacities for wide-area packet switched networks. For semiconductor industries, the penalty function approach is applied to the VLSI circuit layout via GA (Rahmani and Ono, 1993). A GA program is used to figure out a version of the channel routing for the circuit board. More research works on optimisation of designs will be discussed in Section 3.6.

3.5.2 Design Exploration

The use of evolutionary computation techniques in generating designs is becoming a growth research area (Bentley, 1998). A typical approach is to define design problems in terms of search. All the possible solutions to the problem are filed and confined within the search-space where a point defines a solution (Kanal and Kumar, 1988). The evolutionary algorithm is used as a searching mechanism to search for better solutions which in turn improve a design. This approach is successfully applied to optimise designs (Holland, 1992). Bentley has modified this approach to explore designs from scratch without any existing knowledge (Bentley, 1996). However, it is not appropriate to view the design process as a searching process in which distant alternatives in the vast design solution space cannot be compared in any straightforward manner (Janssen et al., 2002).

Another approach takes a view of the design process as a direct analogy with the evolutionary processes of nature (Frazer, 2002). This approach can be described as the application of "GA" to simulate the design process (Holland, 1975; Frazer, 1995). Jian has adopted this approach to automate the design process in a CAD system (Sun, 2002). Two of the branches of design exploration: form creation and product conceptualisation are introduced as follows:

3.5.3 Form Creation

The evolutionary design techniques have been adopted by engineers, architects and researchers in the generation of forms. The design process of form creation is analysed and simulated by these techniques.

The approach of form creation using novel data structures for rule-based and genetic design is introduced (Frazer, 1992). The data structures employ the concept of a mote which is a minute particle in a regular geometrical array. A mote consists of a set of rules knowing its own status such as positions and reasons for its existence and historical records of its information. The information for a mote also includes relationships between other motes and its own properties as a parameterized geometrical described structure to which it might map. The identity or properties of a mote may move when transformation is required but the mote itself does not move. The motes are grouped into two separated structures, one simulates an artefact and the other simulates the environment. Since information travels through the array in the form of logical fields, the form of the structure may change its behaviour and performance in responding to an environment. This approach is analogous to a genotype producing a phenotypic reaction. The successful rules of generated forms can be learned and new rules for improving its form usability can be evolved.

Other approaches have been proposed in the creation of forms using evolutionary design techniques. Soddu uses a morphogenetic approach in generation of forms (Soddu, 1994). A series of generative procedures are used within a dynamic chaotic system to produce designs. The codes of generation and control are guided by the logic which is an emulation of the designer's subjective procedures. Each design generated by the dynamic chaotic systems is unrepeatable (Soddu and Colabella, 1997). Another approach concentrates to evolve a higher level representation of form based on self-improving codes of a shape grammar (Schnier and Gero, 1996). This approach can be used to produce house designs by evolving genes limited to the style of Frank Lloyd Wright. More research work on exploration of form creation will be discussed in Section 3.6.

3.5.4 Product Conceptualisation

The generative and evolutionary design techniques can be applied to assembly of products as well as individual component design. For assembly application, an attempt to evolve the relationships and arrangement of high-level design concepts was made to generate novel preliminary designs (Pham and Yang, 1993). A prototype was developed which used a GA to evolve the organization of a set of conceptual building blocks such as rack and pinion, worm gear, belt drive. The possible networks of interconnected conceptual building blocks are searched through by evolution. These systems employ simple representations of genotype and phenotype, with rudimentary embryologies, if any.

Another approach used by Taura and Nagasaka can tackle two aspects of product conceptualisation, designing 3D shapes and their layouts (Taura and Nagasaka, 1999). An adaptive-growth-type 3D representation is developed for both individual shape design and configuration design. The important notion of representations that can adapt to change in environment and constraint is addressed. An evolutionary design system is used to define the morphology. The morphology describes the density of 'cell' growing on parts of a sphere. Instead of generating shapes directly, the determination of where and how the 'cell' is placed on the sphere is done by evolution. More research work on exploration of product conceptualisation will be discussed in Section 3.6.

3.6 Evolutionary Designs

There are several approaches which could be used within the developmental step of an evolutionary system in order to create alternative evolutionary design systems. This section classifies two types of approaches: 1) The potential approaches which could be modified to adapt or integrate with an evolutionary framework and 2) The approaches which have been implemented in developing the evolutionary systems for evolutionary designs. Some potential approaches have been integrated in the evolutionary framework which will be shown in both sections 3.6.1 and 3.6.2.

3.6.1 Potential Integration Approaches

A) Integration of Shape Grammars and Evolutionary Algorithms

'Artificial embryology' can be used in the mapping of genotypes to phenotypes. It exhibits the ability to describe some forms of 'artificial growth' process with an existing representation. A SG can be used as 'artificial embryology' to specify how shapes should be artificially 'grown' using a given representation (Todd and Latham, 1992; Sims, 1994a, 1994b; Rosenman, 1996a, b, c). A thorough discussion in developing a potential approach using SG integrated with evolutionary algorithms to design has been given in previous chapters (Chapter 1 and 2). All the critical issues and problems have been identified and related researches have been reviewed.

B) History Based Parametric Technique

Another potential approach which can be modified to suit an evolutionary framework is the history based parametric technique. It is based on the concept of a graphically interactive parametric modeller in which the designers can create a master model. In the master model, the designers can input parameters to the system and specify the constraints that will define the model through a completed description of its components (Monedero, 2000). In general, a procedure for the generation of parametric forms is defined. By modifying the constraints in this model which are associated with its procedure, a diversity of forms can be created.

C) The FormGrow Program

The first example of using a history based parametric technique to design is a program called FormGrow which was developed by Todd and Latham (1992, 1999). Artists can use FormGrow to generate abstract organic 3D forms with a set of growth rules. The rules specify the basic growth process which can duplicate a specified input form with a series of translations and transformations. As a result, a compound form can be generated.

D) The Xfrog Program

The second example of using the history based parametric technique to design is a program called Xfrog which was developed by Lintermann and Deussen (1999). Xfrog can create realistic flowers, bushes and trees with different hierarchical structures. Central in the program is a graph which encapsulates the generative rules for the creation of those hierarchical structures. The designers can create the set of nodes and links of the graph using a graphical interface.

E) Grid Based Substitution Technique

The third potential approach which can be modified to suit an evolutionary framework is the grid based substitution technique. Among all the grid based substitution techniques, the best known one is the cellular automata. A simple 1D cellular automata is made of a line of cells of fixed length. With two simple states: on or off in each cell specified by the transition rules, a variety of patterns can be generated. It is important to define the seed pattern of the cellular automata which may compose any configuration of "on" cells. The final pattern is determined either by running after a predefined number of time steps or when a desired pattern emerges during the generation process. Quijano and Rastogi, and Stefan Seemüller have generated a variety of structures created by 3D cellular automata programs. (Frazer, 1995, p. 92-93, p. 46-47). The programs which were developed by Quijano and Rastogi, and Stefan Seemüller, worked on an orthogonal cubic grid.

3.6.2 Evolutionary Design Systems

A) Architectural Design Domain

Several typical systems using evolutionary design approaches are reviewed and the key features of each system are described. Design examples are used to illustrate the exploration and optimisation capabilities of the evolutionary algorithms which are applied in the architectural design domain.

A1) GS

The first evolutionary design system to be reviewed is a parametric evolutionary design system developed by Caldas (2001). The system called the Generative System (GS) for Design Optimisation, is designed for optimisation and used in the building design domain. The key feature of this system is the integration to DOE-2 which is an existing simulation application for evolving low-energy building designs. The GS has been used to optimise facade design.

More specifically, the sizes and positioning of window openings in a facade, or together with overhang sizes that supply shade for windows, have to be optimised. The primary goal of the GS is to reducing the annual energy consumption of the building, in which the DOE-2 application takes into account both space conditioning and lighting, and the climate for that geographical location. Caldas and Norford (2001) presented a variety of alternative facades generated by the GS; those facades were designed for the climate in Oporto.

A2) Design Schema

The second evolutionary design system to be reviewed is a generative evolution framework developed by Janssen (2004). The framework is developed through the formulation of the concept of a design schema which targets to solve the "variability problem" in generating designs. A design schema is a design entity which captures the essential and identifiable characters of a family of designs. In constructing the framework, a design method and a computational architecture are required to be developed. The design method concerns encoding the design schema and using such encoded schema to evolve a specific design. The architecture provides a set of routines in which the encoded schema consisting of the rules and representations is encapsulated.

Janssen (2004) has demonstrated the process of encoding the design schema with examples. One particular example requires the schema to be designed for a family of multi-story buildings which are constructed using standard concrete frame construction. The design schema encompasses the essential and identifiable characters of a family of multi-story buildings, and includes all the related issues of aesthetics, space, structure, materials, and construction. Janssen (2004) presented the results which were created using the generative process. The generated designs reflected the effects of applying the characters of the design schema. By analysing the main features of these designs, their variability in terms of many different criteria such as the overall building form, the organization of spaces, and the treatment of facades has been clearly illustrated.

B) Engineering Design Domain

B1) Yacht Hull Forms

An interactive parametric evolutionary design system developed by Graham et al. (1993) (see also (Frazer, 1995, p. 61, 2002)), is designed for optimisation and used in the engineering design domain. An example of optimising the performance of racing yacht hulls using the system is described. The system can be analysed from its three main parts: 1) Representation, 2) Manipulation, and 3) Evaluation.

For the first part (Representation issue), the phenotypes are a set of parameters which are used to represent curved surfaces of yacht hulls. The parameters are a set of control points which are used to define the curves of the hulls profile. The GA manipulates the set of control points and modifies their corresponding coordinates to generate alternative yacht hulls.

For the second part (Manipulation issue), an adaptation of the genetic algorithm (GA) is used in the development of the system. At the beginning of running the system, an initial population of hulls are randomly generated.

Each hull is calculated with a prismatic coefficient (which is a measure of the fullness of the hull) by the analysis program. Meanwhile, designers judge the scores for the hulls. The GA will then apply the Pareto optimality test with the two scores obtained from the analysis program and the designers' evaluation. The Pareto test is used to solve the conflicting criteria between different objective functions. In this case, the conflicting criteria are the evaluation scores obtained by the analysis program and the designers' evaluation. The Pareto test is to figure out the better solutions within the population in terms of both criteria. Each hull is tested with its "degree of domination" within the population. A biased random selection is used in the selection of parents. The GA then generates a new population and repeats the evolutionary process. The quantifiable criteria can be optimised by the 'natural selection' process. When a GA is running with the 'natural selection' process, 'artificial selection' can be periodically interrupted to include the criteria which are justified by designer's preference.

For the third part (Evaluation issue), the objective function concerns the measures of efficiency including stability, centre of buoyancy, wet surface area, prismatic coefficient, block coefficient and so on. With these measures provided, the hull performance and potential speed can be roughly estimated. A weighting is assigned to all these criteria and is justified based on the use of the hull (racing, day sailing cruising etc). The weighting for each criterion will be assessed to produce a fitness score. Each design is tested and evaluated in a simulated environment. Design information such as data on displacement, wet area, and block coefficients is considered. Besides, ergonomic data are derived from the sections, and aesthetic and intuitive judgements are derived from considering graphic displays of the boat lines. Therefore, ill-defined and conflicting criteria can be considered in this technique.

B2) GADO

A parametric evolutionary design system developed by Rasheed (1998), called the Genetic Algorithm for Design Optimisation (GADO), is designed for optimisation and used in the engineering design domain. In designing supersonic transport aircraft with the GADO, a simplified model was used and a series of experiments has been conducted to optimise the simplified model (Rasheed and Davison, 1999). Rasheed (1998) presented the simplified model of a supersonic aircraft with certain parameters being optimised by GADO.

C) Creative Design Exploration Design Domain

GADES

A generative evolutionary design system developed by Bentley (1996, 1999a), called the GADES (Genetic Algorithm Designer), is designed for exploration of many different types of designs in different application domains. The key feature of this system is that it can consistently evolve good designs "from scratch" for a wide variety of design problems without human intervention. The system can be analysed from its three main parts: 1) Representation, 2) Manipulation, and 3) Evaluation.

For the first part (Representation issue), the phenotypes represent the designs as solid models which are defined by a composition of 'clipped stretched cubes'. The 'clipped stretched cube' is a spatial-partitioning representation of solid objects with a low-parameter, which is capable of approximating curved surfaces more precisely (Bentley and Wakefield, 1996a). It is a solid cut from a six-sided polyhedron with all sides at right-angles to each other, the cutting plane is defined relative to the centre of the polyhedron. The genetic coding of the genotypes follows the structure of the phenotypes. Early versions of Bentley's evolutionary design system used real-coded genotypes with alphabets of high cardinalities. Later versions of the system employ the binary coding of parameters as genotype representation.

For the second part (Manipulation issue), the manipulation of the phenotypes and genotypes and is mainly performed by a modified GA, which creates solid models (phenotypes) from the genotypes through a complex generative process. For the third part (Evaluation issue), the evaluation is performed on the new individuals by multiple objectives and design constraints, which are related to size, mass, surface area, stability and aerodynamics. A specially developed multi-objective ranking technique called Sum of Weighted Global Ratios (SWGR) is chosen to provide a single overall fitness value for each individual. This method is range-independent, supports importance, and consistently produces not just Pareto-optimal solutions, but good designs (Bentley and Wakefield, 1996b). Bentley (1996, 1999b) presented examples of sports car design and table design at different stages of evolution.

D) Product Design Domain

D1) Argenia

Soddu has developed a generic generative design system called Argenia which applies the morphogenetic approach to generate forms (Soddu, 1994). The Argenia is a dynamic chaotic system which encompasses a series of generative procedures in producing designs. The key feature of the system is the logic, which is an emulation of the designer's subjective procedures, to guide and control the codes of generation. Each design generated by the system is always unrepeatable (Soddu and Colabella, 1997). A wide range of applications using Argenia have been developed to generate objects from novel table-lamps, to castles and 3D sculptures in Picasso's style.

One of his examples using Argenia is to design chair, the normal designing procedures of chair are represented and formulated as a 'logic' in a hierarchical way, from the overall form to the details. Each generated chair is a specialized design which means it is uniquely produced by the system. The design characters developed within such a huge number of alternative, marvellous, post-modern and fashionable designs reflect the power of this approach in formulating the design 'logic' as reprogramming actions. In Argenia, designers can interact with the system by artificial selection.

D2) Hierarchical Evolution

Chan et al. (1999a, b, 2000, 2001, 2002) have developed a computational system framework for enhancing design in an interactive evolutionary manner. The key feature of this framework is to provide a structure for supporting design activities at the conceptual design stage. Different levels of representation and manipulation are allowed to coexist in this framework. What makes this framework a potential power structure as used in an evolutionary system is its organisational capability in handling the complex design information and the control capability in manipulating the design activities in an interactive evolutionary process.

With this framework, designers can participate in the development process of a solution in a hierarchical manner by interactively manipulating the design data. In addition, this computational framework supports the exploration and adaptation capabilities through the integration with different computational evolutionary and generative modules.

Chan et al. (2002) presented an example of designing wine glasses to demonstrate the power of such a framework as used in the exploration of product designs. The evolutionary framework allows the genetic algorithm to solve the parametric tuning problem. The GA controls the generation of the profile curves of the wine glasses with designers' evaluation through interactive artificial selection. A small population of evolving designs (phenotypes) which were decoded from a string of bits (genotypes) were visually displayed on computer screen. In each generation, the mutation and crossover operations were performed on the populations with the selected phenotypes having higher survival chances. The evolutionary process continued until satisfactory solutions emerged.

D3) Concept-seeding Approach

The theoretical development of a concept-seeding approach originated from Frazer (1995). The first implementation of the generative evolutionary design system that explicitly adopted the concept-seed approach was developed by Sun (2002), one of Frazer's Ph.D. students (Frazer et al., 1999). The generative evolutionary design system developed by Sun is focused on supporting product design, and has been tested for designing mobile phones, remote controllers and other hand-held products. Sun (2002) presented a range of designs using mobile phones as examples that were created using the evolutionary design system.

In Sun's thesis, the concept-seed concept is divided into two new representations, referred to as rudiments and formatives (Sun, 2002). The definitions of rudiments and formatives are described by Sun (2002, p. 52): "A formative is an encapsulated potential design solution, which defines a set of entities and relations, as well as the generative rules involved during the generative process. A rudiment is a composition element of the formative, which defines a set of entities and related design knowledge. In the domain of product design, a potential product design solution corresponds to a formative and it contains the constitutional parts of a product structure, the relationship of these parts, and the configuration rules to build the product embodiment."

In order to demonstrate the feasibility of applying the concept-seeding approach in an evolutionary design system, Sun has built a prototype system which consists of three main parts: a database, a graphical interface and an evolutionary system (Sun et al., 1999 and Sun, 2002, p. 118–120). The database is for storing the rudiments. The graphical interface is for constructing the formatives. And the evolutionary system is for enabling designers to explore, evaluate and visualise a huge number of resulting designs in an interactive manner.

3.7 Summary

This chapter has described and discussed the working principles, computational issues, basic technical mechanisms, applications and all the related properties and issues of evolutionary algorithms. The key points are as follows:

- An overview of the evolutionary algorithms in supporting the design activities has been given. The working principles of evolutionary algorithms have been introduced with discussion of the background of evolutionary algorithms. The background outlines the original conceptual formulation of the idea of evolutionary computation from natural evolution, the terminologies used and the difficulties in implementation.

- Researches on different types of generative and evolutionary design techniques have been described and in particular, the GA is focused. The formulation, representation and manipulation of the GA have been described and discussed. A thorough description and discussion on the effects of environment and designer interaction has been performed to illustrate how the evolutionary algorithms work for designing in different situations.

- Two system applications have been reviewed to analyse the related researches on the applications of integrated SG and evolutionary algorithm approaches which are the key subject to be investigated in this research. The approaches adopted in each application have been elaborated in order to search for solutions in enhancing the generative capability of these approaches. After elaboration, the limitations and performance of the applications have been discussed. The implications and critical issues of implementing such approaches to design have also been revealed. In this way, the elaboration of the approaches adopted in these two system applications is helpful in developing the completed proposed systematic approach for the construction of the two prototype systems.

- The critical computational issues in the implementation of evolutionary algorithms have been discussed. The applications of evolutionary algorithms for design optimisation and exploration, form creation and product conceptualisation have been described and discussed in detail.

- The related research works which have been developed using generative and/or evolutionary approaches have been classified into two types of approaches: 1) The potential approaches which could be modified to adapt or integrate with an evolutionary framework and 2) The approaches which have been implemented in developing the evolutionary systems for evolutionary designs. The key features of each system using the two types of approaches are clearly identified and discussed. Particularly, the historical development and the current state-of-art for the development of many applications in different application domains using evolutionary algorithms have been reviewed. The key features of each application system have been highlighted in various design domains such as: A) Architectural Design Domain, B) Engineering Design Domain, C) Creative Design Exploration Design Domain, and D) Product Design Domain. The critical issues of implementing different approaches in each application have been highlighted.

In summary, the integration of an interactive evolutionary system with shape grammars remains a new and challenging topic of research, since it involves complex processes, introducing parameters into SG rules in order to explore designs. In the coming chapters, a theoretical framework for integrating SG with Interactive Evolutionary System (IES) will be presented with implemented systems and applications.

Part III Theoretical Development

Part three presents the theoretical basis of the integrated SG and evolutionary algorithm framework using the systematic approach. This part consists of one chapter.

Chapter 4 (The Theoretical Framework) describes the key procedures of the systematic approach to developing the evolutionary IGBDS with two key elements: the parametric SG and the evolutionary architecture. The detailed implementation of the systematic approach is illustrated with two prototype systems which will be described in the next two chapters (Chapter 5 and 6).

Chapter 4

The Theoretical Framework

4.1 Introduction

This chapter presents the systematic approach for supporting the development process of SG. It consists of five main sections: Sections 4.2 to 4.6, each of which highlights certain key procedures in the development of SG. The descriptions of each section are as follows:

- In section 4.2, the systematic approach for supporting the development process of SG is described.

- In section 4.3, the two key issues in the construction of an information network of SG: 1) Identification of product design characteristics and 2) Redefining specific SG, are discussed.

- In section 4.4, the construction of the Core Variant Design Feature Library which stores the relevant information of SG in a well organised manner, is discussed.

- In section 4.5, the development process of parametric SG in accordance to the interrelationships among the components specified in the Core Variant Model, is discussed.

- In section 4.6, the construction of an evolutionary architecture in which evolutionary algorithms are used as adaptation mechanisms in the exploitation of the generative capability of SG rules, is described.

This chapter outlines the key procedures in the development of SG. The details of each part will be described in the coming two chapters which present the development of SG using the systematic approach with two prototype systems.

4.2 Overview of Systematic Approach

The aim of the systematic approach in the development of SG is to deepen our understanding of how shapes and spatial constraints can be extracted from the existing design to help with the new designs which have different requirements. This requires a system of frameworks that operate on a level at which design knowledge can be generalized as applicable rules and primitives for new shape generation and configuration. In order to achieve this, the systematic approach specifies four main procedures for the development of SG in product design domain (Figure 4.1).

These four main procedures are:

1) the construction of an information network of SG,
2) the construction of a Core Variant Model,
3) the development of parametric SG, and
4) the construction of an evolutionary architecture.

The four main procedures are briefly described as follows:

1) **Information Network of Shape Grammars**
An information network of SG is constructed to link up all the relevant information of SG related to the existing designs analysed. The information network of SG helps the designers or SG developer to understand the design requirements, thus helping with discovering, formulating and solving the new design problems, and addressing the key issues related to these problems more clearly.

1A) Identification of Product Design Characteristics

The first main procedure is to analyse essential and identifiable design characteristics from the existing products. During the analysis process, the designers can gain knowledge by learning why those identified design characteristics were created, for what purposes and how they influenced the common concerns of design requirements such as usefulness, usability, functionality, necessity, implementation possibilities, needs for development (why and how), rationality, feasibility in design, emotional appraisal or response, culture elements and etc.

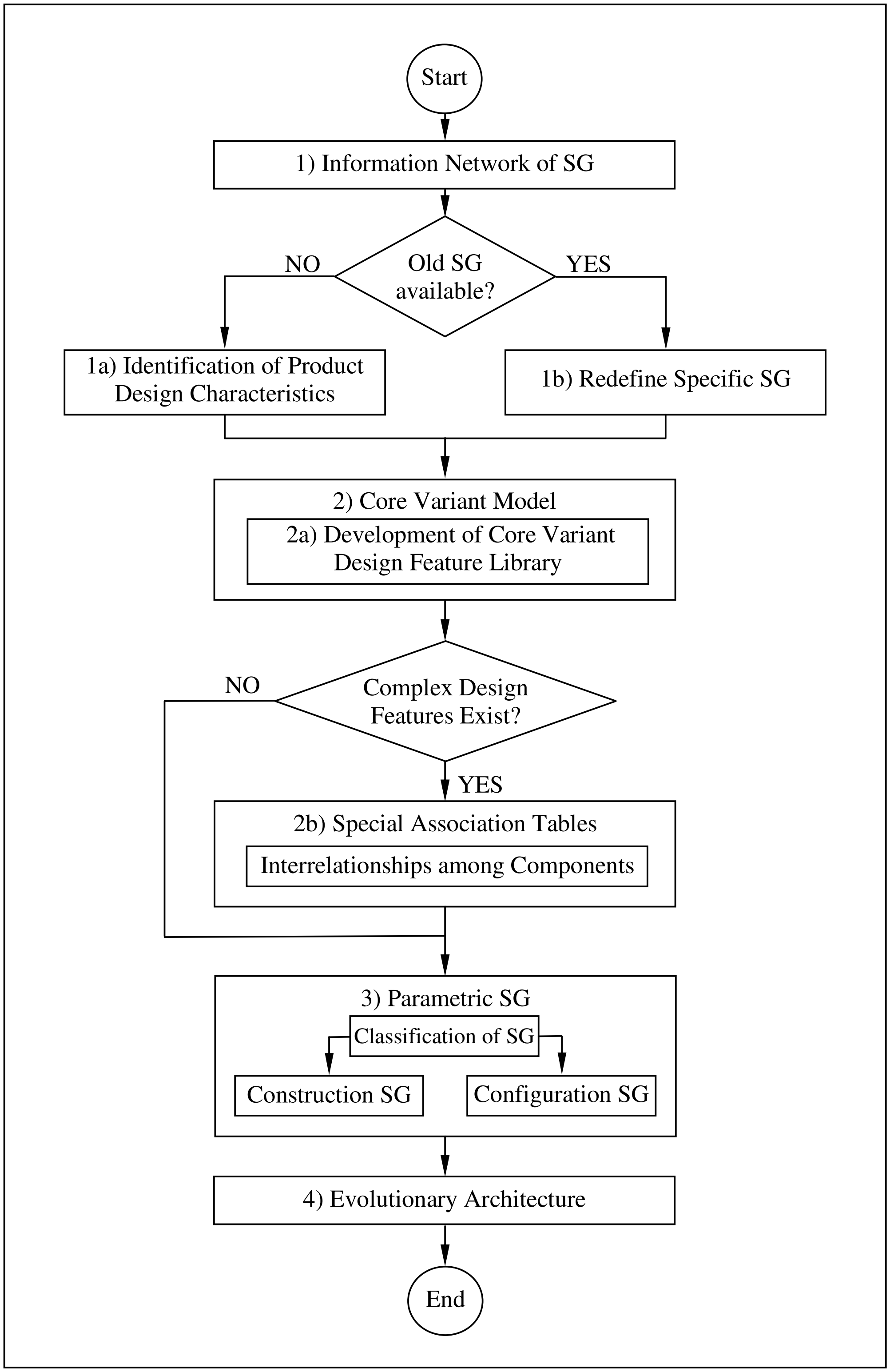


Figure 4.1 Systematic approach in the development of shape grammars for product designs

1B) Redefine Specific Shape Grammars

The procedure in redefining the specific SG is optional if there are existing SG which have been developed for specific applications previously. Through observation from many different perspectives, the designers can redefine the specific SG accordingly. The aims of this procedure are to let the designers learn how to apply creative and innovative ways of thinking in redefining the specific SG for more generic applications or fulfilling different design requirements. During the redefining SG process, the designers can learn the technical design skills in mastering the development of SG. Another significant advantage is that the designers can enhance the ability to understand the design requirements, and, discover, formulate and solve the design problems, and address the key issues related to these problems more clearly with the systematic approach.

2) Core Variant Model

Based on the analysis results, the second main procedure is to organise and extract the usable information from the information network of SG to build a Core Variant Model for particular design applications. The Core Variant Model specifies the flexible structure of a class of products in which the interrelationships among components are controlled by the SG.

2A) Development of Core Variant Design Feature Library

The heart of the Core Variant Model is the Core Variant Design Feature Library which stores all the relevant information of SG in a well organised manner. In the library, standard association tables are used to list all the relevant information of SG.

2B) Special Association Table for Design Features

Special association tables are required to organise the extra information and attributes for complex features of components such as shapes, material, costing information and etc. The interrelationships among components are also specific in the special association tables.

3) Parametric Shape Grammars

After the creation of an information network of SG by the analysis of the existing designs from different perspectives, a Core Variant Model can be established by extracting and organising usable information in the information network of SG. With this Core Variant Model, the third main procedure is to derive the parametric SG in accordance to the interrelationships among components specified in the Core Variant Model.

Classification of Shape Grammars

The SG rules can be classified from many different perspectives such as the geometric or functional points of views. In the product design domain, the generation of components and the configuration of the generated components in an assembly are critical. The SG rules can then be classified into two main categories: Construction and Configuration SG rules from the functional points of views for product design domain accordingly.

4) **Evolutionary Architecture Integration**
The fourth main procedure is to apply evolutionary algorithms to evolve the SG for exploring new designs to fulfil new design requirements. An evolutionary architecture is required, developed and integrated with an IGBDS. Following the main procedures from one to four, the existing products in a particular design domain are analysed to derive shape features in the form of SG rules. The SG rules are created with 3D labelled shapes and represented by both SG and genetic representations. An evolutionary algorithm is applied to evolve the genetic SG rules to generate new designs. With such an evolutionary IGBDS, both product component designs as well as product configuration designs are supported.

4.3 Information Network of Shape Grammars

Since design problems are complex, multi-aspect, dynamic, and ill-structured, the designers need a systematic approach to get benefits from the knowledge of existing designs for them to solve the design problems. An information network of SG is constructed to achieve this goal by better organising the relevant information and analysis results of the existing designs (Figure 4.2).

The information network of SG composes all relevant information used for the construction of SG, a Core Variant Model, and standard and special association tables. The Core Variant Model specifies and organises the interrelationships among components. In the Core Variant Model, standard and special association tables are used to facilitate the use of relevant information of SG to store such information in the Core Variant Design Feature Library.

During the construction process of the information network of SG, the designers participate in understanding the design requirements. The designers can discover the design problems and learn how to formulate them. In this way, the designers can accumulate knowledge by analysing the existing designs and enhancing their sensitivity in observations to analyse the existing designs. As a result, the designers can learn how to solve the design problems and address the key issues related to these problems more clearly.

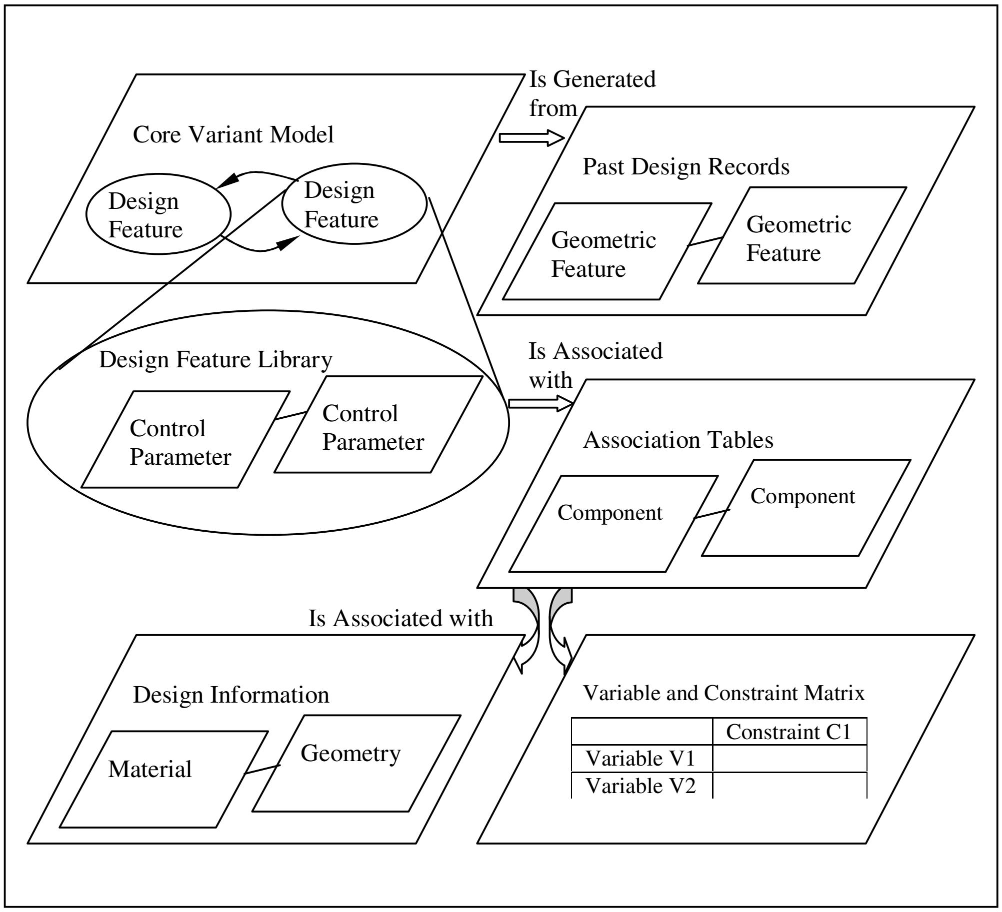


Figure 4.2 Information network of shape grammars

4.3.1 Identification of Product Design Characteristics

The existing products are collected to analyse essential and identifiable design characteristics of the designs. Similar design characters can be identified from many different perspectives. For example, design features of existing designs are identified and grouped based on geometric properties or decomposition of functional components. Each component of a particular type of products has to be analysed in thorough detail in order to identify particular design characteristics which are of interest in particular design applications. For instance, form features of products can be abstracted from several different brands. These form features representing different components are categorised according to their geometric locations in the assembly. For industrial and engineering products, the study of the working principle of existing products is important for the realisation on how the existing designs can fulfil the design requirements.

4.3.2 Redefining Specific Shape Grammars

The designers can take advantage of avoiding the analysis of the existing designs from scratch if there exists specific SG available for particular design applications. Since the existing SG is specific for the generation of a particular type of design and for specific design requirements, the generated designs cannot fulfil new design requirements. In this case, new SG rules can be derived by redefining the existing SG rules.

The redefining process involves the analysis of the design characteristics of the existing designs from many different perspectives such as functional decomposition or geometric points of views. New SG can be constructed based on analysis results. To define a new set of SG rules to be more generic, potential modifiable elements of the existing specific SG are identified. The potential modifiable elements include the vocabulary of shapes, spatial relations, rule sequences and rules themselves. All these elements have to be opened up for the designers to redefine intuitively. These redefined shapes and rules allow the designers to accommodate their preferences for the creation of new designs. As a result, alternative designs as well as configurations can be generated. Since the two prototype systems developed in this research employ entirely new SG, this procedure does not apply to these two prototype systems. An example of applying this procedure can be referenced to the review section in which the pattern design SG redefines the specific Chinese lattice SG for more generic design applications.

4.4 Core Variant Model

A Core Variant Model is set up by extracting and organising usable information in the information network of SG. The Core Variant Model is developed in the previous research to organise the information for product design (Lee et al., 2000). The basic structure of the Core Variant Model is modified and adopted in this research for modelling and organising the information required for the development of SG. Since the information contained in the information network of SG can be enormous and the process time required too long. A Core Variant Model is used to abstract and extract all the relevant information contained in the information network of SG for specific design applications.

The Core Variant Model aims at centralising the design information for the development of SG. On the one hand, the well organised hierarchical structure of the Core Variant Model supports the management of the interrelationships among components. On the other hand, one may suspect that the predefined structure of designs hinders the power of the standard SG approach with its emergent properties.

In order to tackle this problem, there are strategic methodologies to integrate the strength of the standard SG approach and the well organised structure of designs.

First of all, one of the objectives in the development of the information network of SG and the Core Variant Model is to facilitate the ease of use of design information for the development of SG. The structure of the Core Variant Model reflects the current understanding of the design requirements selected by the designers and in itself masters knowledge of existing designs to perform the design tasks. Both the objective and approach match with the standard SG approach with the difference that the systematic approach organises the design information in a well organised structure with a systematic manner.

Second, the systematic approach allows flexible modification in scope and specification of the objects of the Core Variant Model. This is achieved by modelling the objects of the Core Variant Model with different levels of abstraction. For example, the description of an object of the Core Variant Model specifies the exterior form designs of the digital camera. The specification of the object can simply be specified with geometric description of forms like rectangle shape or abstract description of design requirements like "to design a form for elegant appeal". In this way, the objects to be described in the Core Variant Model do not limit themselves at one level of abstraction and more diversified versions of designs can be generated.

Third, there are concerns related to the standard SG approach in that the generated designs do not have a predefined structure or are non-deterministic in nature. This issue can be addressed in the two phases to modify the Core Variant Model. For the first phase, the hierarchical representation of the Core Variant Model structure can be expanded or contracted in size during each revision of design requirements. The extension in size of the Core Variant Model structure allows more detailed design specifications to be specified. In this case, the designers expect that the designs will follow a routine design mode and the creativity is not so high. In contrast, the contraction in size of the Core Variant Model structure allows fewer restrictions on design specifications. In this case, the designers expect that creative designs can be generated. A balance on the detail of specifications for the Core Variant Model has to be determined, tailored, suited and revised for each revision of design requirements.

For the second phase, the objects in the Core Variant Model can be modified easily for the generation of non-deterministic shapes. The shapes specified in the objects of the Core Variant Model can be free forms provided that they suit all the design requirements. For example, in this thesis, the exterior form designs of the digital camera can be generated and modified by evolutionary algorithms. Although the forms are not generated with the "SG emergent" shapes using the maximal representation of SG, new designs can be generated by this computational framework. Furthermore, this Core Variant Model can potentially be integrated with the standard SG approach using maximal representation. The SG developed in this research focuses on using labels and other attributes to integrate with shapes. Since the study of maximal representation of shapes is not the focus of this research, this requires further research to study the use of maximal representation of shapes with the systematic approach.

4.4.1 Core Variant Design Feature Library

The design features, constraints and all related information, which have interrelationships among themselves, are represented by objects. The objects can be organised into hierarchical structures to represent the interrelationships among objects. With this hierarchical representation of design features, a Core Variant Design Feature Library can then be established to allocate all the design information into a Core Variant Model. The design information includes geometric properties, constraints, spatial relations among design objects, interrelationships among design features, construction methods and all other related entities and attributes. The hierarchical representation can be expanded or contracted in size and the level of abstraction for each object can be changed during each revision of design requirements.

4.4.2 Standard Association Table

In order to systematically retrieve the design information from the Core Variant Design Feature Library, association tables are designed to store all the relevant design information. The association tables are classified into two types: standard and special types which address the design information from different levels of complexity. In this section, the standard type of associated table is described in detail whereas the special type will be described in the next section.

The standard association tables are designed to facilitate the ease of storing and retrieving the related design information to and from the Core Variant Design Feature Library. The design information required for the development of SG is extracted from the design features of each component of a particular type of products. In product design applications, most of the components have standard features to be designed for mass production while leaving some parts to be designed with variant form features. Based on this observation, the design features of each component can be classified into two groups: the standard and variant form feature groups. In the standard association tables, each variable of the design feature, which belongs to either standard or variant group, is assigned with default value.

4.4.3 Special Association Table

In the case of storing complex features of components into the Core Variant Feature Library, special association tables are used to describe the interrelationships among components. For example, from the geometric modelling point of view, complex design features of the components are tightly coupled together to perform specific functions, special association tables are used to store all the relevant information of SG, the control parameters and constraints of the design features as well as the interrelationships among the components. The interrelationships among components include the hierarchical relationships among components and the non-deterministic properties of shape evolution of the components.

In addition to the information stored in the standard association tables, special associated tables store extra information such as the conditions for the execution of SG rules, material, costing information, geometric properties, control variables, control parameters, constraints, special relations among shapes and all other relevant information. For example, a special association table (Table 5.1) shown in section 5.2.2, is designed to store the detailed information of the design features of digital cameras. The information to be stored for the Main Body include Control Variables (Body Type), Control Parameters (Classical or Swivel Types), Default Values (Classical) , Apply Rules (Rule 1 or 2), Update Values (labelled points ‘a’, ‘b’, ‘c’, and ‘d’ coordinates), Constraints (C9 which is the maximum boundary for the Main Body).

4.5 Parametric Shape Grammars

All related design information in constructing the Core Variant Model, such as geometric features, constraints and spatial relationships have been identified during the analysis of existing designs. The results obtained from the analysis are organised and represented in the Core Variant Design Feature Library. A set of parametric SG rules with labels can then be derived with reference to the library.

4.5.1 Classification of Shape Grammar Rules

In the product design domain, two types of functional activities are critical: the generation of components and the configuration of the generated components in an assembly. The SG rules can be classified based on these two design activities: Construction SG and Configuration SG from the functional points of views for product design domain. The Construction SG rules are used for the generation of each component whereas the Configuration SG rules are used for the allocation of components in an assembly. After all the components are generated by the

Construction SG rules, the components are configured in an assembly using the Configuration SG rules.

4.5.2 Construction Shape Grammar Rules

The Construction SG rules can be classified into different groups in accordance to the functional decomposition of the components or geometric properties. Labels are used to control the execution orders of the SG rules. The development of the Construction SG rules can further be classified into two types: 2D and 3D SG. The reasons for developing these two types of SG are that most of the components are standardised for ease of manufacture in industries. Some form features of the components can therefore be generated by standard mechanical methods such as extrusion and sweeping of 2D profiles. On the other hand, some form features require 3D free forms.

A set of parametric 2D SG rules with labels can be derived with reference to the Core Variant Design Model. The SG rules are derived to generate the 2D geometric profiles of components. The 2D profiles of components are chosen because they can be further manipulated with different methods such as coiling, extrusion, lofting, revolving and sweeping to create three-dimensional (3D) objects. For example, after a 2D profile is created by the application of the Construction SG rules, it is then extruded with a thickness to form a 3D object. For the generation of a curved 3D object, lofting operations between two profiles are required. Boolean operations on the 3D objects are also required to generate more complex geometric features of the components. In this way, the implementation time can be reduced in generating the form features of the components using 2D SG while longer implementation time is required for the generation of the free form exterior of main body using 3D SG.

In the case of developing parametric 3D SG, the SG rules are derived to generate the 3D geometric profiles of components. Parametric 3D SG rules with labels are developed for the generation of product forms which comprise common engineering shapes. The common engineering shapes are the vocabularies of SG. The common engineering shapes can be classified by their geometric properties like free-form shapes and primitive shapes such as blocks, cylinders, cones, spheres, torus, etc. and their combinations. Non-uniform rational B-spline (NURBS) surfaces are constructed to represent free-form objects in a virtual 3D spatial environment. Labels are used to associate with the control points of the NURBS surfaces and primitive shapes, and the design objects. The labelled control points are used for the identification of NURBS surfaces and primitive shapes. The labels of the design objects are used as function symbolic notations for the control of generation sequence. Both the labelled control points and the labels of design objects have valuables to specify their XYZ geometric coordinates.

4.5.3 Configuration Shape Grammar Rules

After all the components are generated by the Construction SG rules, the generated components will be configured in an assembly in accordance to the Configuration SG rules. The formulation of Configuration SG rules is based on the typical arrangement of components located at the main body. The Configuration SG rules use the boundaries of the designs as configuration constraints for the allocation of the components, and use labels to maintain proper generation sequence.

4.6 Evolutionary Architecture Integration

After going through all the above procedures of the systematic approach, two sets of SG can be obtained: the Construction and Configuration SG rules. These two sets of SG can generate designs which fulfil specific requirements. In order to automate some of the development processes in the systematic approach, evolutionary algorithms can be applied to evolve the SG for exploring designs to fulfil new design requirements. The development of an evolutionary architecture for an IGBDS involves several technical problems as specified in section 1.5.3 which are revisited here:

1) integration issues: One of the key issues in integrating a highly detailed SG and evolutionary computing system is that the random modification properties of evolutionary computing and the capturing of style properties of SG are in conflict. The random modification of product form designs removes the style of the product. More conflicts will occur if one combines different SG rules to derive new shapes,

2) representation issues: A new representation scheme should be developed in order to utilise the power of genetic and SG representations in encoding the shape features of the products, and

3) manipulation issues: A new adaptation mechanism which combines objective functions and control strategies should be developed in order to monitor the evolutionary process of the SG rules.

The first issue relates to the problems of random modification on stylistically consistent designs. Control strategies are developed to control the modification of shapes during the evolutionary process. These are capable of supporting adjustment on control parameters for styling modification.

The second issue relates to the representation for the integrated SG and evolutionary algorithm approach. A new representation is developed to utilise the power of genetic and parametric 2D and 3D SG representations with labels. In this thesis, the new representation named "GP-GA-SG" has been developed in the second prototype system. The second prototype system focuses on the development of parametric 3D shape grammars with labels enhanced by evolutionary computing with control mechanisms for supporting new product form designs.

With such an evolutionary IGBDS, a set of existing products in a particular product design domain is first analysed to derive shape features in the form of SG rules using the systematic approach. The SG rules are created with 3D labelled shapes and represented by both SG and genetic representations. In constructing the evolutionary architecture, the phenotype representation has to be specified prior to defining genotype representation. The phenotypes describe all permissible solutions that can be generated by the system. The phenotypes define all modifiable elements of the SG rules which can be influenced by the evolutionary algorithm. The modifiable elements are the three SG components: 1) Vocabulary of Shapes, 2) Spatial Relations and 3) Rule Sequences which are required to be identified in determining the phenotypes. After the creation of phenotypes and their corresponding genotypes, an evolutionary algorithm can then be applied to evolve the genetic SG rules to generate new designs.

The third issue relates to the manipulation for the integrated SG and evolutionary algorithm approach. An adaptation mechanism is developed to monitor the evolutionary process by objective function and control strategies. Multi-objective functions are developed to cooperate with the control strategies in directing the evolving SG rules to obtain specific design characteristics. The SG rules are evolved to generate designs in fulfilling new requirements and constraints. With such an evolutionary IGBDS, both product component designs as well as product configurations are supported.

The solutions to these technical problems related to the above issues in the exploration of designs using the integrated SG and evolutionary algorithm approach will be described in chapters 5 and 6.

4.7 Summary

This chapter has described and discussed the systematic approach to support the development process of SG. The details of implementing the key procedures of the systematic approach will be described in the coming two chapters in which two prototype systems are presented with design examples. The main points of this chapter are as follows:

- The systematic approach to support the development process of SG has been described. The key procedures of the systematic approach in deriving the SG rules and constructing an evolutionary architecture have been described. The completed systematic approach is then applied for the development of the two prototype systems which will be described in the next two chapters (Chapter 5 and 6).

- The systematic approach provides a unique ability to examine the impact of information technologies and knowledge representation techniques that can model the complex interrelationships among shapes and all related attributes for the development of SG. The systematic approach also aids the SG developer and designers in organising and classifying all relevant SG information that will enhance effective development of SG.

- The construction of an information network of SG has been described. The systematic approach explores the modelling of complex interrelationships among shapes and all related attributes as an information network. This perspective focuses on how to organise and classify all the relevant design information for deriving SG which can compute certain kinds of designs more easily or expressively than with a standard SG approach. The information network is parameterised and instantiated by examining how the existing designs and evaluation criteria can be classified from different perspectives. Two key issues are addressed in the construction of an information network of SG: 1) Identification of product design characteristics and 2) Redefine specific SG.

- The construction of a Core Variant Model has been described. The Core Variant Model is created by organising and extracting the usable information from the information network of SG. The Core Variant Model specifies the flexible structure of a class of products in which the interrelationships among components are controlled by the SG. An important part of the Core Variant Model is the construction of the Core Variant Design Feature Library which stores all the relevant information of SG in a well organised manner.

- Two types of association tables: standard and special types which store design information from different levels of complexity have been described. The association tables help the designers and the evolutionary IGBDS in systematically retrieving the design information from the Core Variant Design Feature Library. The standard association tables store the variables and parameters of the standard and variant design features. Whereas the special association tables store complex features of components which include all the relevant information of SG, the control parameters and constraints of the design features as well as the interrelationships among the components.

- The development process of parametric SG has been described. Parametric shape grammars can be derived in accordance to the interrelationships among components specified in the Core Variant Model. In the product design domain, the generation of components and the configuration of the generated components in an assembly are critical. The SG rules can then be classified into two main catalogues: Construction SG and Configuration SG from the functional points of views for product design domain accordingly.

- The construction of an evolutionary architecture has been described. In the evolutionary architecture, evolutionary algorithms are used as adaptation mechanisms in the exploitation of the generative capability of SG rules. The technical problems in representation and manipulation of integrating a highly detailed SG and evolutionary computing have been discussed. For representation issues, a new representation named "GP-GA-SG" is developed to utilise the power of genetic and parametric 2D and 3D SG representations with labels. For manipulation issues, control strategies are developed to control the modification on shapes during the evolutionary process. Multi-objective functions are developed to cooperate with the control strategies in directing the evolving SG rules to obtain specific design characteristics. The SG rules are evolved to generate designs in fulfilling new requirements and constraints. With such an evolutionary IGBDS, both product component designs as well as product configurations are supported.

In summary, a theoretical framework for integrating shape grammars with evolutionary computing techniques is developed in this thesis, which provides a basis for analysing existing designs to derive parametric 2D and 3D shape grammars, representing and organizing these shape grammars as knowledge for generating new designs, controlling the evolutionary computing process to generate new shape grammar rules that produce alternative designs. Including 2D and 3D parameters in the shape grammar rules based on a systematic approach to analysing and organizing the existing shape information from a real product domain is the key aspect of this theoretical framework, for which the implementation and evaluations involve a wide range of issues in computational design. These will be explained in detail in the context of a design example in the next two chapters (Chapter 5 and 6), which present two prototype systems for the validation of this framework.

Part IV Implementation and Analysis

Part four (Implementation and Analysis) presents a complete development process of the two prototype systems using the systematic approach. This includes the theoretical development, implementation and analysis of the integrated SG and evolutionary algorithm framework, and consists of two chapters.

- Chapter 5 (Parametric 2D Shape Grammars) describes the first prototype evolutionary IGBDS which employs genetic algorithm (GA) as the core evolutionary architecture.

- Chapter 6 (Parametric 3D Shape Grammars) describes the second prototype evolutionary IGBDS which employs genetic programming (GP) as the core evolutionary architecture.

Chapter 5

Parametric 2D Shape Grammars

5.1 Introduction

This chapter (Parametric 2D Shape Grammars) describes the first prototype of evolutionary IGBDS which employs genetic algorithm (GA) as the core evolutionary architecture. With this system, the process of deriving new rules is demonstrated by merging new shapes into the existing SG rules. This chapter consists of four main sections:

- In section 5.2, the systematic approach to supporting the development process of a parametric 2D SG for digital camera form design is described.

- In section 5.3, the construction of an evolutionary architecture used for integration to the IGBDS is described.

- In section 5.4, the system development of the first prototype evolutionary IGBDS is described.

- In section 5.5, the implementation results of the first prototype evolutionary IGBDS for digital camera form designs are presented.

5.2 Digital Camera Form Design with Shape Grammars

In developing the parametric 2D SG for product design applications with the systematic approach, there are two key tasks involved. The first task is to derive a set of SG rules using the systematic approach. The second task is to apply the evolutionary algorithm to evolve the derived SG rules which can generate new designs. In order to illustrate how the systematic approach can be applied to complete these two tasks, a particular type of product design applications is chosen as the first prototype system with the requirements as follows:

1) The chosen products are common types of industrial products.
2) The chosen products have a set of components, with each component having spatial relationships among other components.
3) Each component is designed under various constraints.
4) Some of the components have standard features for mass production and standardisation.
5) Some of the components have free form features for ergonomic or aesthetic design requirements.

The chosen product design application should have the above design characteristics such that parametric SG can be developed with the systematic approach and applied to real design applications. Also, the chosen product design application should be complex enough to demonstrate that the systematic approach is capable of solving the complex problems that occur in the development process of SG. The digital camera form design application is chosen here as the first prototype design application.

The reason for choosing the digital camera form design application is that it fulfils all the above mentioned requirements. Digital camera is a common type of industrial products composed of typical components, with each component having spatial relationships among the other components. Each component of the digital camera is designed under various constraints. Some of the digital camera components are standardised and have standard features such as press buttons, whereas some components of the digital camera have free form features such as the exterior of the main body.

5.2.1 Information Network of Shape Grammars

The first step of the systematic approach is to set up an information network of SG. In the first prototype system, an information network of SG is set up based on the identification of essential and identifiable design characters of the existing digital camera designs.

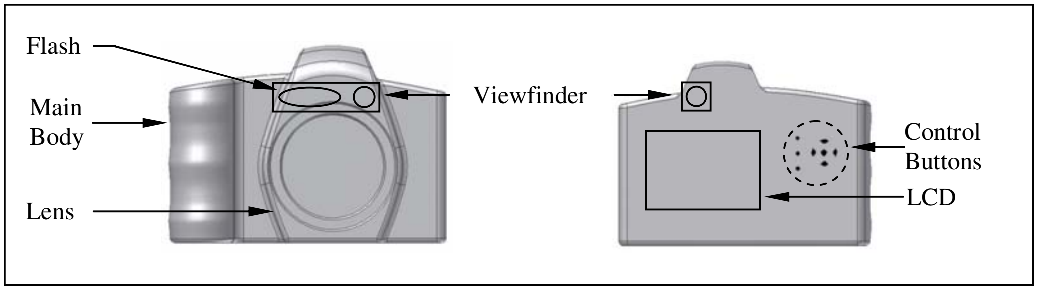


Figure 5.1 Typical arrangement of components configured at the main body

Identification of Product Design Characteristics

The form features of digital cameras are extracted from several different brands. The combination of form features representing different components is categorised according to their geometric locations in the assembly. The main components considered in the first prototype system include the Main Body, Lens, Flash and Liquid Crystal Display (LCD). A typical arrangement of components configured on the Main Body is shown in figure 5.1.

5.2.2 Core Variant Model

Development of Core Variant Design Feature Library

Since the information contained in the information network of SG can be enormous, the processing time required is too long. A Core Variant Model is used to abstract and extract all the relevant information contained in the information network of SG for specific design applications. For example, the geometric features of the digital camera components, which have interrelationships among other features, are hierarchically represented as objects in the Core Variant Design Feature Library. The library consists of different types of association tables which describe all these objects. Table 5.1 shows an example of the associated table used for the design of some particular types of digital cameras. With this arrangement of information specified in the associated table, the Core Variant Model can allow easy access to the information contained in the library. A detailed description of the information specified in the associated table is as follows:

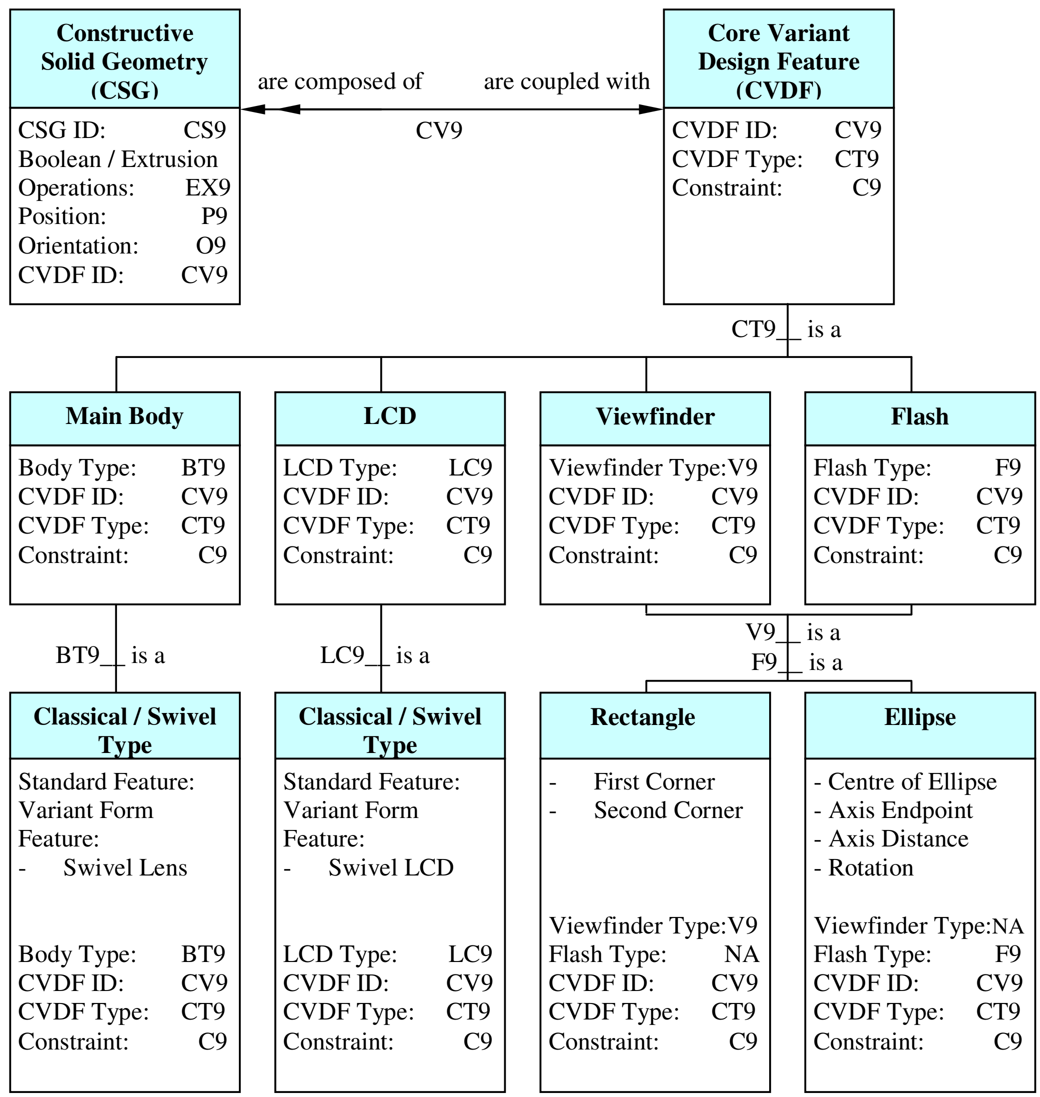


Table 5.1 Association table for the interrelationships among digital camera components

The Core Variant Design Feature (CVDF) object (with CVDF ID: CV9) is the parent object of four children objects: Main Body, LCD, Viewfinder and Flash objects. Both Main Body and LCD objects have one child object which belongs to either Classical or Swivel type object. In the case that the Classical Body or Classical LCD object is chosen, the Swivel Lens or Swivel LCD feature is absent. Both Viewfinder and Flash objects have only one child object which belongs to either Rectangle or Ellipse type object. Since the CVDF is integrated with Constructive Solid Geometry (CSG), the CVDF composed of CSG solids are created from a set of Boolean operations of other solid objects or through the extrusion of 2D profiles.

5.2.3 Parametric 2D Shape Grammars of Digital Camera Forms

All the relevant information in constructing the Core Variant Model such as geometric features, constraints and spatial relationships have been identified and abstracted from the information network of SG during the analysis process of the existing designs. The results obtained from the analysis are organised and represented in the Core Variant Design Feature Library. A set of parametric 2D SG rules can then be derived with reference to the library, especially with all the parametric variables shown in Table 5.1 (different variable names can be used instead of those in the SG). The SG rules are derived to generate the 2D geometric profiles of components. The 2D profile is chosen because it can be further manipulated with different methods such as coiling, extrusion, lofting, revolving and sweeping to create three-dimensional (3D) features. In this application, once a 2D profile is created by the application of SG rules, it is then extruded with a thickness to form a 3D object. Boolean operations on the 3D geometric features are required to generate more complex geometric features such as the Lens part.

Two types of rules are established in this application, one is for the generation of each component and the other for the configuration of the components in an assembly. After all the components are generated by the Component SG rules, the components are configured in an assembly using the Configuration SG rule as shown in Table 5.9. There are forty-five Component SG rules and one Configuration SG rule, only the main Component and Configuration rules are implemented in order to test the first prototype system. The system development and implementation details of the first prototype system are presented in section 5.4 and 5.5 respectively.

5.2.4 Construction Shape Grammar Rules

The first set of Component rules generates the profile of a digital camera (Table 5.2). Rule 1 starts with a rectangular shape labelled ‘I’ with the constraint points ‘a, b, c, d’. These constraint points limit the maximum boundary of the main body. The rule changes the label of the shape from ‘I’ to ‘CD’ for the classical design. Rule 2 generates a rectangular shape labelled ‘SD’ with a slot for the swivel lens design. The label ‘SX’ is used to match the swivel device created by rule 6.

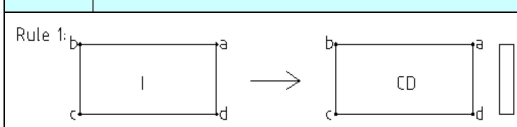
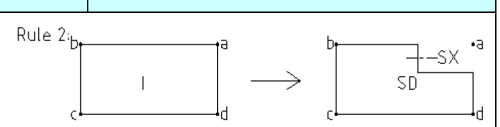
Item	Shape Grammar Rules	Item	Shape Grammar Rules
			

Table 5.2 SG rules for the exterior of the main body

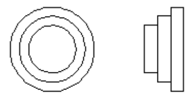
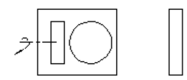
Item	Shape Grammar Rules	Item	Shape Grammar Rules
Rule 3:	L → 	Rule 4:	L → 
Rule 5:	L → 	Rule 6:	SX → 

Table 5.3 SG rules for the zoom lens and swivel device

The second set of Component rules is for the design of lens and swivel device (Table 5.3). For the classical design, rules 3 to 5 produce circular shapes for optical zoom lenses with zoom ranges: 1X (fixed), 2X and 3X respectively. For the swivel type design, a swivel device is attached to the slot of the main body. Rule 6 produces the swivel device in which the lens and flash are flush on the same surface. This allows the flash always to point to the object wherever the lens focuses. The label 'SX' is used for matching the label in rule 2 so that a swivel device can rotate on a swivel path of 300 degrees.

The third set of Component rules is for the generation of flashes and viewfinders. Rules 7 to 9 generate three types of flashes and rules 10 to 12 create three types of viewfinders (Table 5.4).

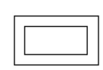
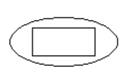
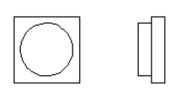
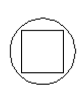
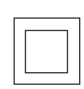
Item	Shape Grammar Rules	Item	Shape Grammar Rules
Rule 7:	F → 	Rule 8:	F → 
Rule 9:	F → 	Rule 10:	V → 
Rule 11:	V → 	Rule 12:	V → 

Table 5.4 SG rules for the flash and viewfinder

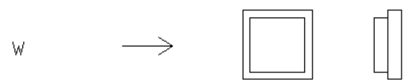
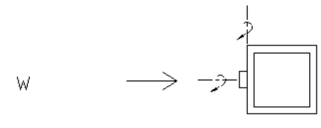
Item	Shape Grammar Rules	Item	Shape Grammar Rules
Rule 13:		Rule 14:	

Table 5.5 SG rules for the LCD screen

The fourth set of Component rules provides two types of LCD screens (Table 5.5). Rule 13 generates a rectangular shape for the LCD screen. Some LCD screens are added with swivel features. Rule 14 constructs a LCD screen with swivel features, which allow a screen to be moved independently of the camera body. With the swivel features, the designers can view the image on the LCD screen while shooting at the “over-the-head” or “low-angle” positions.

The fifth set of rules is for the generation of the multi-purpose control buttons (Table 5.6). Rules 15 to 19 produce the menu button designs with vertical patterns. Rule 15 starts with a label ‘B’ for the button design. The rule changes the label ‘B’ to ‘PB2’ for the two-button design. Rule 17 uses the label ‘PB2’ to generate two circular buttons. Rule 18 also uses the label ‘PB2’ to generate another style of two-button design. Rule 16 changes the label ‘B’ to ‘PB3’ for the three-button design. Rule 19 uses the label ‘PB3’ to generate the three-button design.

Rule 20 starts with a label ‘PB’ for the control button design. The rule generates a triangular shape with label ‘PB’. Rules 21 to 23 use these labelled triangular shapes as initial shapes to generate different button designs. The label ‘PB4’, ‘PB5’ and ‘PB6’ are added to the shapes on the right-side of the rules to distinguish the four-, five-, and six-button designs.

Rules 24 to 29 detail the four-button designs with different styles whereas rules 30 and 31 detail the five-button designs with circular patterns. Rules 32 and 33 detail the six-button designs with butterfly like style.

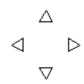
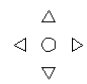
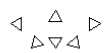
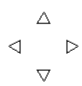
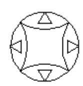
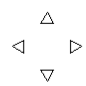
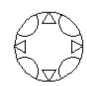
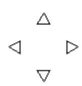
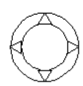
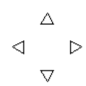
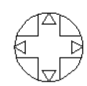
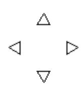
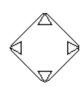
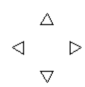
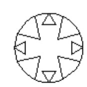
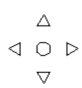
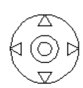
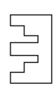
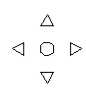
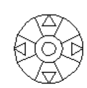
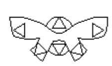
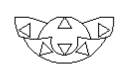
Item	Shape Grammar Rules	Item	Shape Grammar Rules
Rule 15:	B → PB2	Rule 16:	B → PB3
Rule 17:	PB2 → 	Rule 18:	PB2 → 
Rule 19:	PB3 → 	Rule 20:	PB → △ PB
Rule 21:	△ PB →  PB4	Rule 22:	△ PB →  PB5
Rule 23:	△ PB →  PB6		
Rule 24:	 PB4 → 	Rule 25:	 PB4 → 
Rule 26:	 PB4 → 	Rule 27:	 PB4 → 
Rule 28:	 PB4 → 	Rule 29:	 PB4 → 
Rule 30:	 PB5 →  	Rule 31:	 PB5 → 
Rule 32:	 PB6 → 	Rule 33:	 PB6 → 

Table 5.6 SG rules for the multi-purpose control button

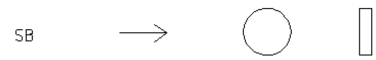
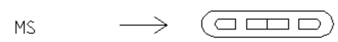
Item	Shape Grammar Rules	Item	Shape Grammar Rules
Rule 34:		Rule 35:	
Rule 36:			

Table 5.7 SG rules for the shutter and setting buttons

The sixth set of rules is for the generation of the shutter and setting buttons (Table 5.7). Rules 34 designs the shutter button which is of circular shape with size close to the nail of forefinger. It is assumed that the label 'SB' is located at the top right position of the digital camera. The designers can easily locate the shutter button by their forefingers. Therefore, there is no explicit rule to specify the location of the shutter button. Rule 35 uses the label 'MS' to generate a 'mode switch' setting button with an indication lamp. Rule 36 generates another style of the setting button.

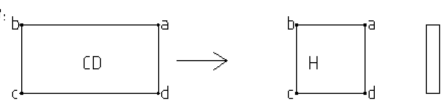
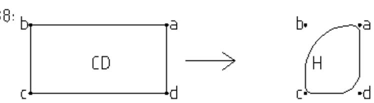
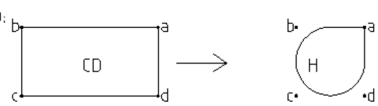
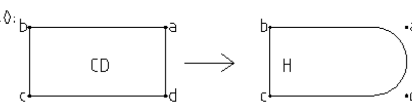
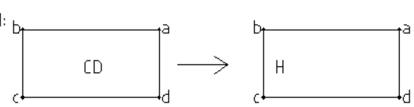
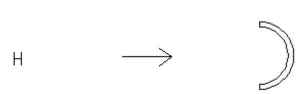
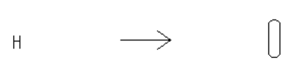
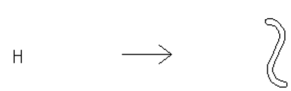
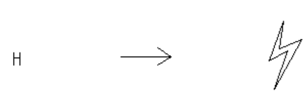
Item	Shape Grammar Rules	Item	Shape Grammar Rules
Rule 37:		Rule 38:	
Rule 39:		Rule 40:	
Rule 41:		Rule 42:	
Rule 43:		Rule 44:	
Rule 45:			

Table 5.8 SG rules for the exterior form styles of the main body and decorative features

The final set of Component rules generates the form styles of the Main Body and decorative features (Table 5.8). An elegant appearance gives a good impression to designers and attracts the designers' attention. Rules 37 to 41 generate the appearance of the Main Body based on the rectangular shape generated by rule 1. For the swivel lens design, it is assumed that the shapes of the main body and the swivel lens, generated by rule 2 and 6, are combined to form a rectangular shape. Also, the label 'SD' is used instead of 'CD' for the swivel lens design.

Rules 42 to 45 generate a variety of decorative features. The label 'H' is used to match the form created by rules 37 to 41. The decorative feature of appearance can also facilitate handling.

5.2.5 Configuration Shape Grammar Rules

After all the components are generated by the Component SG rules, a Configuration SG rule will be applied to specify the configuration of the generated components (Table 5.9). The formulation of Configuration SG rules is based on the typical arrangement of components located on the Main Body.

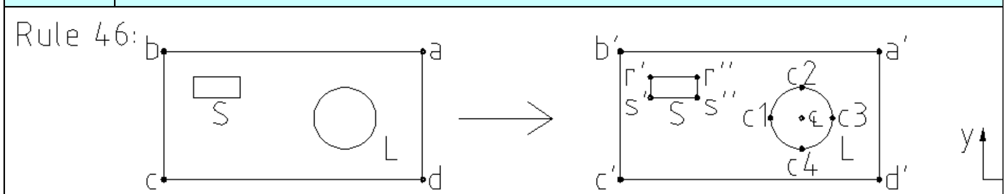
Item	Shape Grammar Rules
Rule 46:	

Table 5.9 The configuration SG rule

Relations	Main Body to Lens	Main Body to 'Flash and Viewfinder' unit	Flash to Lens
Constraints	$xb' = xc' < xc_1$, $xa' = xd' > xc_3$, $yb' = ya' > yc_2$, $yc' = yd' < yc_4$.	$xb' = xc' < xr' = xs'$, $xa' = xd' > xr'' = xs''$, $yb' = ya' > yr' = yr''$, $yc' = yd' < ys' = ys''$.	$ys'' > yc_4$.

Table 5.10 The configuration constraints

The Configuration SG rule 46 uses the boundaries: 'a, b, c, d' as configuration constraints for the allocation of the lens (L) and the combined "flash and viewfinder" unit (S) in the front view of the Main Body. The same Configuration SG rule is applied to the components configured on the back view of the digital camera. The combined "LCD screen and Viewfinder" unit (S) and the Button Unit (L) are allocated at specified locations on the back view of the Main Body. Table 5.10 lists the constraints for the configuration of the components.

5.3 Evolutionary Architecture Integration

This section describes the evolutionary architecture used for the integration of the first IGBDS. In this section, the methodology used to analyse and identify the modifiable elements of a specific SG based on the design characteristics of the digital camera is described first. These modifiable elements are: 1) Vocabulary of Shapes, 2) Spatial Relations and 3) Rule Sequences which are the phenotypes of the system. Second, based on the analysis results, the methodology used to derive new SG rules using the modifiable elements for new shape generation is described. Third, the methodology to control construction rule order sequences is described. Fourth, the manipulation of coded modifiable elements which are the genotypes is described. Fifth, the operations of the genetic algorithm (GA) which is the core of this evolutionary architecture are described. Lastly, the evaluation method of the generated designs using artificial selection is described.

5.3.1 Identification of Modifiable Elements (Phenotypes)

The phenotype representation describes all permissible solutions that can be generated by the system. Permissible solutions form the design space for evolutionary search. Since the goal in this first prototype system is to enhance the generative capability of SG rules by integrating an evolutionary algorithm to an IGBDS, the permissible solutions will be the refinement of specific SG rules or the creation of new SG rules. Therefore, the main aspect to be studied in determining the phenotype representation is to search what elements of a specific SG can be modified by the evolutionary process and what elements or shapes can be added to the existing SG rules for the creation of new SG rules. In other words, the phenotype representation enumerates the search spaces, defining all modifiable elements of the SG rules, which can be influenced by the evolutionary algorithm.

Prior to determining the genetic representation used in this first prototype system, identification of what elements of a specific SG can be modified by the evolutionary process is studied. The modifiable elements of the three SG components: 1)

Vocabulary of Shapes, 2) Spatial Relations, and 3) Rule Sequences have to be identified in the analysis.

A) The First SG Component: Vocabulary of Shapes

A subset of specific SG rules developed in the previous section is analysed to realise the design characteristics of the products. Component rules 37 to 41 which are used for constructing the form styles of the Main Body are selected for analysis. It is observed that an elegant form style provides a good impression to the designers. An elegant style can emerge from a combination of, or by introducing new features to, the existing designs. By observing that the shapes on the right hand side of the Component rules describe the form styles of the Main Body, the shapes specified in the right hand side of the Component rules are therefore identified as the modifiable elements of the phenotype.

B) The Second SG Component: Spatial Relations

When the spatial relations of the shapes on the right hand side of the Component rules are modified, the form styles of the Main Body are modified accordingly. The modification of the form styles of the Main Body can be simulated by a newly derived Component rule. A newly derived Component rule is constructed by copying the existing shapes on the right hand side of a Component rule to the left hand side of the newly derived Component rule, and introducing new shapes to the right hand side. The newly derived Component rule describes a new form style of the Main Body. Consequently, the spatial relations of the shapes on the right hand side of the Component rules are identified as the modifiable elements of the phenotype.

C) The Third SG Component: Rule Sequences

In determining suitable rule execution sequences for the generation of new configuration designs, it is necessary to study the effects arising from the replacement of the components and their corresponding positions in an assembly. The first step is to analyse how the components can be categorised into different groups for ease in the replacement process. By observation of the existing product designs, each product has a set of components which can be classified according to their functional characteristics or geometric properties. Since the components are generated by the Construction rules, the corresponding Construction rules can be grouped accordingly. The replacement of components for a new configuration design can be achieved by determination of suitable rule execution sequences, which specifies the execution order of appropriate Construction rules selected from different Construction rule groups. The corresponding positions of the components generated by the selected Construction rules have to be determined accordingly.

In general situations, each component is configured by the Configuration rules under spatial constraints to form an assembly. When the Configuration rule executes, one Construction rule from each Construction rule group will be selected to generate the component in advance before configuring that component in the assembly. The Construction rules are executed in accordance to specific rule order sequences. The modification of the rule execution sequences can therefore generate alternative configuration designs with suitable components and the configuration of these components in the assembly. As a result, the rule execution sequences of selecting Component SG rules for configuration design are considered to be the modifiable element of the phenotype.

D) Components of Phenotypes

After the identification of modifiable elements of the SG rules, the phenotypes can be defined. The phenotypes consist of three elements: 1) Vocabulary of Shapes, 2) Spatial Relations and 3) Rule Sequences (Table 5.11).

1. Vocabulary of Shapes	2. Spatial Relations	3. Rule Sequences

Table 5.11 The phenotype representation for the first prototype evolutionary IGBDS

5.3.2 Generating New Shape Grammar Rules

After defining the phenotypes in the previous section, this section focuses on the first two modifiable SG components: 1) Vocabulary of Shapes and 2) Spatial Relations of the phenotype which are critical in generating new SG rules. The phenotype representation consists of the encoded Vocabulary of Shapes and Spatial Relations. In this prototype system, the evolutionary algorithm manipulates the coded phenotype representation (Genotypes) in order to generate new designs for new form style of the Main Body.

Genetic Algorithm (GA) is chosen as the evolutionary algorithm for the first prototype system. The GA generates the new shapes for the right hand side of the new Construction rule. The working principle of how the GA generates new shapes is presented. This is an alternative approach to generate “emergent” shapes compared with the recursive application of SG rules. Knight (2003) has presented a detailed description of SG on how to compute with the emergent properties of shapes.

Item	1	2	3	4	5	6	7
Action	Remain unchanged	B + C	B – C	C – B	B ● C	Random generated	Designer specified

Table 5.12 Seven ways available for the GA to take action

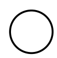
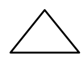
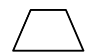
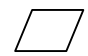
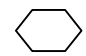
Item	C1	C2	C3	C4	C5
Shapes					

Table 5.13 Available shapes in a library

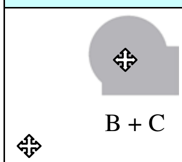
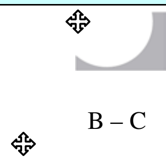
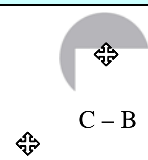
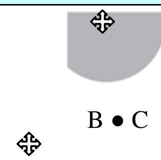
An Example of Algebra Operations to Generate Four New Shapes			
 B + C	 B – C	 C – B	 B ● C

Table 5.14 An example of algebra operation to generate four new shapes (Shape B and C should be located at the corresponding reference points for each operation.)

The Change of Form Style of Main Body by the GA

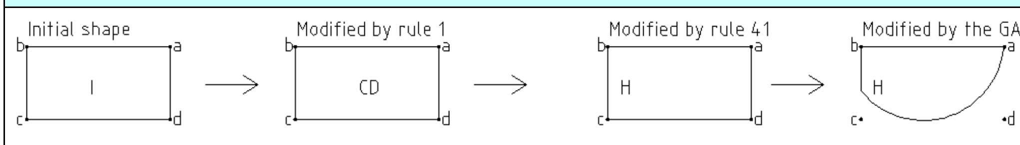


Table 5.15 The change of form style of main body by the GA

An Example of a New Rule is Derived

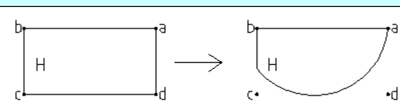


Table 5.16 An example of a new SG rule is derived

For each generation in the evolutionary process, the newly derived Construction rules are continuously modified with the old and new shapes on the left and right hand sides of the SG rules respectively; the rules are evolving. As a result, a new Construction rule can be chosen from each generation in the evolutionary process.

To generate new shapes for the SG rules, the first step is to modify the two modifiable elements of the phenotype: 1) Vocabulary of Shapes and 2) Spatial Relations which are specified in the right hand side of the Construction rules. There are seven ways available for the GA to take action (Table 5.12). The choice of action number 1 is to keep the current SG rule unchanged. Action numbers 2 to 5 specify a shape 'C' to be extracted from a library (Table 5.13). A label 'B' is first added to the shape on the right hand side of the SG rule. Then, the shape 'C' is added to shape 'B' by different algebra operations. Action number 6 specifies the extracted shape 'C' to be randomly generated by the computer. Action number 7 specifies the extracted shape 'C' to be provided by the designers. If action number 6 or 7 is selected, the shape 'C' replaces the shape 'B'. As a result, action numbers 2 to 7 introduce a new vocabulary of shapes to the SG.

Three types of shape algebra operations change the spatial relations of shapes, + (sum), - (difference) and ● (product). Chase (1996) has presented the operations of shape algebras and formal logic in design modelling. An example of adding a circular shape 'C' to the rectangular shape 'B' on the right hand side of the rule 41 is shown in table 5.14. The algebra operations have been tested to see how these operations might be utilised in the first prototype system. However, results obtained from the initial experiments do not satisfy all operations. Therefore, only the union operation is adopted for new rule construction in the implementation. An example of the change of form styles of the Main Body by the GA is used to illustrate the key operations (Table 5.15).

Table 5.16 shows an example of a newly derived Construction rule which is constructed by the following procedures:

- It is supposed that the Main Body form design is generated by the Construction rule 41.
- The existing shapes on the right side of Construction rule 41 are copied to the left and right side of the new Construction rule.
- A label 'B' is added to the right hand side of the new Construction rule.
- A circular shape with label 'C' is randomly extracted from the library by the system.

- The selected circular shape with label 'C' is united with the label shape 'B'. A new shape is therefore created by an algebra operation B ● C and placed to the right hand side of the newly derived Construction rule. After this simple algebraic operation, a new SG rule is created.

5.3.3 Construction Rule Order Sequences

In order to modify the rule execution sequences for alternative configuration design generation, the rule execution sequences of selecting appropriate Construction rules for configuration designs is represented by a list of the corresponding rule numbers and stored in an array for the GA manipulation. Table 5.17 shows different groups of Construction rules for a GA to determine the rule execution sequences.

Item	Type	Lens	Flash	View finder	LCD	Form style
Rule Number	Choose 1 to 2	Choose 3 to 6	Choose 7 to 9	Choose 10 to 12	Choose 13 to 14	Choose 37 to 45

Table 5.17 Construction rule order sequences

Item	Bit	Genotype - Binary digits													
		0	1	00	01	10	11	000	001	010	011	100	101	110	111
		Phenotype - Decimal digits													
1. Type	1	1	2												
2. Lens	2			3	4	5	6								
3. Flash	2			7	8	9	7								
4. View finder	2			10	11	12	13								
5. LCD	1	13	14												
6. Menu(M) button	1	15	16												
7. Two M. button	1	17	18												
8. Control(C) button	2			21	22	23	21								
9. Four C. button	3							24	25	26	27	28	29	24	25
10. Five C. button	1	30	31												
11. Six C. button	1	32	33												
12. Setting button	1	35	36												
13. Form style	3							37	38	39	40	41	37	38	39
14. Form features	2			42	43	44	45								
15. Rule	3							1	2	3	4	5	6	7	1
16. Value sign	1x8	-	+												
17. Configuration	3x8							0	5	10	15	20	25	30	35
Total bits	58														

Table 5.18 Genotype and phenotype representations for the first prototype system

5.3.4 Manipulation of Coded Modifiable Elements (Genotypes)

The genotypes are the coded modifiable elements of a specific SG; they are the coded phenotypes. The genotype is represented by a single chromosome. A chromosome is a list of alleles which is the binary encoding of the phenotypes. Table 5.18 presents a sample of the genotype representation. The main components: Item 1, 2, 5, 13 and 15 are implemented in the first prototype system.

5.3.5 Genetic Algorithm

The GA performs three main functions: modifying alleles within chromosomes using genetic operators, decoding the genotype to produce the phenotype, and evaluating the phenotype to identify the fittest solutions (Figure 5.2).

A GA generates an initial population of individuals with random values. The main loop begins at this stage. Each individual is then evaluated and assigned a fitness value by a fitness function. Based on the score obtained from each solution, the solution with higher score will be selectively copied to a temporary area termed ‘mating pool’.

The second loop of the evolutionary cycle starts at this stage. Two of the solutions are randomly selected as parents from this ‘mating pool’. These two parents generate two offspring by random crossover and mutation operators. These two offspring replace the parents of the population. The crossover and mutation processes repeat to generate offspring until every parent of the old population is replaced, and a new population with fitter solutions is established (Holland, 1975).

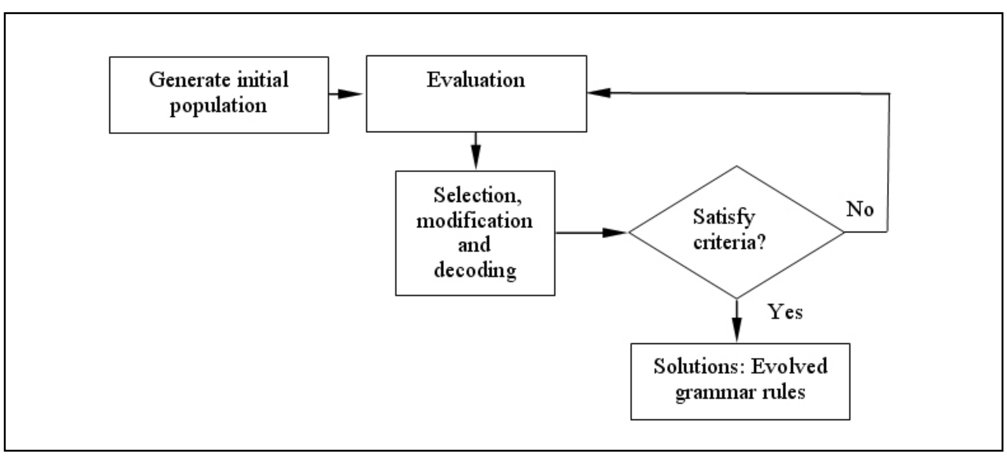


Figure 5.2 Operation of GA

For each generation, the genotype is converted into the phenotype which represents possible solutions. The solutions are a number of individuals, each of which consists of a set of rule numbers and the instructions to create new SG rules. After execution of the SG rules in accordance to the generated rule sequences, the components are generated. The GA repeats the main loop of evaluation and reproduction processes for a specified number of generations, or the GA will stop automatically if a satisfactory solution emerges.

5.3.6 Evaluation Method

In the first prototype system, artificial selection is adopted as the evaluation criterion. For each generation, the system generates 20 designs. The designers can evaluate the designs generated by the evolved SG rules in accordance to the designers' preferences. This is achieved by visualising the designs and subjectively selecting the best one by the designers in each generation. The selected design is graded with the highest score. Those grammar rules which can generate the designs with higher scores will have higher survival chance in the next generation. In the implementation, the maximum number of generations is set to 1000. Designers can halt the generation progress at any time as long as the designers are satisfied with the designs.

5.4 System Development

This section describes the system development of the first prototype evolutionary IGBDS. Generative and evolutionary design techniques are developed based on the inspiration from natural evolution. Centred on the techniques are the generative methods and the testing of the designs generated by these methods (Simon, 1969). Four main types of evolutionary algorithms were developed: Evolutionary Strategies (Rechenberg, 1973), Evolutionary Programming (Fogel, 1963), Genetic Algorithms (GA) (Holland, 1975) and Genetic Programming (Koza, 1992). For all these approaches, the GA is the most widely used evolutionary algorithm in many application domains. In this first prototype evolutionary IGBDS, a classical GA is used as the core of the evolutionary architecture for evolving new SG rules (Figure 5.3). The evolved SG rules are used to generate new designs.

5.4.1 System Architecture

Figure 5.3 illustrates the evolutionary IGBDS framework in which the SG and evolutionary algorithm are utilised to support the interactive design process and design activities. The integrated SG and evolutionary algorithm approach consists of two key elements: 1) SG as the knowledge for design and 2) Evolutionary computing as the generative mechanism. During the interactive design process, the designers can interact with the evolutionary IGBDS by inputting the design criteria and evaluate the designs.

In this integrated SG and evolutionary algorithm paradigm, a set of SG rules is first developed using the systematic approach and then encoded as the 'code scripts' of a GA. The SG rules are used to generate a population of random or predefined solutions at the beginning of evolutionary design process. For each new generation, the GA decodes the 'code script' to produce a set of new SG rules. In addition, the system allows the designers to participate in the construction of the new SG rules. Consequently, more SG rules are generated and the designers can use the evolved new SG rules to explore alternative designs. The design of the forms of digital camera is used as an application example to illustrate the operations of the evolutionary IGBDS and to evaluate the effectiveness of the systematic approach.

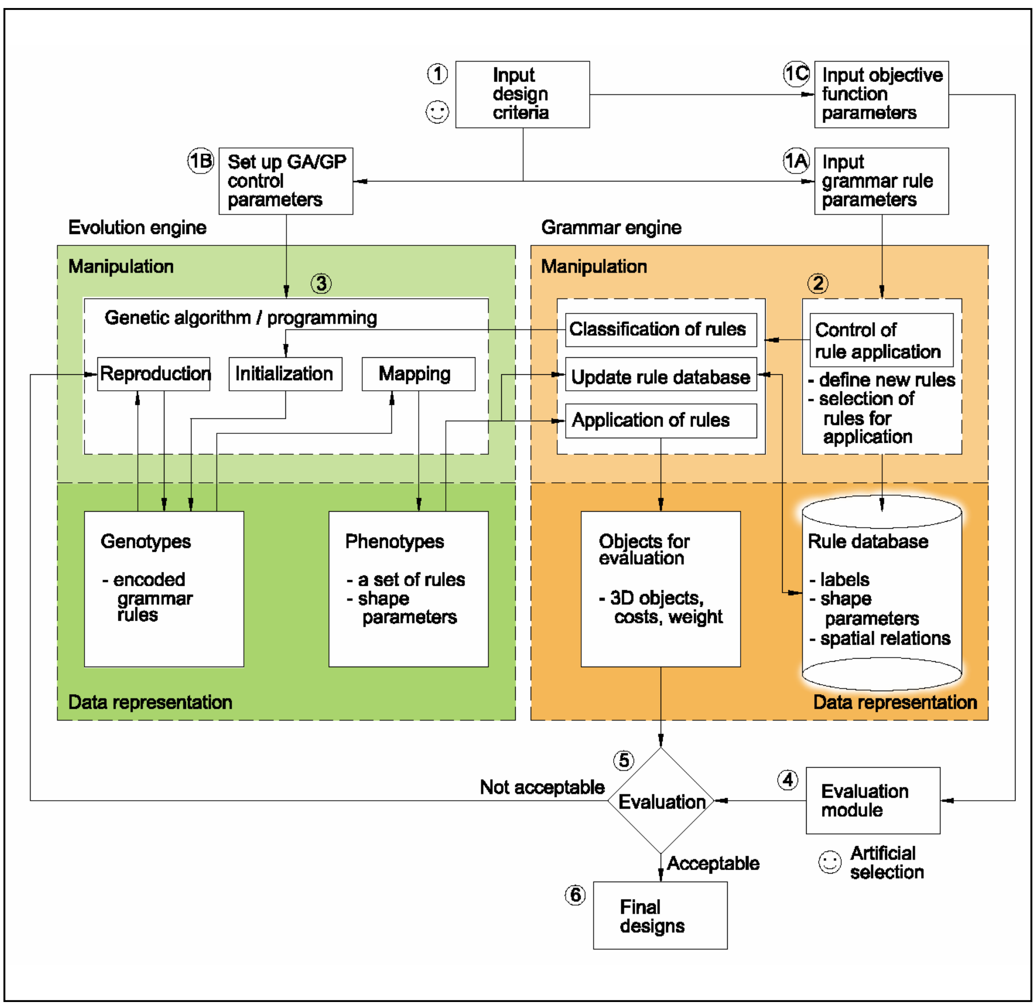


Figure 5.3 Interactive design process utilising shape grammars and evolutionary algorithms

5.4.2 System Operations

An evolutionary architecture has been developed and integrated with an IGBDS to evolve the SG rules (Lee and Tang, 2004). Parametric 2D SG rules with labels have been developed to generate 3D objects by extrusion of 2D labelled shapes. The 2D SG rules are composed of rule numbers, shapes and their corresponding parameters. Each SG rule is classified into three main groups: 1) Exterior of the product form generation, 2) Component generation, and 3) Configuration generation. The execution of each SG rule follows an ordered sequence which is determined by the GA. The control parameters of labelled shapes and rule numbers are encoded as code scripts of the GA. The GA evolves the SG rules to generate both existing and novel designs. New designs composed of new shapes and features are generated by Boolean operations among existing shapes and the shapes in the shape repository.

For each generation, the evolutionary IGBDS generates 20 designs. The generated shapes are evaluated artificially based on the designers' preferences by visual inspecting the designs generated in each generation. In some cases, more than one generation is required for genetic operations of crossover and mutation to achieve significant effects on the generated designs, before the designers evaluate the generated designs. For these cases, each periodically observed generation is usually set to 50 generations or larger. The selected design is graded with the highest score. Those grammar rules which can generate the designs with higher scores will have higher survival chance in the next generation. The designers can halt the generation progress at any time as long as the designers are satisfied with the designs.

5.5 Implementation Results

This section presents the implementation results of the first prototype system. The first prototype system has been developed as a stand-alone application using Visual Basic and linked in the OpenGL programming environment. Figure 5.4 and 5.5 show the first and second user interfaces of the first prototype system.

5.5.1 Application of IGBDS

At the beginning, designers can explore designs individually using the SG based design approach without the integration of the evolutionary architecture (Figure 5.4). The system provides different sets of component rules for the designers to select. The designers can specify the type of each component and its corresponding parameters.

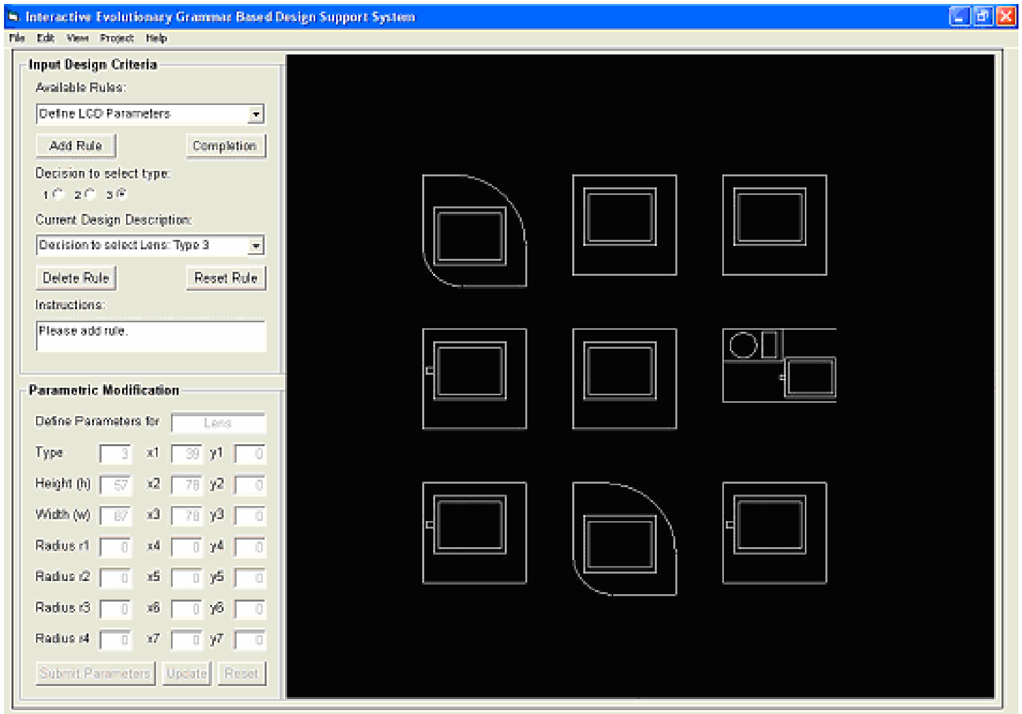


Figure 5.4 The first user interface of the first prototype evolutionary IGBDS

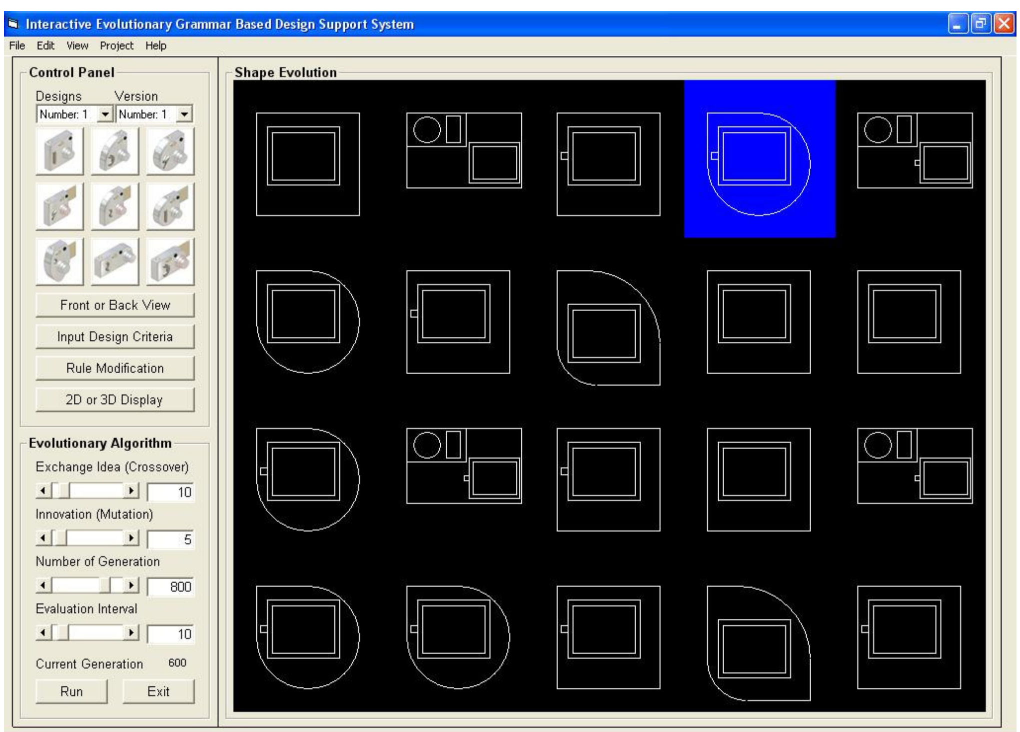


Figure 5.5 The second user interface of the first prototype evolutionary IGBDS

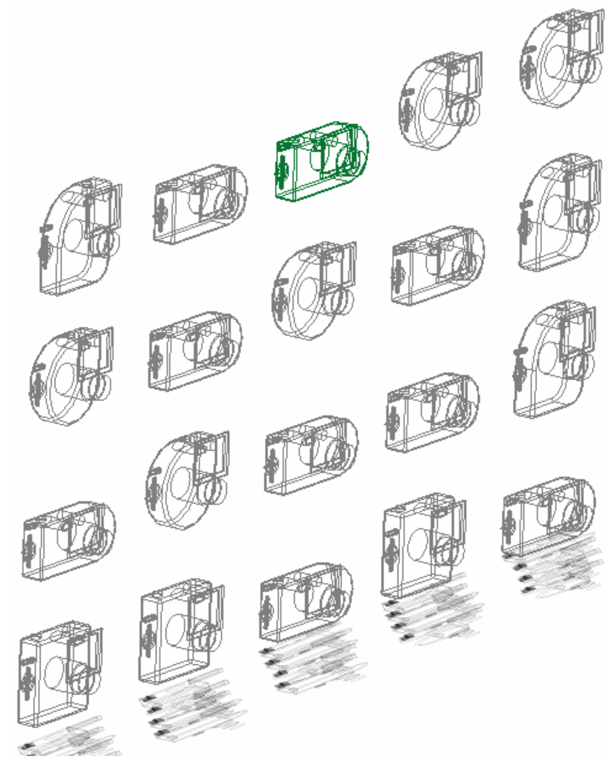


Figure 5.6 The GA has generated the population of the results at 200 generations

5.5.2 Application of IGBDS with Evolutionary Process

The designers can then explore the designs using the SG based design approach with the integration of the evolutionary architecture (Figure 5.5). The designers can input the design criteria such as the number of generations, crossover and mutation rate. After over two hundred generations, the GA has generated the population of the results as shown in figure 5.6. The main components such as Main Body, Lens, Swivel Lens and Flash device, Liquid Crystal Display (LCD) and Swivel LCD device are generated by the system whereas the decorative features, buttons and view finders are added manually for illustration purpose only.

Figure 5.7 The two designs with higher scores after over 300 generations

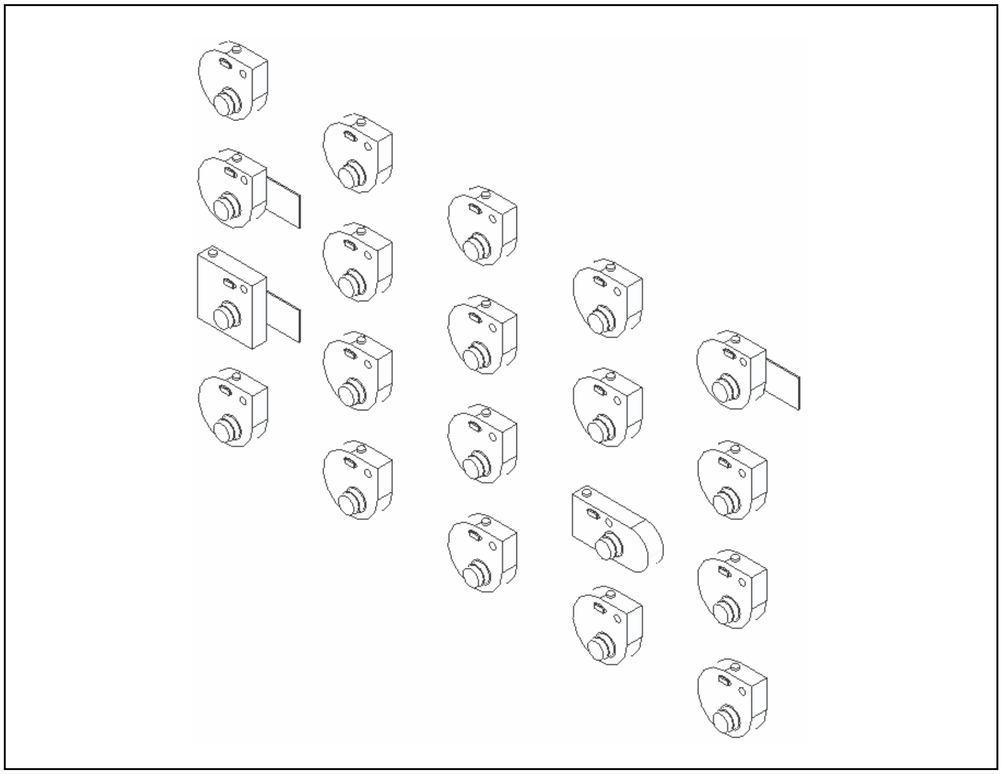


Figure 5.8 Solutions are converged after over 300 generations

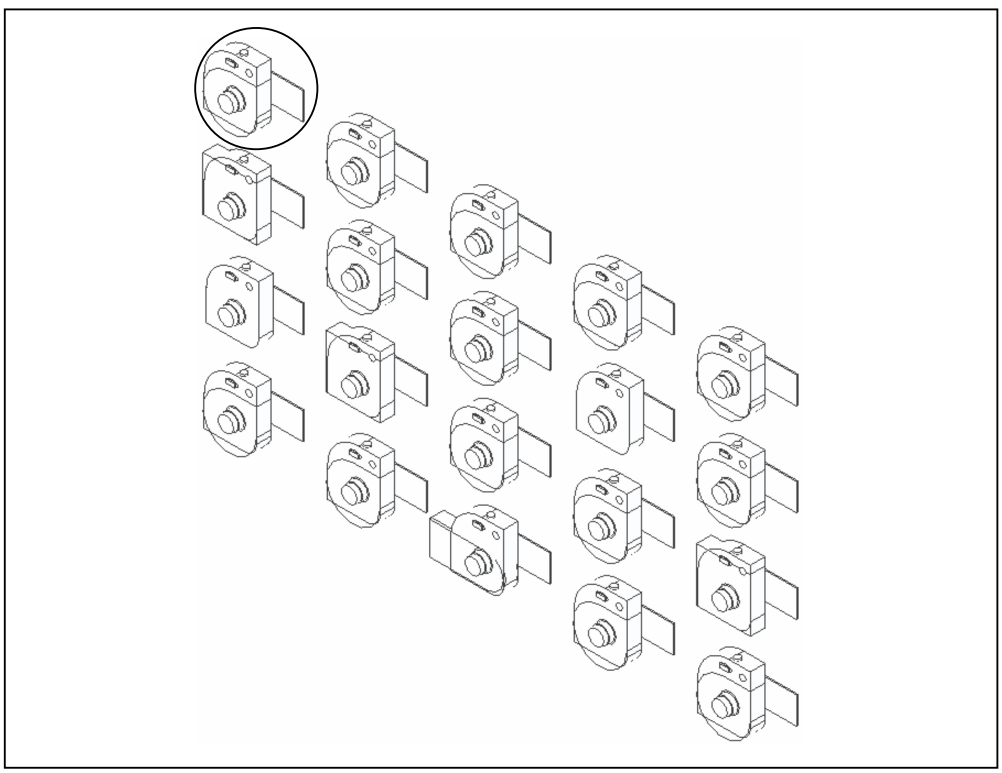


Figure 5.9 New design generation

These designs are graded by the designers intuitively. The system uses the SG rules that produced higher scores to generate the designs for the next generation. The whole design process repeats until the designers satisfy the results. The two designs with the highest scores after over 300 generations are shown in figure 5.7. These two designs become the major population in the next generations and the solutions converge (Figure 5.8).

The system can generate the designs by evolving the existing SG rules. Alternatively, the system can introduce new shapes to the existing SG rules in order to generate the new designs (Figure 5.9).

The results obtained from this first prototype system are not satisfactory for all the algebraic operations. In this implementation, only the union algebraic operation is applied to derive new rules. Therefore, another approach to generate the free form objects is developed and shown in the next chapter for the second prototype system.

5.5.3 Examples of Deriving New Rules

The procedures to derive a new SG rule for the generation of the exterior main body with circular mark shown in the figure 5.9 are explained. A newly derived Construction rule is constructed by the following procedures:

- The Main Body form design is generated by the rule 38 (Table 5.8).

- The existing shapes on the right hand side of the rule 38 are copied to the left and right sides of the newly derived Construction rule.

- A label 'B' is added on the right hand side of the newly derived Construction rule. A circular shape with label 'C' is randomly extracted from the library by the system (Table 5.14).

- The selected circular shape with label 'C' is united to the label shape 'B'. A new shape is therefore created by an algebra operation B + C and placed to the right hand side of the newly derived Construction rule. After this simple algebraic operation, a new SG rule is created (Figure 5.10). The procedures to derive this new rule are similar to the illustration shown in Table 5.16.

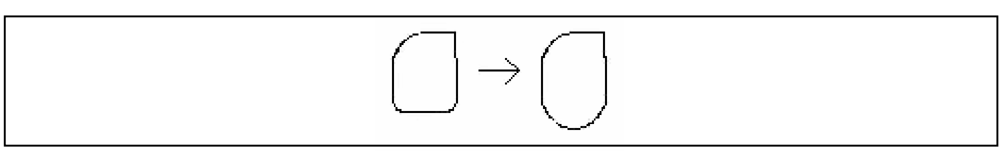


Figure 5.10 A new SG rule is derived for the chosen design

5.6 Summary

This chapter has described the systematic approach to enhancing the generative capability of SG rules by integration of an evolutionary architecture to an IGBDS. The key points are as follows:

- The system development of the first prototype system has been described. Especially, the key features which include the relevant technological details of the implementation for the system. This evolutionary IGBDS is capable of generating existing designs as well as alternative complex designs. This is achieved by integrating an evolutionary algorithm to an IGBDS to evolve new SG rules. The new SG rules are derived based on artificial selection and used for the exploration of new designs.

- An evolutionary architecture using GA has been developed and implemented to integrate with an IGBDS. The evolutionary architecture for the first prototype system consists of phenotype and genotype representations of SG. The genotypes are manipulated by selection process and genetic operators such as crossover and mutation. During the evolutionary process, a population of solutions are evolving and becoming better in each generation. These solutions refer to large numbers of alternative design concepts, forms, embodiments, structures and assemblies which are automatically generated in a controlled and supervised manner.

- A critical step in constructing the evolutionary architecture is to analyse and identify the modifiable elements of a specific SG based on the design characteristics of the specific type of products, for example, the digital camera SG. These identified modifiable elements of a specific SG are encoded into the genetic representation (genotypes and phenotypes) of the evolutionary algorithm in order to derive new SG rules. The genotypes can be manipulated by the GA to derive new SG rules for new design generation. Another important issue in constructing the evolutionary architecture is to control the rule order sequences such that the alternative configuration designs can be generated.

- Examples of digital camera form designs are used for demonstration of the systematic approach in deriving new SG rules. In the first prototype system, there are two modes of interactions between the designers and the evolutionary IGBDS: with and without evolutionary processes involved. The designers can compare the two modes of interaction in using the system. The application of the derived new SG rules in generating new designs is also provided in the demonstration.

- This chapter has demonstrated the operation process of the first prototype system. These demonstrations together with the implementation results empirically verify two key parts of the theoretical propositions of the systematic approach: 1) the development of SG and 2) the extension of the generative capability of the SG by the integration of an evolutionary architecture to an IGBDS.

In summary, the first prototype presented in this chapter has demonstrated the capability of SG integrated with IES. The limitations of this prototype will be analysed in the next chapter (Chapter 6) in which a new and improved system will be described and evaluated.

Chapter 6

Parametric 3D Shape Grammars

6.1 Introduction

This chapter (Parametric 3D SG) describes the second prototype evolutionary IGBDS which employs genetic programming (GP) as the core evolutionary architecture. The second prototype consists of parametric 3D SG with labels which can be enhanced by evolutionary computing with control mechanisms for supporting new product form designs. New key features have been developed in the second prototype system for technological advancement in deriving SG. The new key features are developed and targeted to enhance the performance of the three most critical elements of the system: 1) Representation, 2) Manipulation, and 3) Evaluation which are critical to the development process of SG and evolutionary computing.

This chapter consists of five main sections:

- In section 6.2, the systematic approach for supporting the development process of a parametric 3D SG for digital camera form design is described.
- In section 6.3, the development of the evolutionary architecture for the second prototype is described.
- In section 6.4, the system development of the second prototype system is described.
- In section 6.5, the implementation results of the first experiment using the second prototype system for digital camera form design are presented.
- In section 6.6, the implementation results of the second experiment using the second prototype system for digital camera form design are presented.

6.2 Parametric 3D Shape Grammars

This section focuses on the derivation of parametric 3D SG with labels using the systematic approach. A set of existing products in a particular product design domain is first analysed to derive shape features in the form of SG rules. The SG rules are then created with 3D labelled shapes. An example of deriving the parametric 3D SG with labels for the exterior of the digital camera main body design is used to illustrate the systematic approach. The 2D SG is also included for the generation of other components. In the following sections, the limitations of the first prototype are first reviewed. It then follows with the development of parametric 2D and 3D SG for the generation of components and their configuration in an assembly.

6.2.1 Limitations of the First Prototype

The limitations of the first prototype system can be categorised by two main issues: the representation and manipulation issues. For the representation issues, the first prototype system employs parametric 2D SG with labels to generate 3D shapes. This limits the modelling capability of the system in generating free form designs, because the parametric 2D SG with labels lacks the flexibility to modify 3D free form objects. The generated 2.5D or 3D designs are therefore limited in variety. For the manipulation issues, evaluation of the generated designs is limited to artificial selections only. As a result, the complex effects produced by the SG rule modifications during the evolutionary process can not be fully explored and analysed.

The second prototype system is an extension to the first prototype which improves the performance of the computational framework. New key features including those stated in the introduction section of this chapter have been developed in the second prototype system. In particular, this section (section 6.2) presents one of the key features, which is the parametric 3D SG with labels for more powerful modelling capability in dealing with sophisticated free form generation.

6.2.2 Parametric 3D Shape Grammars for Free Form Design

Since all the procedures of the systematic approach such as the development of information network of SG and the Core Variant Model have been done in the first prototype system, this second prototype system can take the advantage to use such information without starting from scratch.

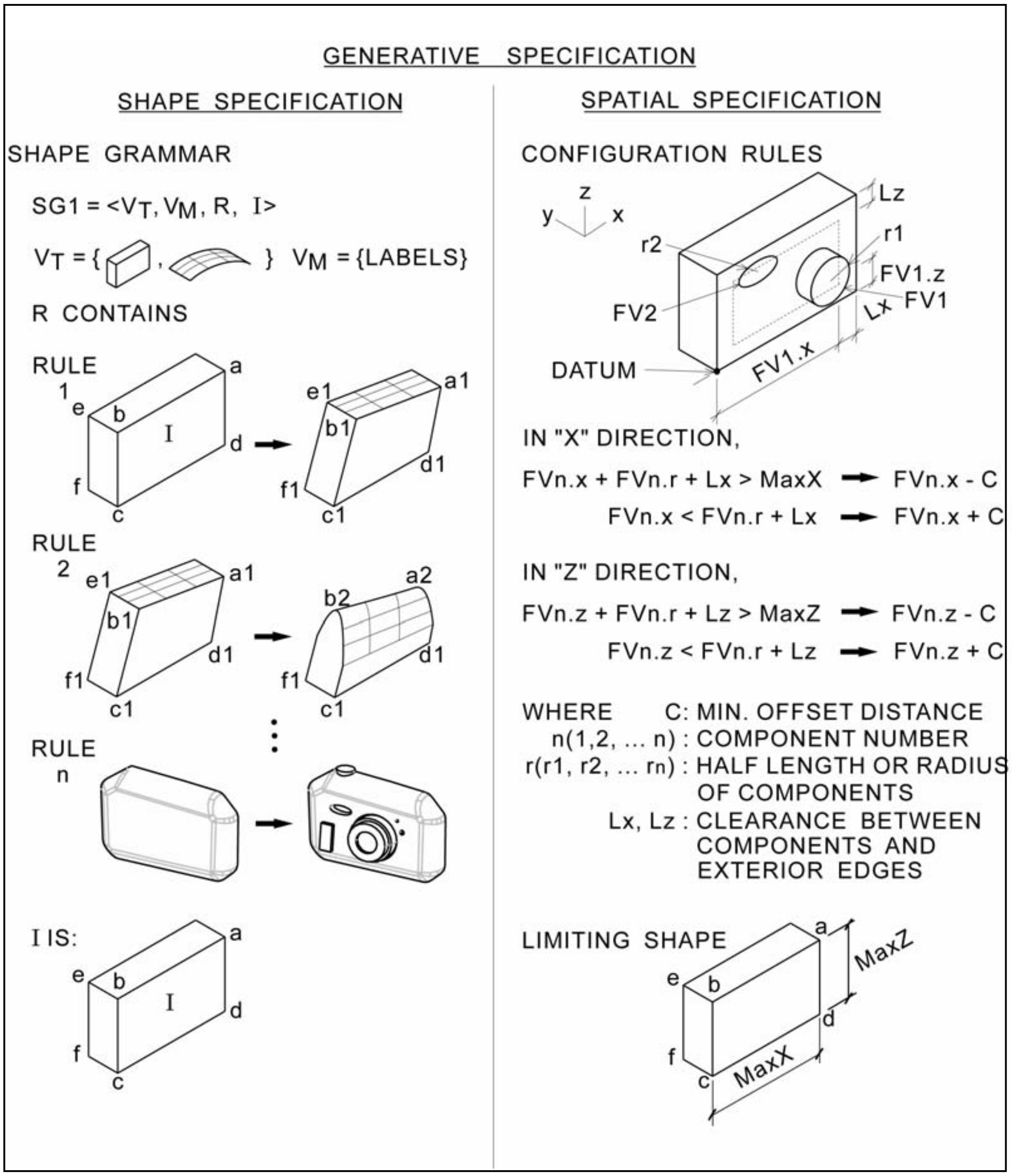


Figure 6.1 Generative specification of the class of compact digital camera forms

Therefore, it is faster to get the necessary information from the information network of SG and the Core Variant Model in developing the parametric 3D SG with labels. A class of compact digital camera forms is defined by two parameters (SH, SP). SH is a specification of a class of shapes and consists of a SG, defining a language of 3D shapes. SP is a specification of spatial configuration for the shapes defined by SH and consists of a finite list of configuration rules and a limiting shape. Figure 6.1 illustrates an example of generative specification of the class of compact digital camera forms.

Parametric 3D shape grammars with labels are developed for the generation of product forms which comprise common engineering shapes. The common engineering shapes are the vocabularies of the SG knowledge base. The common engineering shapes can be classified by their geometric properties such as free-form shapes and primitive shapes including blocks, cylinders, cones, spheres, torus, etc., and their combinations. Non-uniform rational B-spline (NURBS) surfaces are constructed to represent free-form objects in a virtual 3D spatial environment.

Labels are used to associate the control points of the NURBS surfaces and primitive shapes with the design objects. The labelled control points are used for the identification of NURBS surfaces and primitive shapes. The labels of the design objects are used as functional symbolic notations for the control of generation sequence. Both the labelled control points and the labels of design objects have values to specify their geometric coordinates in X, Y and Z axes.

In the implementation, both parametric 2D and 3D SG with labels are constructed to generate the components and the free form exterior of the main body respectively. The reasons for developing these two sets of Construction SG are that most of the components are standardised for ease of manufacture in industries. Some standardised form features of the components can therefore be generated by standard mechanical methods such as extrusion and sweeping of 2D profiles. On the other hand, some irregular form features are generated with sophisticated 3D free form shapes.

The parametric 2D SG with labels first generate the 2D profiles of components. The 2D profiles of components can then be further manipulated with different methods such as coiling, extrusion, lofting, revolving and sweeping to create 3D form features of components. In this way, the implementation time can be reduced to generate the form features of the components while longer implementation time is required for the generation of the free form exterior of the main body.

6.2.3 Construction SG Rules for Free Form Generation

An abstracted Core Variant Model representing a class of typical design of compact, durable and all-weather digital cameras is shown in figure 6.2. The abstracted Core Variant Model, visually described instead of by text in figure 6.2, is composed of combined NURBS surfaces and components. Several NURBS surfaces are combined to form the exterior of the main body.

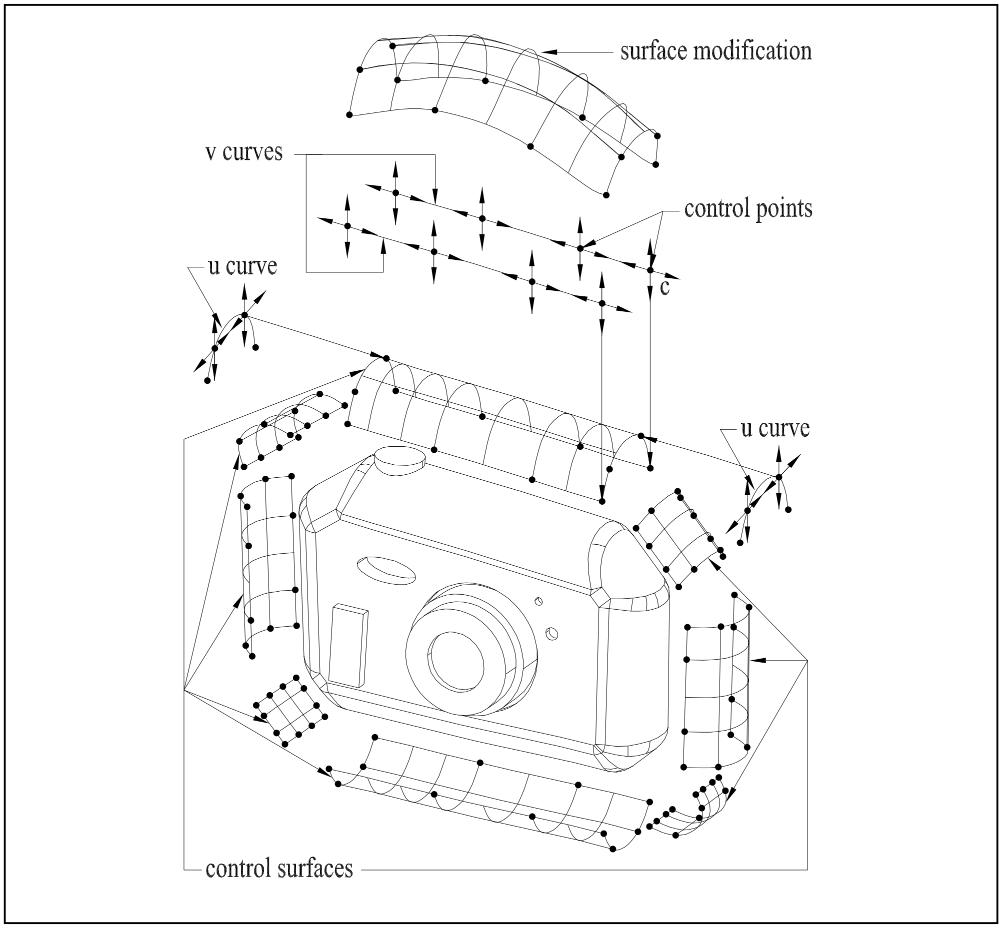


Figure 6.2 An abstracted Core Variant Model composed of combined NURBS surfaces and components for the generation of the digital camera form design

With emphasis on the aesthetic quality, the exterior of the main body must be a unique design that attracts users. This can be achieved by modifying the control points of u and v curves of each NURBS surface in the Core Variant Model. A detailed specification of the completed set of parametric 3D SG with labels for the exterior of the main body is discussed in the following paragraphs.

There are eight construction rules for free form exterior main body generation (Figure 6.3). Each rule has constraints applied to the control points with respect to their XYZ coordinates. The control points are set in the range [minX, maxX], [minY, maxY], [minZ, maxZ]. Special arrangements of the SG rules are allowed in both text and visual descriptions of the construction SG.

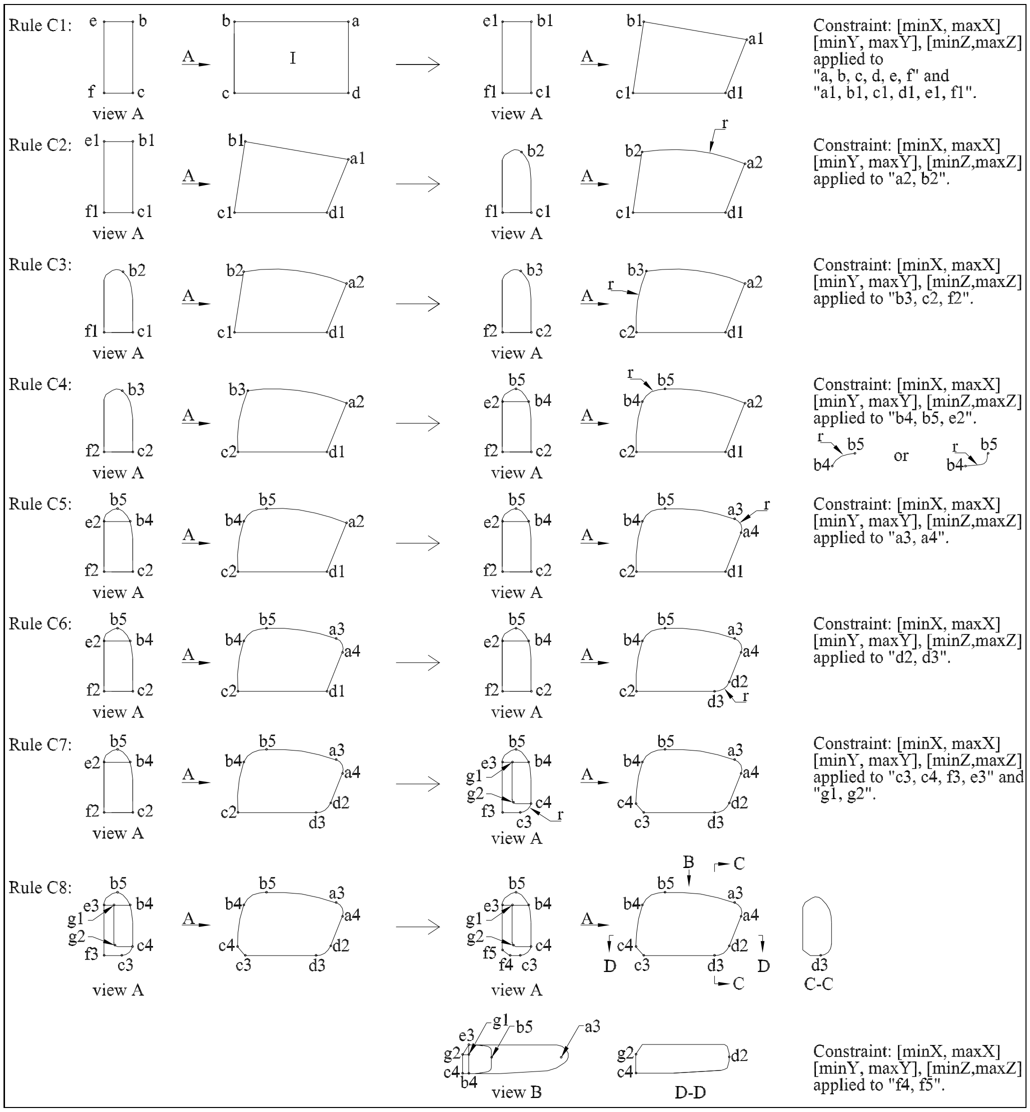


Figure 6.3 Construction SG rules for free form exterior of main body generation

Rule C1 starts with a rectangular shape with labelled points: a, b, c and d. These labelled points specify the maximum boundary of the camera body. Within the boundary, any possible form can be generated provided that the forms generated are under the constraints specified in the subsequent sets of SG rules. Rule C1 deforms the rectangular shape to a quadrilateral shape. This is the most critical transformation rule which specifies the main skeleton of the exterior camera body.

Rule C2 and C3 define an unconventional camera skeleton based on the conceptual profile of a water droplet. This can be achieved by deforming the quadrilateral shape to a curved profile. The 3D curved profile with labels c2, b3 and a2 identifies an unique digital lifestyle icon and provides handling comfort.

Rule C4 generates an arc either bending up as a round corner or down as a slot at the upper sharp corner b3. If a slot is generated, a circular shape mode dial button can be placed to the slot. The mode dial supports the ergonomic control of digital camera. The two end points of the round corner are labelled with b4 and b5.

Rules C5 and C6 modify the upper sharp corner a2 and lower sharp corner d1 of the exterior to two arcs with different radius r. The end points of the two arcs are labelled with a3, a4 and d2, d3 respectively. Avoiding sharp corners and using generous fillets and radii are an universal design rule for most of the products (Bralla, 1998). Both the manufactured part and tool can have a longer lifetime if generously rounded corners are used. In product design, generous radii and fillets are greatly preferred. The radii and fillets ensure aesthetic quality and handling comfort.

Rule C7 makes a radius along the bottom part of the exterior starting from d2 to c4. The radius becomes a fillet from c4 to g2 and continuously extends to g1. The angle and depth of the fillet must be determined to closely match with the radius, not to adversely affect the aesthetic quality.

Rule C8 creates another fillet of labels f4 and f5 along the bottom part of the exterior. Large angles and depth of fillet should be avoided as the fillet will reduce the usable area for component placement in the back side of the digital camera. Additional views (view B, section C-C and D-D) are provided for clear indication of the overall profile of the exterior of the camera body based on this rule.

6.2.4 Construction SG Rules for Component Generation

The second set of construction rules is for the design of components (Figure 6.4). Rule C9 uses the label MD to develop a unique rotating mode dial. The mode is a special feature which facilitates the ergonomic control of the camera. It allows the user to spin to set the camera. For ease of control consideration, a slot can be tailor-made by rule C4 at the top corner of the camera’s body for the mode dial to be placed. Once the user spins to select the desired camera mode, such as taking pictures, recording movies or reviewing images taken, the camera can be turned on by pressing the power switch. Therefore, the power switch is positioned close to the mode dial. The mode dial is an optional component if other control buttons with the same functionalities are used instead.

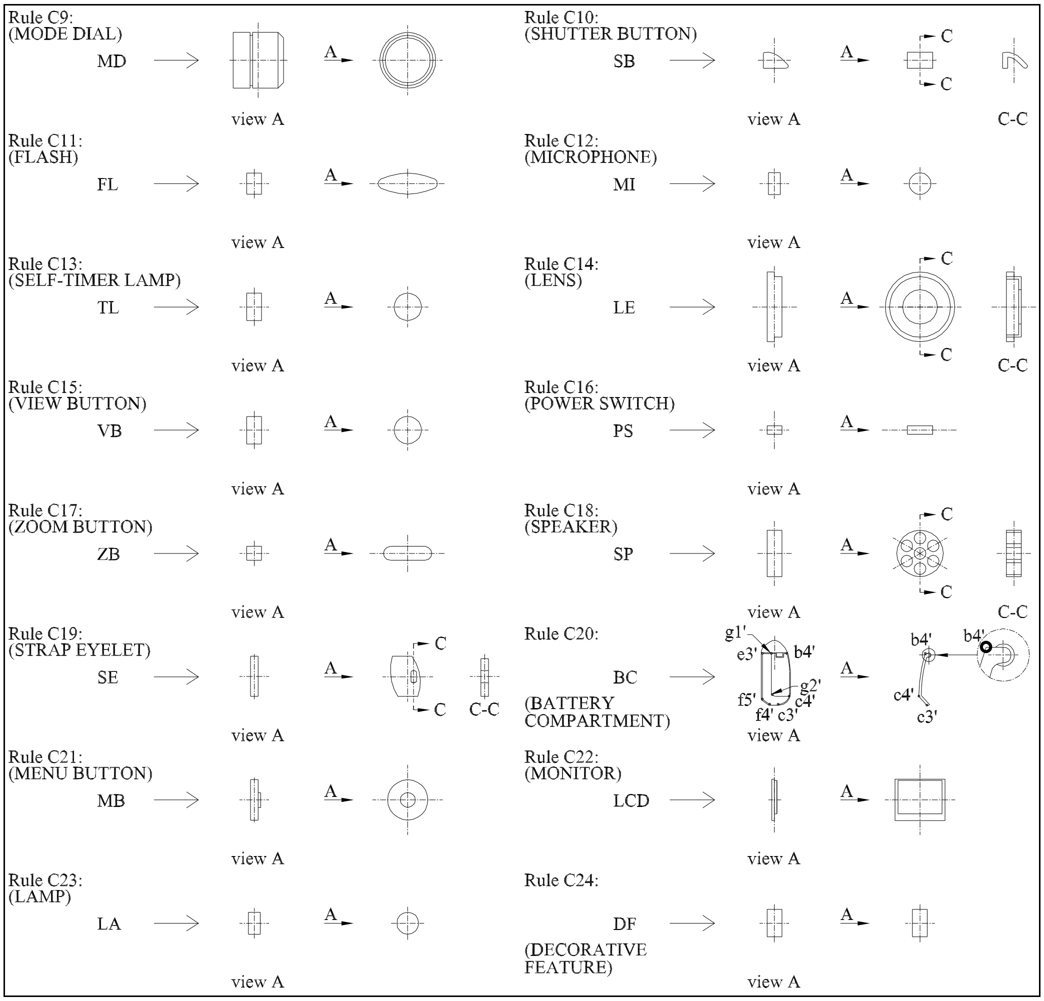


Figure 6.4 Construction SG rules for component generation

Rule C10 produces a shutter button from label SB. For conventional designs, some empirical guidelines can be followed. In designing the shutter button, the basic requirement is to facilitate the user with ease of use and comfortable feeling. Most users are right-hand; most of the shutter buttons are allocated at the top right corner of the camera. This allows the user's finger to naturally locate the shutter button. This common practice becomes the rule of thumb in allocating the shutter button position.

Rule C11 changes the label FL to an oval shape for the flash. When designing the flash for a compact size digital camera, the form will usually be designed in a rectangular shape. Other soft and practical forms like semi-circle flash shape can be used to keep uniformity to the camera body. Most of the flashes are positioned at the upper part of the camera body so that the flash is positioned higher than the lens.

This arrangement ensures flash coverage by allowing more flash light to be spread across a wider area. However, due to the design constraints of the compact digital camera size, the flash is not positioned far enough from the lens.

Rule C12 constructs a microphone which provides additional features to add value to a digital camera. The first feature is Voice Memo or Voice Annotation which allows the users to describe the photographs (still pictures) either right before or after they shoot. The second feature is Movie Mode with Sound which allows the users to take small movies, complete with sound, and process them into AVI or QT (QuickTime) files.

Rule C13 builds a self-timer lamp which facilitates part of the operations of a self-timer feature. The self-timer feature allows the users who control the camera to include themselves in pictures. The main function of the self-timer lamp is to indicate the time left before the picture is taken by blinking for approximately 10 seconds. Therefore, when designing the self-timer lamp, it should be positioned on the front of the camera with just enough size for indicating purpose.

Rule C14 generates an optical zoom lens with circular shapes. With the new zoom technology, lens elements can be compressed into a shorter space for 3X zoom. The optical zoom lens can be retracted into the body and extended from 1X to 3X. A built-in lens cover closes over the front element when the camera is powered down. Some of the camera designs provide lens thread for add-on lens or filters.

Rules C15 to C17 are for the control buttons. Rule C15 generates a quick view button which displays the last picture taken on the monitor. Rule C16 designs a power switch which turns the camera on. As stated in rule C9, it is placed next to the mode dial. The power switch and mode dial can be viewed as a pair of buttons which perform the selection and switching functions in sequence. Rule C17 generates a zoom button which controls the zoom operation.

Some of the control buttons are designed with small size. This prevents them from being pressed unintentionally. Priority has to be determined when deciding either to increase the space between buttons for shooting comfort or purposely making them small.

Rule C18 generates a speaker which provides start-up and shutter sounds. Rule C19 generates a strap eyelet which prevents dropping the camera inadvertently. Rule C20 generates a battery compartment cover for the replacement of the rechargeable battery.

Rule C21 generates a menu button which allows the user to select different settings. The menu button is a navigation control pad with four arrow control pads around the button and one OK button in the middle.

Rule C22 generates a monitor of rectangular shape for the display of pictures. A bright LCD with a size 1.8-inch display with higher pixel resolution can deliver over 160° angle of viewing and is a typical choice for monitors. Other monitors can be chosen provided that the monitors can be seen clearly from different angles and viewable even in a bright sunlight environment.

Rule C23 generates a lamp for indicating purposes when downloading the images to the computer. Rule C24 generates a decorative feature. When there are no grip features specifically designed, the decorative features serve both griping and decorative purposes.

6.2.5 Configuration Shape Grammar Rules

The third set of SG rules is configuration rules used for the allocation of components in the main body (Figure 6.5). The configuration rules use labels to maintain proper generation sequence. For the sake of clear identification of the components, rule F1 temporarily removes unnecessary labels after the generation of the exterior of the main body. Rule F2 divides the components into three groups: FRONT, BACK and SIDE according to the spatial arrangement of the components. The labels “FRONT, BACK and SIDE” refer to the components positioned with respect to the front, back and side views. Each group of components is generated sequentially in accordance to a specific generation sequence.

Rules F3 to F5 assign the components for the three groups. Rule F3 assigns seven members of labels FV1 to FV7 to the FRONT group; rule F4 assigns eight labels BV1 to BV8 to the BACK group; rule F5 assigns one label SV1 to the SIDE group. After the assignment of the components to the three groups, rule F6 allocates the components to the main body with their corresponding positions.

Rules F7 to F22 are used to control sequential generation of the components (Figure 6.6). Rules F7 to F13 generate the components for the FRONT group in sequence by modifying the labels from FV1 to FV7. Rules F14 to F21 generate the components for the BACK group in sequence by modifying the labels from BV1 to BV8. Rule F22 generates one component for the SIDE group by modifying the label SV1.

After the modification of labels, the corresponding construction rules are executed to generate the components. For example in rule F20, the label BV7 is changed to "LCD" which is matched to the label in the corresponding construction rule C22, rule C22 is then executed to generate the LCD monitor.

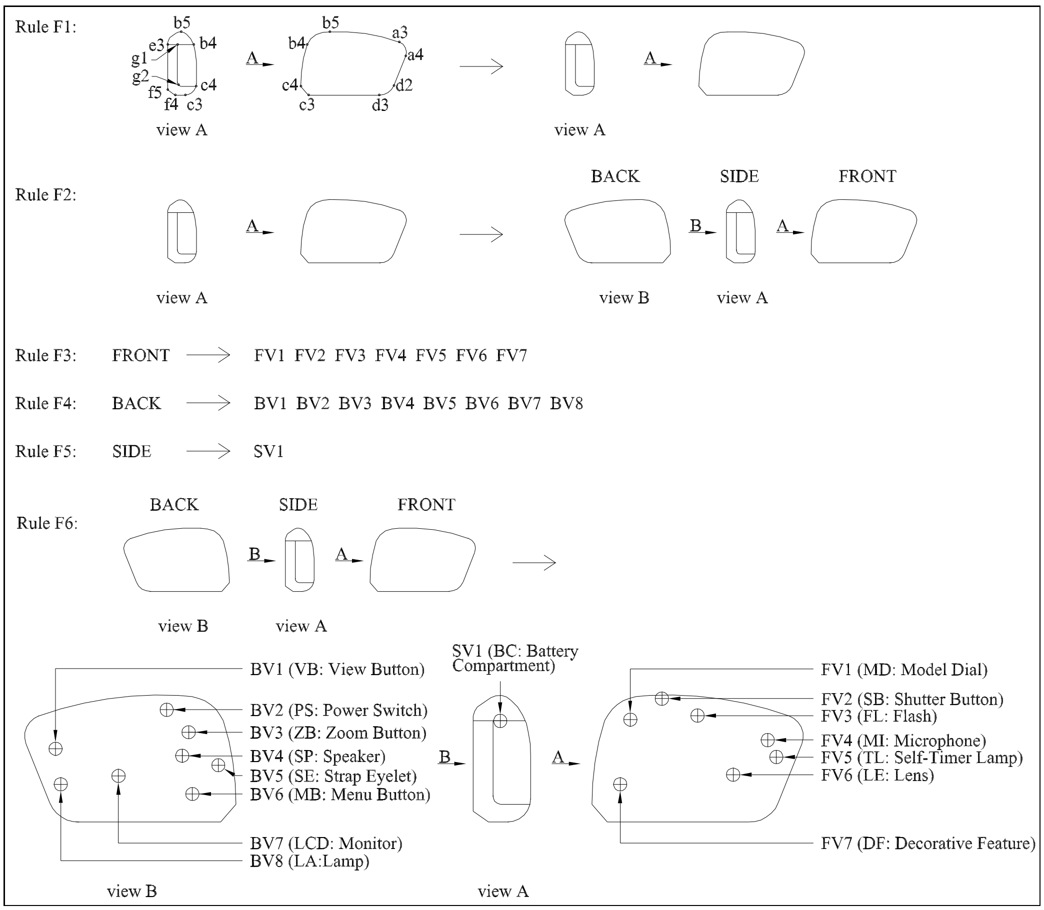


Figure 6.5 Configuration shape grammar rules for assembly

Rule F7:	FV1 FV2 FV3 FV4 FV5 FV6 FV7 ⟶ MD FV2 FV3 FV4 FV5 FV6 FV7
Rule F8:	MD FV2 FV3 FV4 FV5 FV6 FV7 ⟶ MD SB FV3 FV4 FV5 FV6 FV7
Rule F9:	MD SB FV3 FV4 FV5 FV6 FV7 ⟶ MD SB FL FV4 FV5 FV6 FV7
Rule F10:	MD SB FL FV4 FV5 FV6 FV7 ⟶ MD SB FL MI FV5 FV6 FV7
Rule F11:	MD SB FL MI FV5 FV6 FV7 ⟶ MD SB FL MI TL FV6 FV7
Rule F12:	MD SB FL MI TL FV6 FV7 ⟶ MD SB FL MI TL LE FV7
Rule F13:	MD SB FL MI TL LE FV7 ⟶ MD SB FL MI TL LE DF
Rule F14:	BV1 BV2 BV3 BV4 BV5 BV6 BV7 BV8 ⟶ VB BV2 BV3 BV4 BV5 BV6 BV7 BV8
Rule F15:	VB BV2 BV3 BV4 BV5 BV6 BV7 BV8 ⟶ VB PS BV3 BV4 BV5 BV6 BV7 BV8
Rule F16:	VB PS BV3 BV4 BV5 BV6 BV7 BV8 ⟶ VB PS ZB BV4 BV5 BV6 BV7 BV8
Rule F17:	VB PS ZB BV4 BV5 BV6 BV7 BV8 ⟶ VB PS ZB SP BV5 BV6 BV7 BV8
Rule F18:	VB PS ZB SP BV5 BV6 BV7 BV8 ⟶ VB PS ZB SP SE BV6 BV7 BV8
Rule F19:	VB PS ZB SP SE BV6 BV7 BV8 ⟶ VB PS ZB SP SE MB BV7 BV8
Rule F20:	VB PS ZB SP SE MB BV7 BV8 ⟶ VB PS ZB SP SE MB LCD BV8
Rule F21:	VB PS ZB SP SE MB LCD BV8 ⟶ VB PS ZB SP SE MB LCD LA
Rule F22:	SV1 ⟶ BC

Figure 6.6 Control of sequential generation of the components

6.3 Evolutionary Architecture Integration

This section focuses on the development of an evolutionary architecture which integrates the IGBDS for exploring new designs. New key features have been developed to overcome the limitations of the first prototype system. In particular, this section (section 6.3) presents three key features which are:

1) the new genetic representation scheme called “GP-GA-SG” interface of phenotypes and genotypes (Section 6.3.2),

2) the new manipulation methods called “Control Strategies” (Section 6.3.3 to 6.3.5), and

3) the new control mechanism which integrates the power of the control strategies and multi-objective functions (Section 6.3.3 and 6.3.7).

The above mentioned key features of the second prototype system have been developed and tested with real design examples (Lee and Tang, 2006). In order to clearly illustrate the key features, an overview of the evolutionary architecture for the second prototype system is first given.

6.3.1 Overview of the Evolutionary Architecture

Most evolutionary algorithms are developed and used as optimisation tools in solving engineering problems. Evolutionary algorithms simulate the natural genetic variation and natural selection processes in solving engineering problems. This is achieved by evolving a population of candidate solutions to a given problem using genetic operators such as crossover and mutation, and selection strategies. In the second prototype system, GP is selected as the evolutionary algorithm to explore and optimise product form designs.

In exploring product form design, this framework utilities the power of genetic and SG representations by introducing a three-layer integration interface named "GP-GA-SG". The GP-GA-SG interface is embedded in the evolutionary architecture and manipulated by the control strategies and multi-objective functions. Based on the setting of the control strategies, the GP can evolve a set of stylistically inconsistent designs and gradually modify these designs into stylistically consistent designs. The modification rate of the designs can also be modified when setting up the control strategies.

In optimising product form design, the GP performs the SG rule selection steps and the determination of parameters automatically while satisfying parametric constraints and functional requirements, such as resolving configuration conflicts for different design features, and controlling the exterior shell volume of the main body.

In building up an evolutionary architecture with GP, there are five preliminary steps to follow: choosing the terminals, the functions, the fitness function, control parameters and the termination criterion (Koza, 1992). The first two steps can be regarded as representation issues while the last three steps are manipulation issues. Both genetic representation and genetic manipulation are critical in developing the evolutionary IGBDS as both issues will affect the performance of the system in generating product form designs.

For genetic representation, a new genetic representation named "GP-GA-SG" interface is developed to utilise the power of genetic and SG representations. The SG rules and parameters are extracted from the SG and encoded in the GP-GA-SG interface.

For genetic manipulation, the control variables of the GP-GA-SG interface are regulated by the control strategies. The control strategies apply mapping and modification processes to regulate the rule sequences and parametric spatial relations among the shapes. The execution of each SG rule therefore follows an ordered sequence determined by the control strategies.

6.3.2 "GP-GA-SG" Genetic Representation

In terms of representation issues, the basic premise in developing the genetic representation for the evolutionary IGBDS concerns utilising the power of genetic and SG representations. The genetic representation should facilitate the GP to easily manipulate the SG rules. A three-layer representation interface of phenotypes and genotypes called "GP-GA-SG" is therefore developed to enhance the performance of the evolutionary IGBDS (Figure 6.7).

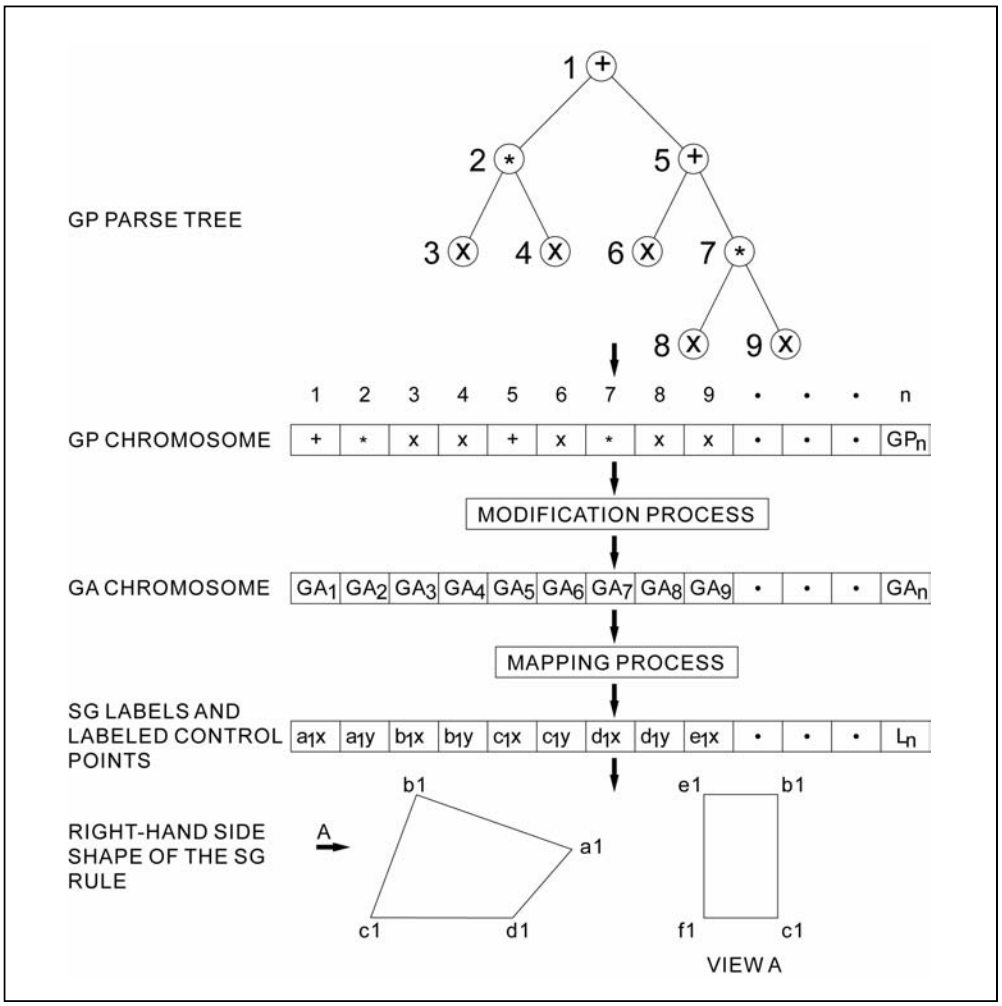


Figure 6.7 Genetic representation of the evolutionary algorithm – "GP-GA-SG" interface

The process of encoding the SG rules into the new genetic representation is demonstrated as follows:

The first layer "GP interface" is the genotype used by the evolutionary algorithm. The genotype is assigned with a set of modification variables. The modification variables are the GP components organised as tree structures. Each tree represents an evolved program and can be interpreted as a candidate solution to a given problem. Genotype is coded into one dimensional array data structure in the context of the parameters of GP components.

According to Koza's terminology, the GP components of the evolved programs consist of the terminals and the functions. The functions refer to the junctions in the tree and the terminals the end leaves. For example, figure 6.7 shows that a function like " * " takes two arguments. The function branches from the trunk into two branches in the tree. Terminals are the end leaves and can only be used as arguments to a function. Terminals might be assigned a constant such as 4 or an input such as x.

The second layer "GA interface" serves two purposes: 1) As a transformation interface interpreting the effects produced by the GP components, and 2) Encoding the SG rules in terms of their rule numbers, associated shape parameters and constraints. GA interface is coded into one dimensional array data structure in the context of "encoded" SG rule numbers, associated shape parameters and constraints.

The third layer "SG interface" is the phenotype used by both the evolutionary algorithm and SG. SG interface allows mapping between the GA elements and the SG elements. The phenotype consists of a set of SG rules and parameters which can be used by SG implementation module to generate the actual design shapes. SG interface is coded into a one dimensional array data structure in the context of the "actual representation" of SG rule numbers, associated shape parameters and constraints. All the meanings of the parameters and the relationships among these parameters in the three layers: 1) The GP interface or the genotype, 2) The GA interface, and 3) The SG interface or the phenotype interpreted in accordance to the control strategies.

6.3.3 Control Strategies

The control strategies are developed in manipulating the new genetic representation and systematically evaluating the evolving designs during the evolutionary process. The processes of manipulating the SG rules and transforming the genotypes to phenotypes are demonstrated as follows:

The control strategies aim at assigning specific sets of terminals and functions to particular types of design problems, and of monitoring the effects of the terminals and functions produced in the generated designs. Based on the choice of GP components of the evolved program (i.e. terminals and functions) and the fitness functions, a search space is then determined for GP to solve particular types of design problems.

The control strategies first define how the control variables in GP interface should modify the control variables in GA interface. The control variables in GA interface in turn modify the control variables in SG interface. Since the control variables in SG interface are the SG rules and parameters, the modified SG rules and parameters define a new combination of shape features for alternative designs.

An example of designing the exterior form and components of a compact digital camera, and determining the configuration of the components, is shown in figure 6.8. A set of parametric 3D SG rules with labels and parameters are extracted from the SG and put into the SG interface. The second step is to encode the control variables in SG interface as the 'code scripts' of the GA interface by means of the mapping process. The third step is to determine how the control variables in the GP interface should modify the control variables in the GA interface by means of the modification process. Both the mapping and modification processes are regulated by sets of equations which consist of constant-valued parameters and parametric spatial relations among shapes. Table 6.1 illustrates the details of an example of a particular type of control strategy (the first control strategy) for a compact digital camera form design.

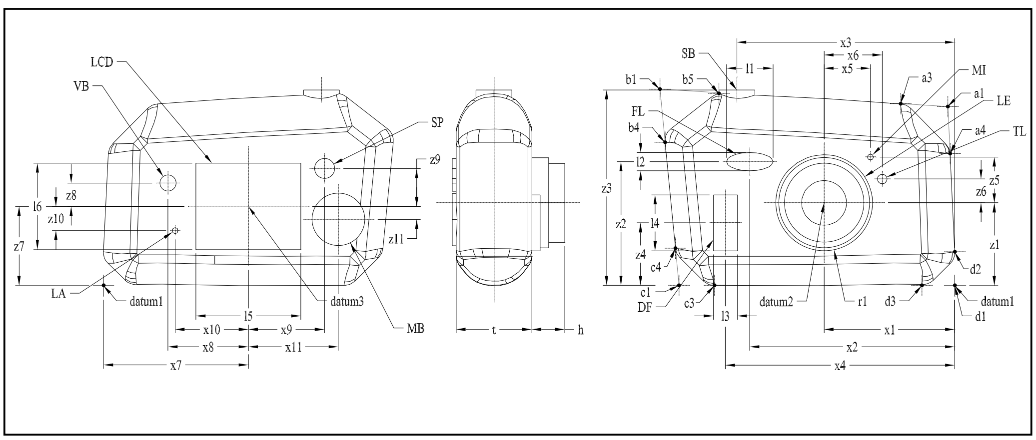


Figure 6.8 Parameters and control variables for the form design of a compact digital camera

GP groups / elements	Modificat-ion to GA interface	GA groups / elements	Mapping to SG interface	SG groups / elements
GP group 1		**GA group 1**		**SG group 1**
GP_1 to GP_8 Elements: {+, -, *, /, x} Notation: {+: +1, -: -1, *: +2, /: -2, x: 0}	$GA_n = GP_n + GA_n$ ($n = 1$ *to* 8)	GA_1 to GA_8	$a_{p1}x = GA_1 + c_{ap1x}$ $a_{p2}y = GA_2 + c_{ap2y}$ $b_{p3}x = GA_3 + c_{bp3x}$ $b_{p4}y = GA_4 + c_{bp4y}$ $c_{p5}x = GA_5 + c_{cp5x}$ $c_{p6}y = GA_6 + c_{cp6y}$ $d_{p7}x = GA_7 + c_{dp7x}$ $d_{p8}y = GA_8 + c_{dp8y}$	A subset of discrete variables $SG1_p$ ($SG1_p \in N_1, N_1 =$ $\{a_1x,...a_{n1}x, a_1y,...a_{n2}y, b_1x,...b_{n3}x,$ $b_1y,...b_{n4}y, c_1x,...c_{n5}x, c_1y,...c_{n6}y,$ $d_1x,...d_{n7}x, d_1y,...d_{n8}y\}, 0 < p < n,$ $0 < p_1 < n_1,...0 < p_8 < n_8$ where N_1 is the total set of the variables for $SG1$. n_1 to n_8 are the total number of variables for each group of control parameters. n is the total number of variables for $SG1$ which is equal to the sum of n_1 to n_8.
GP group 2		**GA group 2**		**SG group 2**
GP_9 to GP_{12} Elements and Notations: Same as GP group 1	$GA_n = GP_n$ ($n = 9$ *to* 12)	GA_9 to GA_{12}	$x_1 = GA_9 + c_{x1}$ $z_1 = GA_{10} + c_{z1}$ $x_2 = GA_{11} + c_{x2}$ $z_2 = GA_{12} + c_{z2}$	A subset of discrete variables $SG2_p$ ($SG2_p \in N_2, N_2 =$ $\{x_1,...x_{11}, z_1,...z_{11},$ $l_1,...l_6, r_1, t, h\}, \quad 0 < p < n,$ where N_2 is the total set of the variables for $SG2$ and n is the total number of variables for $SG2$.
GP group 3		**GA group 3**		**SG group 3**
GP_{13} Elements: {+, -, *, /, x} Notation: {+: 1, -: 2, *: 3, /: 4, x: 5}	$GA_{13} = GP_{13}$	GA_{13}	$FV_7 = DF_{GA13}$	A subset of discrete variables $SG3_p$ ($SG3_p \in N_3, N_3 =$ $\{FV_1,...FV_7, BV_1,...BV_8,$ $, SV_1\}, \quad 0 < p < n,$ where N_3 is the total set of the variables for $SG3$ and n is the total number of variables for $SG3$.

Table 6.1 Parameters and control variables for the first control strategy

The SG rules are grouped into three categories: 1) Exterior of the product form generation, 2) Configuration generation, and 3) Component generation. Each SG group has its corresponding GA and GP groups for monitoring the effects of the terminals and functions produced in the generated designs. Each SG group consists of modifiable elements such as construction parts, configuration, SG rules, structures and spatial relations. The SG rules in the SG groups are specified with their own parameters, for example the XYZ control variables of the labelled point "a_1" shown in figure 6.8. Table 6.2 illustrates the details of the properties in each SG group.

SG groups	Construction parts / Configuration	Rules	Structures / Spatial relations
SG group 1	Exterior	C1 to C8	Free form 3D solids
SG group 2	Configuration	F1 to F22	Spatial relations
SG group 3	Mode dial, Shutter button, Flash, Microphone, Self-timer lamp, Lens, View button, Power switch, Zoom button, Speaker, Strap eyelet, Battery compartment, Menu button, Monitor, Lamp and Decorative feature.	C9 to C24	2.5D and 3D solids

Table 6.2 Properties in each SG group

6.3.4 An Example of Control Strategy Application

One of the key issues in integrating a highly detailed SG with evolutionary computing is that the random modification properties of evolutionary computing and the capturing style properties of SG conflict with each other. The random modification of product form design removes the style of the product. More conflicts will occur if combining different SG rules to derive new shapes.

The control strategies solve this problem by controlling the modification effects produced in the generated designs. The scope of applying this methodology is similar to objective functions that continuously modify the designs until the designs are satisfy the requirements. However, the detailed implementation is different in that the control strategies focus on the modification to every single component. The designs generated can fulfil the general requirements defined by objective functions as well as specific requirements defined by control strategies.

For example, the first control strategy aims to regulate the generated designs with regular or symmetric properties. The generated designs can have different forms generated by different sets of SG rules but all designs appear to have symmetric properties. This can be achieved by regulating the differences among the control points a_1, b_1, c_1 and d_1 (Figure 6.8). The difference pairs (a_1, b_1) and (c_1, d_1) in x direction, and (a_1, d_1) and (b_1, c_1) in z direction are monitoring in the first control strategy. When either one of the difference pairs gets close to zero, the modification effects to the designs produced by the corresponding control variables of the GP-GA-SG interface stabilise. The corresponding control variables of the first layer: GP interface in the GP-GA-SG interface, have no effects produced in subsequent layers, GA interface and SG interface, except that better designs emerge.

The advantages in separating the scope in the GP-GA-SG interface includes control of modification to the final designs as performed indirectly. This lies in the principle of evolutionary algorithm that the genotype can be evaluated indirectly by evaluating the solutions (phenotype).

Every product form feature has a set of SG rules and parameters put into the third layer of GP-GA-SG interface: SG interface, the blueprint to describe how that product form feature is built up from the SG rules is encoded in the second layer: GA interface. The control of such blueprint is specified in the first layer: GP interface. In this arrangement, the modification of specific product form features can be monitored without affecting the optimisation search performed on solutions.

GP groups / elements	**Modification to GA interface**	**GA groups / elements**	**Mapping to SG interface**	**SG groups / elements**
GP group 1		**GA group 1**		**SG group 1**
GP_1 to GP_8	$GA_n = GP_n + GA_n$ ($n = 1$ *to* 8)	GA_1 to GA_8 {-2, 47, -77, 51, -72, 0, 0, 0}	$a_1x = GA_1 + 0$, $a_1y = GA_2 + 0$, $a_3x = GA_1 + (-12)$, $a_3y = GA_2 + 1$, $a_4x = GA_1 + 1$, $a_4y = GA_2 + (-12)$, $b_1x = GA_3 + 0, b_1y = GA_4 + 0$, $b_4x = GA_3 + 1, b_4y = GA_4 + (-14)$, $b_5x = GA_3 + 15, b_5y = GA_4 + (-1)$, $c_1x = GA_5 + 0, c_1y = GA_6 + 0$, $c_3x = GA_5 + 9, c_3y = GA_6 + 0$, $c_4x = GA_5 + (-1), c_4y = GA_6 + 10$, $d_1x = GA_7 + 0, d_1y = GA_8 + 0$, $d_2x = GA_7 + 0, d_2y = GA_8 + 9$, $d_3x = GA_7 + (-9), d_3y = GA_8 + 0$	$a_1x = -2$, $a_1y = 47$, $a_3x = -14$, $a_3y = 48$, $a_4x = -1$, $a_4y = 35$, $b_1x = -77, b_1y = 51$, $b_4x = -76, b_4y = 37$, $b_5x = -62, b_5y = 50$, $c_1x = -72, c_1y = 0$, $c_3x = -63, c_3y = 0$, $c_4x = -73, c_4y = 10$, $d_1x = 0, d_1y = 0$, $d_2x = 0, d_2y = 9$, $d_3x = -9, d_3y = 0$.
GP group 2		**GA group 2**		**SG group 2**
GP_9 to GP_{12}	$GA_n = GP_n$ ($n = 9$ *to* 12)	GA_9 to GA_{12} {-34, 22, -54, 33}	$x_1 = GA_9 + 0, z_1 = GA_{10} + 0$, $x_2 = GA_{11} + 0, z_2 = GA_{12} + 0$.	$x_1 = -34$, $z_1 = 22$, $x_2 = -54$, $z_2 = 33$.
GP group 3		**GA group 3**		**SG group 3**
GP_{13}	$GA_{13} = GP_{13}$	$GA_{13}\{1\}$	$FV_7 = DF_{GA13}$	$FV_7 = DF_1$

Table 6.3 Initial setting of parameters and control variables for the first control strategy

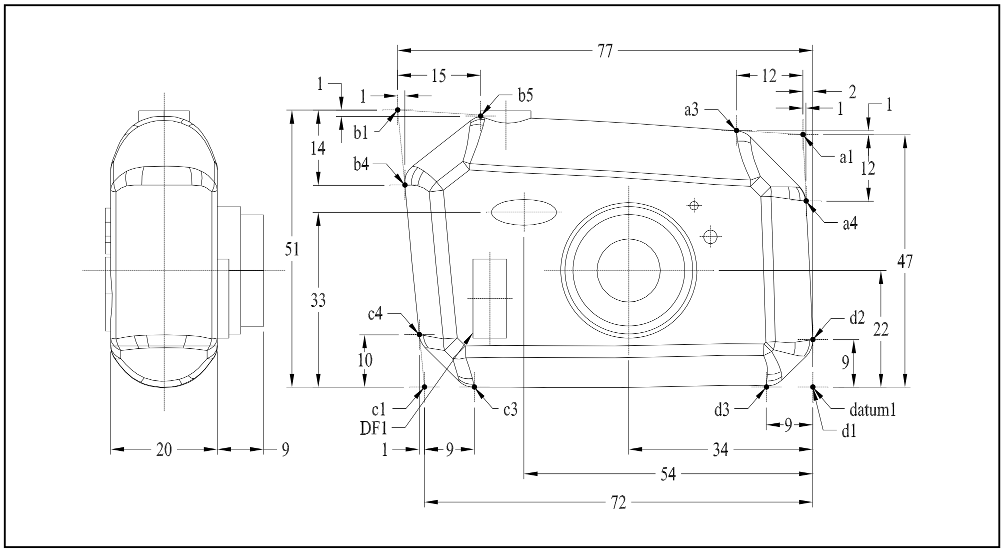


Figure 6.9 Initial setting of parameters and control variables for a compact digital camera

The first control strategy is developed to illustrate the methodology for the regular or symmetric type of product form designs. An example of the initial setting of the first control strategy is shown in Figure 6.9. The equations for modification and mapping processes between GP interface to GA interface, and GA interface to SG interface were shown in Table 6.1. Initial setting of the parameters for the GA interface and SG interface are specified in Table 6.3. An illustration of the procedures to apply the first control strategy follows in the next section.

6.3.5 Operation of Control Strategy

An example of crossover operation is demonstrated and shown in figure 6.10. After the application of the crossover operation, the first control strategy determines the GP parameters. The GP parameters then modify the GA parameters which in turn map to the SG parameters. The resulting design is shown in Figure 6.11 and the modified setting of the parameters and control variables is shown in table 6.4.

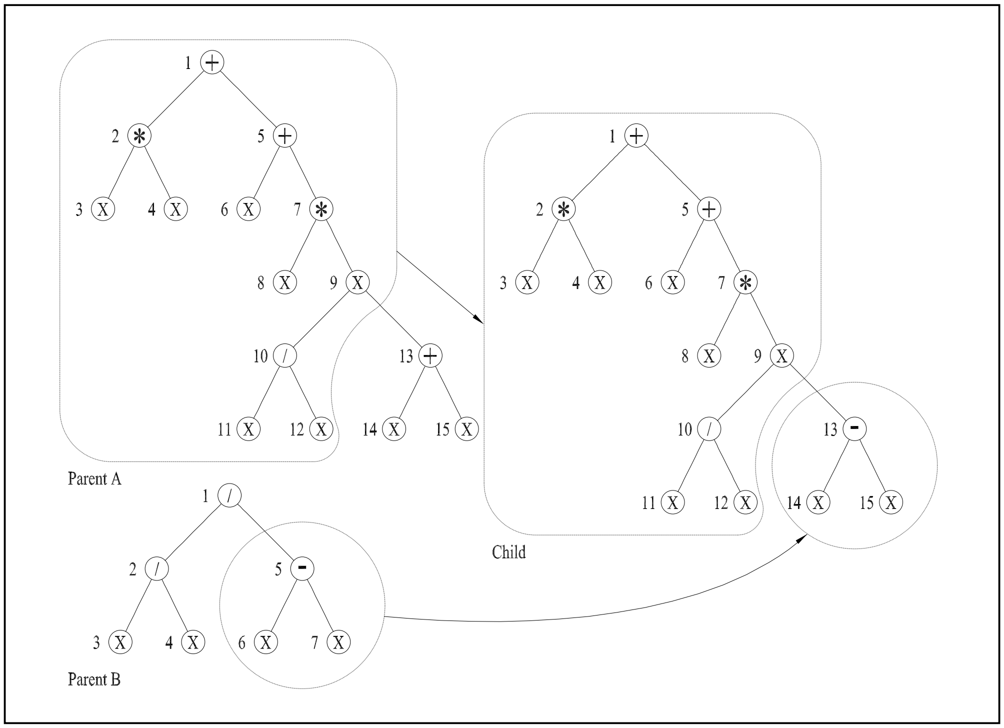


Figure 6.10 GP crossover – Parent A crossed with parent B to produce the child

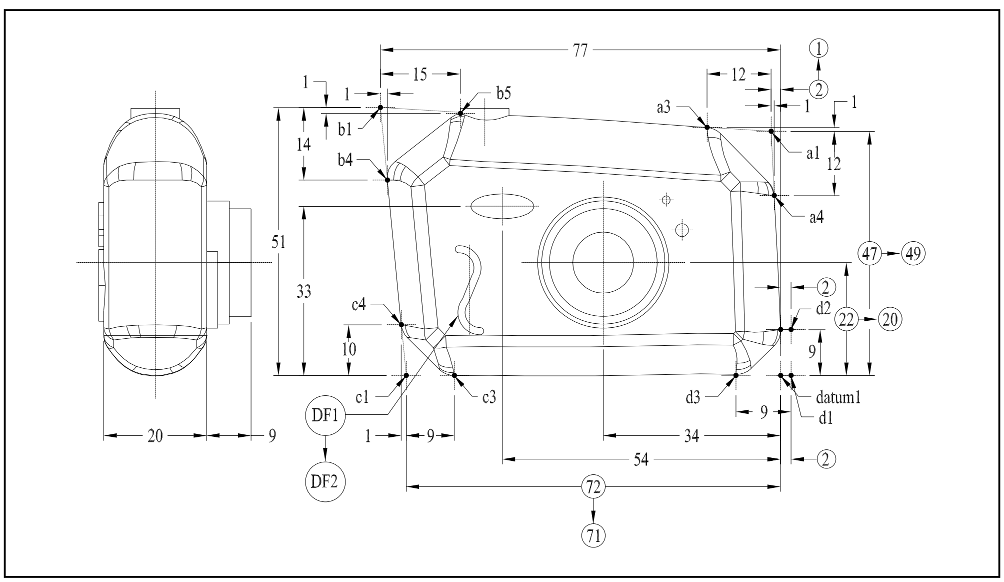


Figure 6.11 Modified setting of parameters and control variables by crossover operation for the first control strategy

GP groups / elements	**Modification to GA interface**	**GA groups / elements**	**Mapping to SG interface**	**SG groups / elements**
GP group 1		**GA group 1**		**SG group 1**
GP_1 to GP_8 {+, *, x, x, +, x, *, x}	$GA_1 = GP_1 + (-2) = 1 - 2 = -1$, $GA_2 = GP_2 + 47 = 2 + 47 = 49$, $GA_3 = GP_3 + (-77) = 0 - 77 = -77$, $GA_4 = GP_4 + 51 = 0 + 51 = 51$, $GA_5 = GP_5 + (-72) = 1 - 72 = -71$, $GA_6 = GP_6 + 0 = 0 + 0 = 0$, $GA_7 = GP_7 + 0 = 2 + 0 = 2$, $GA_8 = GP_8 + 0 = 0 + 0 = 0$.	GA_1 to GA_8 {-1, 49, -77, 51, -71, 0, 2, 0}	Same as initial setting.	$a_1x = -1, a_1y = 49$, $a_3x = -13, a_3y = 50$, $a_4x = 0, a_4y = 37$, $b_1x = -77, b_1y = 51$, $b_4x = -76, b_4y = 37$, $b_5x = -62, b_5y = 50$, $c_1x = -71, c_1y = 0$, $c_3x = -62, c_3y = 0$, $c_4x = -72, c_4y = 10$, $d_1x = 2, d_1y = 0$, $d_2x = 2, d_2y = 9$, $d_3x = -7, d_3y = 0$.
GP group 2		**GA group 2**		**SG group 2**
GP_9 to GP_{12} {x, /, x, x}	$GA_9 = GP_9 + (-34) = 0 - 34 = -34$, $GA_{10} = GP_{10} + 22 = -2 + 22 = 20$, $GA_{11} = GP_{11} + (-54) = 0 - 54 = -54$, $GA_{12} = GP_{12} + 33 = 0 + 33 = 33$.	GA_9 to GA_{12} {-34, 20, -54, 33}	Same as initial setting.	$x_1 = -34, z_1 = 20$, $x_2 = -54, z_2 = 33$.
GP group 3		**GA group 3**		**SG group 3**
GP_{13} {-}	$GA_{13} = GP_{13} = 2$	GA_{13} {2}	Same as initial setting.	$FV_7 = DF_2$

Table 6.4 Modified setting of parameters and control variables by crossover operation for the first control strategy

6.3.6 Genetic Programming

In the evolutionary IGBDS, the GP performs three main functions: 1) Modifying alleles within chromosomes using genetic operators, 2) Decoding the genotype to produce the phenotype in accordance to the control strategies, and 3) Evaluating the phenotype to identify the fittest solutions.

At the beginning of running the system, the GP generates an initial population of 500 individuals with random values. Due to the complexity of displaying the virtual models in the limited display area of computer screen, a maximum of twelve individuals are extracted from the population for visualisation. However, the designers can choose, to keep the displayed selected designs during evolution to trace the modification effects on the selected designs, or to replace the selected designs with the fittest ones during evolution while searching the best designs.

The main loop begins at this stage. Each individual is then evaluated and assigned a fitness value by fitness functions and artificial selection. Based on the scores obtained from each solution, the solutions with higher scores will be selectively copied to a temporary area termed ‘mating pool’.

Entering to the second loop, two of the solutions are randomly selected as parents from this ‘mating pool’. These two parents generate two offspring by random crossover and mutation operators and replace the parents of the population. The crossover and mutation processes repeat to generate offspring until every parent of the old population is replaced, a new population with fitter solutions is then established.

For each generation, the genotype is converted into the phenotype which represents the solutions. The solutions are a number of individuals, each of which consists of a set of parametric 3D SG rules with labels and parameters. After execution of the SG rules by the SG implementation module in accordance to the generated rule sequences, both the exterior main bodies and components are generated. The GP repeats the main loop of evaluation and reproduction processes for a specified number of generations, or the GP will stop if satisfactory solutions emerge.

6.3.7 Multi-objective Functions

Exterior form generation of compact digital cameras and the configuration of the components are designed to fulfil a set of requirements such as artificial selection, spatial geometric constraints and desired exterior shell volume. The design requirements can be formulated into objective functions. Objective functions are set up for the evaluation of the generated designs. General objective functions are set up for general requirements while control strategies have their own sets of objective functions for specific requirements. Analysis of the evaluation results will help in the investigation of and understanding of combinatorial effects on the generated designs based on the control strategies. To effectively evaluate the design performance, a metric is formulated as the summation of design objectives and weighted constraint violations.

$$\begin{aligned} Index\ function &= Objective\ \ index + Constraint\ \ index \\ &= \sum_{i=1}^{l} Objective\ \ index_i + \sum_{j=1}^{m} Constraint\ \ index_j \end{aligned} \tag{1}$$

$$\begin{aligned} where: l &= number\ of\ objectives, \\ m &= number\ of\ constraints. \end{aligned}$$

Objective and penalty functions are defined to assign positive and negative fitness scores respectively. Penalty functions are activated if the generated designs violate the constraints. Both design objectives and constraints have weighting factors to determine the relative trade-off among design objectives. The designers can assign different weighting factors on each variable.

$$Objective\ \ index = \sum_{i=1}^{l} (Objective\ \ weight_i \bullet Objective\ \ value_i) \tag{2}$$

$$where: l = number\ of\ objectives.$$

$$Constraint\ \ index = \sum_{j=1}^{m} (Constraint\ \ weight_j \bullet Constraint\ \ violation_j) \tag{3}$$

$$where: m = number\ of\ constraints.$$

For the artificial selection requirements, $Objective\ \ index_1$ is used as the measurement of accumulated effect on selected designs. The selected designs will be assigned with higher fitness scores if they are frequently selected by the designers.

$$\begin{aligned} Objective\ \ index_1 &= \sum_{i=1}^{n} (Selection\ \ weight_i \bullet Selection\ \ value_i) \\ &\{Selection\ \ value_i = 0\ \ or\ \ 1\} \end{aligned} \tag{4}$$

where n = number of generations; $Objective\ \ index_1$ is the accumulated score for each design; $Selection\ \ weight_i$ is the weighting factor for each design; $Selection\ \ value_i$ is assigned with 1 when the designs are selected, otherwise 0. Since the selection cost of each design is the accumulated score from each generation, selection on one or more designs in a particular generation will not significantly impact the whole population. As a result, the population is determined by the accumulated effect on the selected designs.

Under the spatial geometric constraints, the components have to be configured without collision among each other and within the boundary of the exterior of camera body. Geometric variables of the component positions and the boundary positions of the exterior of the camera body are assumed to be configuration design variables, subject to a set of constraints. The objective functions of configuration of components can be determined by the designers with selective options. For example, the selective options of configuration are: to maximize or minimize the total distance (TD_1) among components.

$$\begin{aligned}
&\textit{For Maximize option selected:}\\
&\textit{Objective index}_2 = \textit{Configuration weight} \bullet TD_1 \qquad (5)\\
&\textit{For Minimize option selected:}\\
&\textit{Objective index}_2 = \textit{Configuration weight} \bullet \frac{1}{TD_1 + C} \qquad (6)
\end{aligned}$$

$$TD_1 = \sum_{i=1}^{n}\sum_{j=1}^{n} d_{ij},\ \{i \neq j\} \qquad (7)$$

$$\begin{aligned}
&\textit{Subject to (a set of constraints):}\\
&d_{ij} \geq l_i + l_j + l_c,\ \{i = 1 \text{ or } 2 \text{ or} ,..., \text{ or } n\},\\
&\{j = 1 \text{ or } 2 \text{ or} ,..., \text{ or } n\} \text{ and } \{i \neq j\}
\end{aligned}$$

$$\textit{Constraint index}_1 = \sum_{i=1}^{n}\ \sum_{j=1}^{n}(\textit{Configuration constraint weight} \bullet \textit{Constraint violation}_{ij}) \qquad (8)$$

$$\begin{aligned}
&\{\textit{Constraint violation}_{ij} = -(l_i + l_j + l_c - d_{ij}),\ \textit{if the constraints are violated}\},\\
&\{\textit{Constraint violation}_{ij} = 0,\ \textit{if the constraints are not violated}\}.
\end{aligned}$$

where C is a constant; n is the number of components; l_i , l_j are the half length or radius of components; l_c is the clearance between components; coefficient d_{ij} is the distance between components i and j. The distance between two components is defined as the distance between the centres of both components as shown in Figure 2. The summation of all the distances between any two components (TD_1) reflects the dispersion among components.

For exterior shell volume calculation, the objective is to minimize the difference between the shell volume and a desired target shell volume of the exterior of camera body.

$$\textit{Objective index}_3 = (\textit{Volume weight} \bullet f(v)) \qquad (9)$$

$$\textit{Minimise } f(v) = \frac{1}{\left|(v - v_{target})\right| + C} \qquad (10)$$

$$\textit{where: } C = \textit{constant}.$$

The value of an exterior shell, v, refers to the approximate volume estimation of the exterior of the camera body. A constant C is added to $\textit{Objective index}_2$ and $f(v)$ to ensure that the objective indices take only positive values in their domains (Michalewicz, 1996). The addition of constant C to the objective indices also avoids the error arising from dividing zero.

Since multi-objective functions exist in the evolutionary IGBDS, a wide number of designs belonging to the Pareto Optimal Front (POF) can be identified. To simplify the implementation, the use of a weighting approach is sufficient to explore different settings of parameters. Further study of an advanced POF technique will lead to performance improvement.

6.4 System Development

In this section, the system development of basic elements and control mechanisms for the second prototype of the integrated SG and evolutionary algorithm framework are discussed.

6.4.1 System Architecture

A classical genetic programming (GP) is used as the core of the evolutionary architecture for evolving SG rules. The evolved SG rules are used to generate different new designs. Figure 1.2 shown in chapter one outlines such an approach to support the design process with two key elements: SG as the knowledge base for design, and evolutionary computing as the generative mechanism.

For the first element, the form features of compact digital cameras are abstracted from several different brands in accordance to the systematic approach. These form features represent different components categorised according to their spatial relationships among components based on geometric locations in the assembly. The form features considered in the second prototype system are categorised into two main groups: 1) The exterior of the main body and 2) Components such as mode dial, shutter button, flash, microphone, self-timer lamp, lens, view button, power switch, zoom button, speaker, strap eyelet, battery compartment, menu button, monitor, lamp, grip and decorative feature (a total of 17 components for group 2). Three sets of parametric 2D and 3D SG rules with labels are established for the form feature generation and the determination of their corresponding geometric locations in the assembly: 1) Exterior of the product form generation, 2) Component generation, and 3) Configuration generation. There are 46 core SG rules used in the second prototype system.

For the second element, the algorithm chosen for evolutionary computing is based on GP (Koza, 1992). GP is a subclass of GA, with the aim for getting computers to automatically solve a problem (Koza et al., 2003). GP creates a computer program to solve the problem. The evolving programs are directly represented in the chromosome as trees. The evolving programs consist of components (i.e. terminals and functions) which are predefined at the beginning of the evolution. In the systematic approach, the evolving programs are assigned to different control strategies. In the control strategies, the effects of components are controlled during the evolutionary process. The control strategies can also specify the modification rate of the form features. Initial case studies demonstrated that the control strategies are capable of evolving stylistically inconsistent designs which are gradually modified into stylistically consistent designs.

At the beginning of running the system, the designers first input a set of design criteria by specifying design control plan types, shape parameters and initial setting of objective functions, e.g. weight factors. Entering the evolutionary cycle, based on the designers' inputs, the selected control strategy determines the GP parameters. The GP parameters then modify the GA parameters which in turn map to the SG parameters. The SG implementation module then generates the actual design shapes based on the SG parameters. The actual design shapes are evaluated by the evaluation module. If the results are not satisfactory, the designers can intuitively select satisfactory better designs, modify the objective functions and/or reset the control parameters. Genetic operations such as crossover and mutations will then be used to evolve the SG in accordance to the control strategies. Another evolutionary cycle starts and repeats until satisfactory results emerge or maximum generations are reached.

6.4.2 System Operations

The second prototype system has been developed to enhance the generative capability of SG. Parametric 2D and 3D SG with labels are used in the second prototype system to generate new 3D designs. The SG rules are composed of rule numbers, shapes and their corresponding parameters. Each SG rule is classified into three main groups: 1) Exterior of the product form generation, 2) Component generation, and 3) Configuration generation.

The SG rules and parameters are extracted from the SG and encoded in the GP-GA-SG interface. The control variables of the GP-GA-SG interface are regulated by the control strategies. The control strategies apply mapping and modification processes to regulate the rule sequences and parametric spatial relations between control variables. The execution of each SG rule is therefore followed by an ordered sequence determined by the control strategies. The evolutionary cycles continue to run until satisfactory solutions emerge as described in section 6.3.6.

6.5 Implementation Results of the Second Prototype

The second prototype evolutionary IGBDS has been developed using Visual C++ and ACIS 3D modelling kernel, and tested. Figure 6.12 shows the designer interface of the evolutionary IGBDS.

6.5.1 Initial Setting of the System

At the beginning of running the evolutionary IGBDS, the designers first input a set of design criteria such as selecting design control plan types, specifying types of components and their corresponding shape parameters, and initial setting of objective functions, e.g. weighting factors.

6.5.2 Implementation Results – First Experiment

Entering the evolutionary cycle, at the first generation, the system applies the construction and configuration rules to randomly generate a population of designs (Figure 6.13).

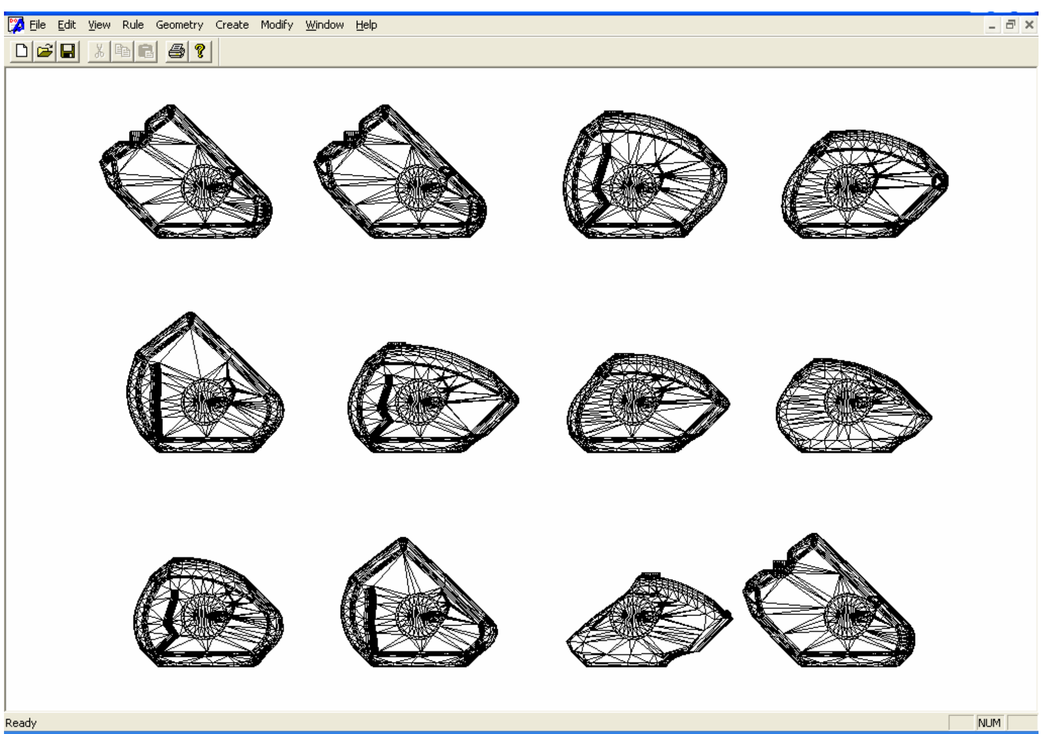


Figure 6.12 User interface of the evolutionary IGBDS

The initial form style of each compact digital camera is defined by rule C1 with a pocket-size metal body of rectangular shape. It is deformed to a quadrilateral shape which is the main skeleton of the exterior camera body by rule C1 (Figure 6.3). Rules C2 to C8 then modify the main skeleton of the exterior of camera body to a curved profile (Figure 6.3).

Rule C9 to C24 generate the components: rotating mode dial (by rule C9), shutter button (by rule C10), flash (by rule C11), microphone (by rule C12), self-timer lamp (by rule C13), optical zoom lens (by rule C14), quick view button (by rule C15), power switch (by rule C16), zoom button (by rule C17), speaker (by rule C18), strap eyelet (by rule C19), battery compartment (by rule C20), menu button (by rule C21), monitor (by rule C22), lamp (by rule C23) and decorative feature (by rule C24) respectively (Figure 6.4). The SG implementation module generates the actual design shapes based on the SG parameters.

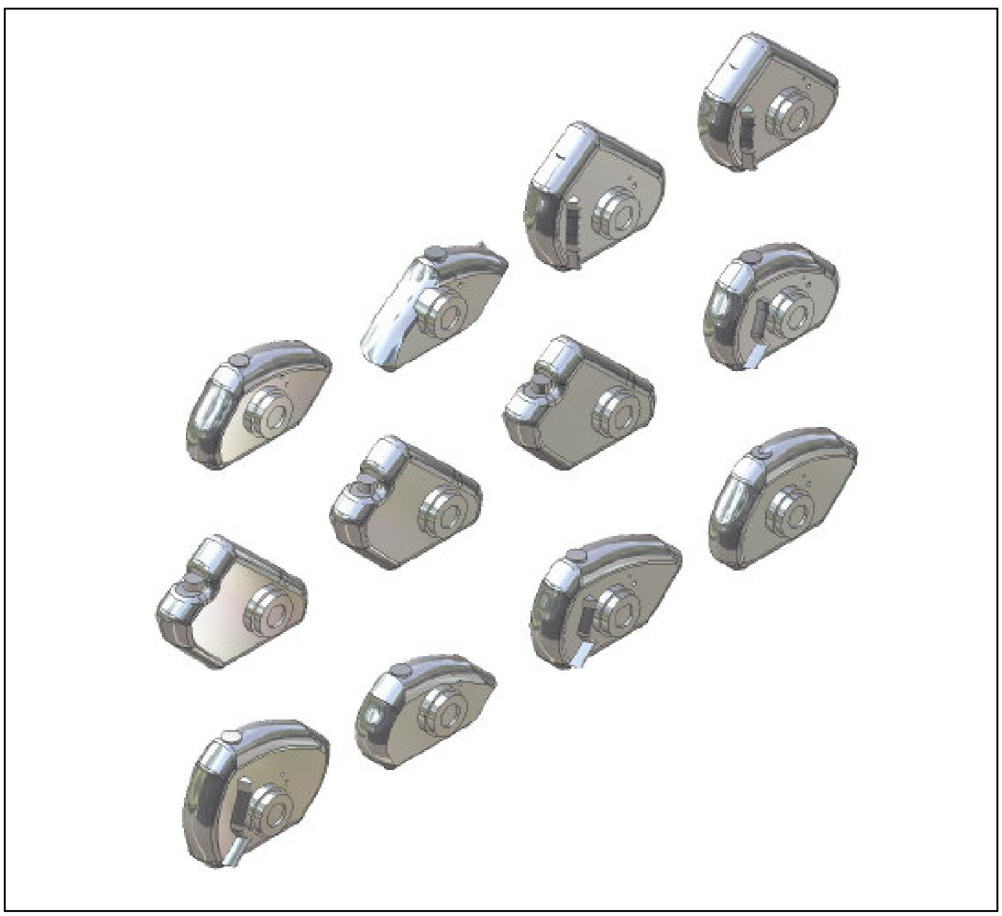


Figure 6.13 Initial random generation of designs

After all the components have been generated in order, they are positioned in the camera body in accordance to the configuration rules F1 to F22 (Figures 6.5 and 6.6). In the implementation, all the significant components are generated to demonstrate the potential usage of the evolutionary IGBDS as shown in figure 6.12, 6.13 and 6.14 while leaving some insignificant components to be implemented in the future. The actual design shapes are evaluated by the evaluation module. If the results are not satisfactory, the designers can modify the objective functions, reset the control parameters or intuitively select the generated designs. Genetic operations such as crossover and mutations will then be applied to evolve the SG rules.

6.5.3 Application of the First Control Strategy

In order to clearly demonstrate the operations of the system, the first control strategy for designing regular or symmetric type designs is first applied and illustrated with examples. The descriptions of the first control strategy can be referred to section 6.3.4. An illustration of the procedures in applying the first control strategy that the system has implemented implicitly is depicted with diagrams and data sheets.

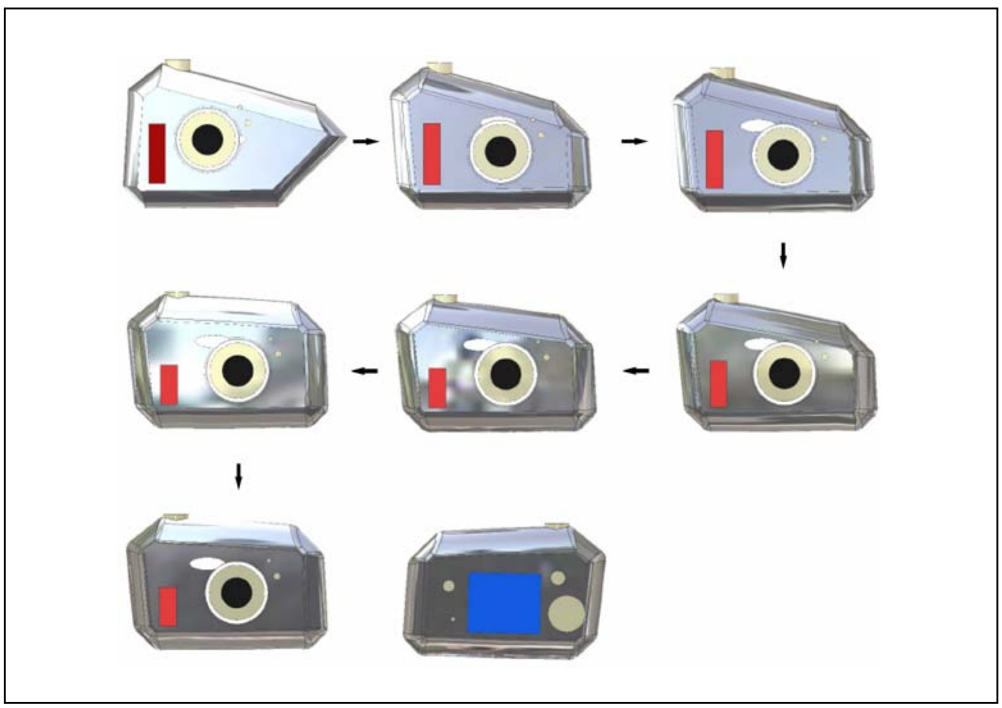


Figure 6.14 Regular type designs - results obtained from the first generation (top left), 50 generations (top middle), 100 generations (top right), 150 generations (middle right), 200 generations (middle middle), 250 generations (middle left), 300 generations (bottom left) and the back view of the generated design (bottom middle)

This experiment emphasises and illustrates the interactions between the designer and the system. Continuing with the operations after the initial running of the system, the designer can select the favourable design intuitively from the twelve designs. The designer can also choose to keep the all displayed designs during evolution for tracing the modification effects on the designs. The modification effects on the selected design are shown in figure 6.14.

By adjusting the parameters of the objective functions and selecting the appropriate control strategies, the designer can flexibly study the effects on the generated designs and then determine which strategy is most suitable for a particular application. Other control strategies such as slim, asymmetric and mixed can also be defined to test the flexibility and effectiveness of the evolutionary IGBDS approach in product form design generation. Finally, another evolutionary cycle starts and repeats until satisfactory results emerge or maximum generations are reached. Another experiment has been conducted to test the system using other control strategies for new requirements illustrated in the next section.

6.6 Implementation Results - Second Experiment

This second experiment shows how the second prototype system strategically applies the control strategies and multi-objective functions to control the evolving SG rules for the generation of new designs with particularly desired design characteristics.

6.6.1 Initial Setting of the System

The setting of the evolutionary IGBDS is initialised by the designer prior to the system runs. By setting the population size to be 500, crossover rate 0.6, and mutation rate 0.01, the system generates the designs in accordance to different requirements. Implementation examples are carefully planned to demonstrate how the designers can interact with the system and what the results would be in respect to the requirements. By setting the control parameters of the evolutionary IGBDS in each periodically observed generation in a tabular format and by evaluating the corresponding results visually and numerically, a clear picture of the complex effects produced by the objective functions is depicted. Based on the analysis of the results, the designer can select appropriate control parameters and control strategies to explore designs during the evolutionary design process. The process should follow the procedures as specified in the scenario as shown in section 1.5.2., and continuously run until reaching the ending conditions.

6.6.2 Implementation Results

Figure 6.15 shows the implementation results obtained from the second prototype system, starting at the first generation and ending at five hundred generations. The generated models can be post-processed by other commercial software for rendering with surface contour patterns. The surface contour patterns allow the designers to evaluate the surface quality of the generated models more effectively. The designers can visually inspect the continuity between surfaces of the generated models.

Together with the aid of a comprehensive table listing all the relevant information of the evolving forms of a product, results can be analysed numerically. Table 6.5 depicts the detailed specifications and control parameters of the system. The detailed specifications include: 1) Design number, 2) Control strategy, 3) Main objective index, 4) Artificial selection fitness, 5) Configuration fitness, 6) Volume fitness, 7) Configuration constraint index, and 8) Volume estimation, while the control parameters include the parameters of GP groups (item 9 to 11), GA groups (item 12 to 14) and SG groups (item 15 to 20).

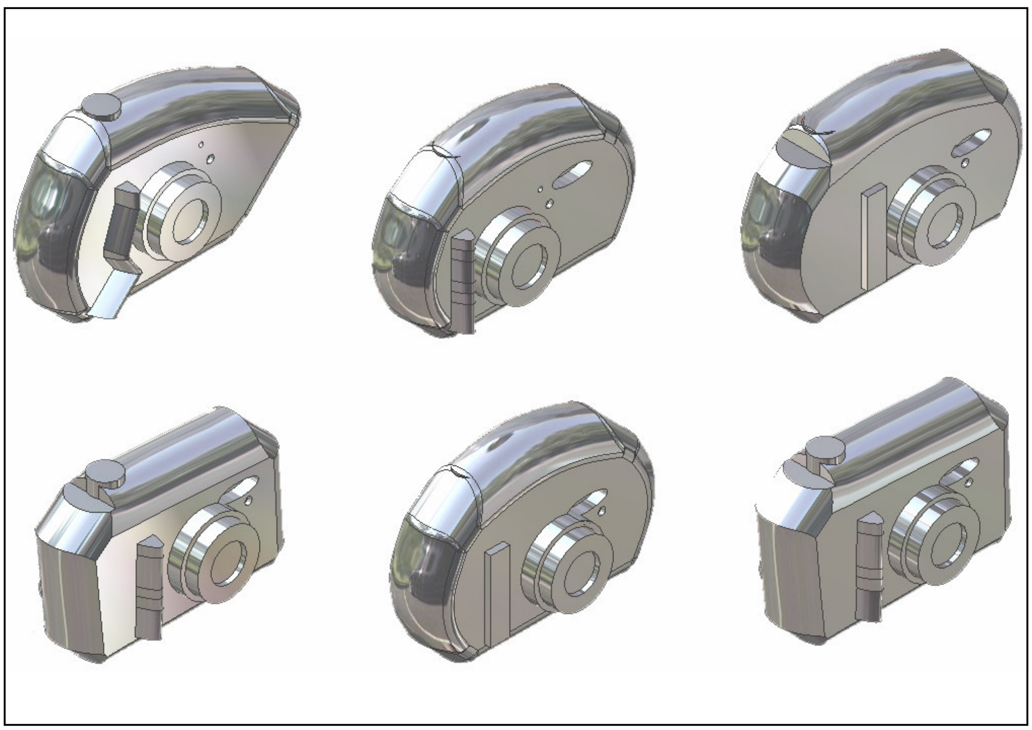


Figure 6.15 Results obtained from the first generation (top left), 100 generations (top middle), 200 generations (top right), 300 generations (bottom left), 400 generations (bottom middle) and 500 generations (bottom right)

6.6.3 Evaluation of Implementation Results

In order to analyse the generated results, a complete historical record showing how the designers interact with the system is depicted. The second and third control strategies are selected to illustrate the regulation of the generated designs with symmetric properties. Each control strategy has different form features as shown in figure 6.15.

- At the first generation, the GP generates an initial population of 500 individuals with random values. All the weighting factors are pre-set to 1.0 except the configuration weighting factor is pre-set to 0.1, and the target shell volume is pre-set to 24.528 cm^3. The designers start to modify the control parameters. First, the result of design number: 494 is intuitively selected by the designers. Second, the designers select the third control strategy and modify the selection weighting factor and the configuration constraint weighting factor with value: 10. Then, the designers evaluate the results at each periodically observed generation (every 100 generations).

- At 100 generations, all the designs are generated based on the third control strategy. Most of the generated designs have similar form features to the previously selected design. This is caused by setting the selection weighting factor with value: 10. The artificial selection criterion influences the configuration of components and the selection of decorative features. Also, the adverse effects of constraint violation between components of the generated designs are improved. This is caused by setting the configuration constraint weighting factor with value: 10. As an example, the generated design (number: 456) indicates that the configuration constraint is of zero value. It can be visually determined by the designers that the lens and the flash are dispersed far apart from each other on the exterior of the main body.

- After evaluation of the generated designs, the designers then continue to modify the control parameters. The result of design number: 456 is intuitively selected by the designers. The designers apply the same settings of control parameters for the next 100 generations.

- At 200 generations, all the designs are generated based on the third control strategy. The previously selected design would probably not appear in the first few periodically observed generations. This is because the selection fitness value is based on the accumulated scores of selected designs. Since the accumulated scores of selected designs in the first few observed generations are not significant, the chances of the selected design appearing in the subsequent observed generations becomes small. Therefore, most of the artificial selection fitness values of the selected designs are zero in the first few observed generations.

- After evaluation of the generated designs, the designers then continue to modify the control parameters. The result of design number: 299 is intuitively selected by the designers. The designers would like to explore other types of product form designs by selecting the second control strategy and setting the target volume to be 27 cm^3.

- At 300 generations, all the designs are generated based on the second control strategy. Unexpected outcomes astonish the designers, most of the generated designs get poor volume fitness values. This is caused by the low volume weighting factor with value 1.

- After evaluation of the generated designs, the designers then continue to modify the control parameters. The result of design number: 124 is intuitively selected by the designers. The designers would like to explore other types of product form designs with larger distance between the lens and the flash. Therefore, the designers select the third control strategy and set the configuration weighting factor to be 1.

- At 400 generations, all the designs are generated based on the third control strategy. Still, most of the designs get poor volume fitness values. However, the designers favour most of the designs in this generation.

- After evaluation of the designs, the designers then continue to modify the control parameters. The result of design number: 51 is particularly attractive to the designers. Although the flash is so close to the lens, the designers select this design by their own accord. The designers look for better designs by setting the second control strategy and assigning the volume weighting factor to be 10 for the next 100 generations.

- At 500 generations, the result of design number: 418 satisfies the designers. Even though some minor requirements are still not satisfied, the designers could continue to explore better designs by better understanding the complex effects provided by the modification of control parameters.

The assumption that better designs could be obtained is made provided that the requirements have to be refined considerably. If there are conflicting requirements, the designers should report those conflicting criteria to the relevant professionals and ask the professionals to consider modifying the requirements if necessary. All these procedures have been specified in section 1.5.2.

Detailed specifications	Constraint Index	Volume (cm^3)	GP groups	GA groups	SG groups
Generation: 1					
1. Design number: 494 **2.** Control strategy: 3 **3.** Main objective index (Overall fitness): 0.00 **4.** Objective index 1 (Artificial selection fitness): 0.00, Selection weighting factor: 1.00, Selection index value: 0.00 **5.** Objective index 2 (Configuration fitness): 0.00, Configuration weighting factor: 0.10, Configuration index value: 0.00 **6.** Objective index 3 (Volume fitness): 0.00, Volume weighting factor: 1.00, Volume index value: 0.00	**7.** Constraint index 1 (Configuration constraint): 0.0000, Configuration constraint weighting factor: 1.00, Configuration constraint index value: 0.0000	**8.** Shell volume: 28.4, Target volume: 24.5	**9.** GP group 1 (GP1 to GP8): + * x + x + x + **10.** GP group 2 (GP9 to GP12): x + - x **11.** GP group 3 (GP13): x	**12.** GA group 1 (GA1 to GA8): 34 44 -68 63 -75 -1 8 -1 **13.** GA group 2 (GA9 to GA12): 39 31 49 31 **14.** GA group 3 (GA13): 5	**15.** SG group 1: (a1x, a1y, a3x, a3y, a4x, a4y): 35 46 24 34 25 33 **16.** SG group 1: (b1x, b1y, b4x, b4y, b5x, b5y): -68 64 -64 51 -57 56 **17.** SG group 1: (c1x, c1y, c3x, c3y, c4x, c4y): -75 0 -65 7 -66 8 **18.** SG group 1: (d1x, d1y, d2x, d2y, d3x, d3y): 8 0 -1 9 -2 8 **19.** SG group 2: (Lens and Flash: x1, z1, x2, z2): 39 31 49 31 **20.** SG group 3: (Decorative feature:): DF5
Generation: 100					
1. Design number: 456 **2.** Control strategy: 3 **3.** Main objective index (Overall fitness): 3.43 **4.** Objective index 1 (Artificial selection fitness): 0.00, Selection weighting factor: 10.00, Selection index value: 0.00 **5.** Objective index 2 (Configuration fitness): 3.24, Configuration weighting factor: 0.10, Configuration index value: 32.38 **6.** Objective index 3 (Volume fitness): 0.19471467, Volume weighting factor: 1.00, Volume index value: 0.19471467	**7.** Constraint index 1 (Configuration constraint): 0.0000, Configuration constraint weighting factor: 10.00, Configuration constraint index value: 0.0000	**8.** Shell volume: 20.3, Target volume: 24.5	**9.** GP group 1 (GP1 to GP8): - - x x - - x x **10.** GP group 2 (GP9 to GP12): x **11.** GP group 3 (GP13): NIL	**12.** GA group 1 (GA1 to GA8): 13 58 -75 58 -74 1 12 0 **13.** GA group 2 (GA9 to GA12): 28 26 57 40 **14.** GA group 3 (GA13): 3	**15.** SG group 1: (a1x, a1y, a3x, a3y, a4x, a4y): 12 57 1 45 2 44 **16.** SG group 1: (b1x, b1y, b4x, b4y, b5x, b5y): -75 58 -70 45 -64 50 **17.** SG group 1: (c1x, c1y, c3x, c3y, c4x, c4y): -75 0 -65 7 -66 8 **18.** SG group 1: (d1x, d1y, d2x, d2y, d3x, d3y): 12 0 3 9 2 8 **19.** SG group 2: (Lens and Flash x1, z1, x2, z2): 28 26 57 40 **20.** SG group 3: (Decorative feature:): DF3
Generation: 200					
1. Design number: 299 **2.** Control strategy: 3 **3.** Main objective index (Overall fitness): 3.05 **4.** Objective index 1 (Artificial selection fitness): 0.00, Selection weighting factor: 10.00, Selection index value: 0.00 **5.** Objective index 2 (Configuration fitness): 2.63, Configuration weighting factor: 0.10, Configuration index value: 26.25 **6.** Objective index 3 (Volume fitness): 0.42636650, Volume weighting factor: 1.00, Volume index value: 0.42636650	**7.** Constraint index 1 (Configuration constraint): 0.0000, Configuration constraint weighting factor: 10.00, Configuration constraint index value: 0.0000	**8.** Shell volume: 23.1, Target volume: 24.5	**9.** GP group 1 (GP1 to GP8): x **10.** GP group 2 (GP9 to GP12): **11.** GP group 3 (GP13): NIL	**12.** GA group 1 (GA1 to GA8): 9 61 -86 61 -87 1 8 1 **13.** GA group 2 (GA9 to GA12): 36 26 58 40 **14.** GA group 3 (GA13): 5	**15.** SG group 1: (a1x, a1y, a3x, a3y, a4x, a4y): 9 61 -2 49 -1 48 **16.** SG group 1: (b1x, b1y, b4x, b4y, b5x, b5y): -86 61 -82 48 -75 53 **17.** SG group 1: (c1x, c1y, c3x, c3y, c4x, c4y): -87 1 -77 8 -78 9 **18.** SG group 1: (d1x, d1y, d2x, d2y, d3x, d3y): 8 1 -1 10 -2 9 **19.** SG group 2: (Lens and Flash x1, z1, x2, z2): 36 26 58 40 **20.** SG group 3: (Decorative feature:): DF5

Table 6.5a Implementation results of the evolved parameters of SG

Detailed specifications	**Constraint Index**	**Volume (cm^3)**	**GP groups**	**GA groups**	**SG groups**
Generation: 300					
1. Design number: 124 **2.** Control strategy: 2 **3.** Main objective index (Overall fitness): -28.48 **4.** Objective index 1 (Artificial selection fitness): 0.00, Selection weighting factor: 10.00, Selection index value: 0.00 **5.** Objective index 2 (Configuration fitness): 2.07, Configuration weighting factor: 0.10, Configuration index value: 20.68 **6.** Objective index 3 (Volume fitness): 0.14281634, Volume weighting factor: 1.00, Volume index value: 0.14281634	**7.** Constraint index 1 (Configuration constraint): -30.6950, Configuration constraint weighting factor: 10.00, Configuration constraint index value: -3.0695	**8.** Shell volume: 21.0, Target volume: 27.0	**9.** GP group 1 (GP1 to GP8): + - x * x * - x **10.** GP group 2 (GP9 to GP12): x x + x **11.** GP group 3 (GP13): x	**12.** GA group 1 (GA1 to GA8): 2 58 -87 55 -88 -2 4 0 **13.** GA group 2 (GA9 to GA12): 35 26 50 40 **14.** GA group 3 (GA13): 5	**15.** SG group 1: (a1x, a1y, a3x, a3y, a4x, a4y): 3 57 -8 45 -7 44 **16.** SG group 1: (b1x, b1y, b4x, b4y, b5x, b5y): -87 57 -83 44 -76 49 **17.** SG group 1: (c1x, c1y, c3x, c3y, c4x, c4y): -88 0 -78 7 -79 8 **18.** SG group 1: (d1x, d1y, d2x, d2y, d3x, d3y): 3 0 -6 9 -7 8 **19.** SG group 2: (Lens and Flash: x1, z1, x2, z2): 35 26 50 40 **20.** SG group 3: (Decorative feature:): DF5
Generation: 400					
1. Design number: 51 **2.** Control strategy: 3 **3.** Main objective index (Overall fitness): -25.57 **4.** Objective index 1 (Artificial selection fitness): 0.00, Selection weighting factor: 10.00, Selection index value: 0.00 **5.** Objective index 2 (Configuration fitness): 19.26, Configuration weighting factor: 1.00, Configuration index value: 19.26 **6.** Objective index 3 (Volume fitness): 0.05755382, Volume weighting factor: 1.00, Volume index value: 0.05755382	**7.** Constraint index 1 (Configuration constraint): -44.8908, Configuration constraint weighting factor: 10.00, Configuration constraint index value: -4.4891	**8.** Shell volume: 10.6, Target volume: 27.0	**9.** GP group 1 (GP1 to GP8): x **10.** GP group 2 (GP9 to GP12): NIL **11.** GP group 3 (GP13): NIL	**12.** GA group 1 (GA1 to GA8): 12 61 -75 60 -76 1 12 1 **13.** GA group 2 (GA9 to GA12): 42 26 55 40 **14.** GA group 3 (GA13): 1	**15.** SG group 1: (a1x, a1y, a3x, a3y, a4x, a4y): 12 61 1 49 2 48 **16.** SG group 1: (b1x, b1y, b4x, b4y, b5x, b5y): -75 60 -71 47 -64 52 **17.** SG group 1: (c1x, c1y, c3x, c3y, c4x, c4y): -76 1 -66 8 -67 9 **18.** SG group 1: (d1x, d1y, d2x, d2y, d3x, d3y): 12 1 3 10 2 9 **19.** SG group 2: (Lens and Flash: x1, z1, x2, z2): 42 26 55 40 **20.** SG group 3: (Decorative feature:): DF1
Generation: 500					
1. Design number: 418 **2.** Control strategy: 2 **3.** Main objective index (Overall fitness): -8.18 **4.** Objective index 1 (Artificial selection fitness): 0.00, Selection weighting factor: 10.00, Selection index value: 0.00 **5.** Objective index 2 (Configuration fitness): 20.68, Configuration weighting factor: 1.00, Configuration index value: 20.68 **6.** Objective index 3 (Volume fitness): 1.83870849, Volume weighting factor: 10.00, Volume index value: 0.18387085	**7.** Constraint index 1 (Configuration constraint): -30.6950, Configuration constraint weighting factor: 10.00, Configuration constraint index value: -3.0695	**8.** Shell volume: 22.6, Target volume: 27.0	**9.** GP group 1 (GP1 to GP8): + x * x x **10.** GP group 2 (GP9 to GP12): NIL **11.** GP group 3 (GP13): NIL	**12.** GA group 1 (GA1 to GA8): 6 59 -89 59 -88 0 6 0 **13.** GA group 2 (GA9 to GA12): 35 26 50 40 **14.** GA group 3 (GA13): 5	**15.** SG group 1: (a1x, a1y, a3x, a3y, a4x, a4y): 7 59 -4 47 -3 46 **16.** SG group 1: (b1x, b1y, b4x, b4y, b5x, b5y): -87 59 -83 46 -76 51 **17.** SG group 1: (c1x, c1y, c3x, c3y, c4x, c4y): -88 0 -78 7 -79 8 **18.** SG group 1: (d1x, d1y, d2x, d2y, d3x, d3y): 6 0 -3 9 -4 8 **19.** SG group 2: (Lens and Flash: x1, z1, x2, z2): 35 26 50 40 **20.** SG group 3: (Decorative feature:): DF5

Table 6.5b Implementation results of the evolved parameters of SG

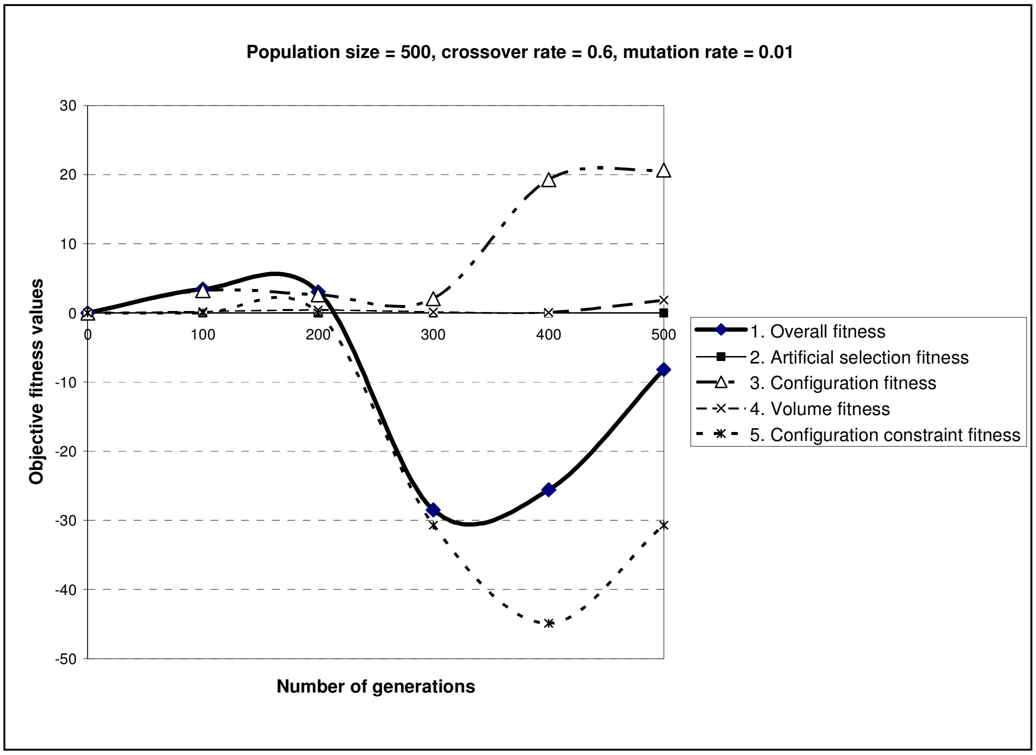


Figure 6.16 Variation of objective fitness values with different weighting factors and control strategies

The control parameters of the GP groups can be adjusted within appropriate ranges. The larger the range of control parameters as defined, the higher the modification rates to the designs will be. As a result, more dramatic modifications to the designs appear to result from erratic forms generation. Balance on the rate of modification and the quality of the generated models should be achieved by adjusting appropriate ranges for the control parameters.

Figure 6.16 shows the variation of objective fitness values with different weighting factors and control strategies. The data are taken from the selected designs in each periodically observed generation. The purpose of this diagram is to record the whole design process historically starting from the first generation to the last of the design process. Since the nature of a real design process is iterative and changes from each periodically observed generation, the diagram can be further analysed to enhance human computer interaction rather than used for purely computational analysis.

6.7 Summary

This chapter has described the development of the second prototype system which aims to enhance the performance of the first prototype. The second prototype system further advances the systematic approach in deriving SG rules and generating new designs. The second prototype system solves the problems and weaknesses associated with the first prototype system. Certain new key features of the second prototype system are highlighted which include relevant technological details of the implementation of the system. The main points are as follows:

- There are four main new key features of the second prototype system which have been developed for technological advancement in deriving SG. First, the new parametric 3D SG with labels has been developed in the second prototype system and tested for the exploration of free form designs. Second, the new genetic representation scheme called "GP-GA-SG" interface of phenotypes and genotypes has been developed to utilise the power of genetic and SG representation. Third, the new manipulation methods called "Control Strategies" have been developed to manipulate the new genetic representation and systematically evaluate the evolving designs during the evolutionary process. Finally, the new control mechanism which integrates the power of the control strategies and multi-objective functions has been developed to systematically evaluate the evolving designs during the evolutionary process. With all these new key features developed in the second prototype system, the complex effects produced by the SG rules modification can be fully explored and analysed during the evolutionary process.

- In the implementation stage, the relevant technological details of the implementation of the second prototype system have been described and presented. These include the descriptions of the initial settings of the system and the demonstration of the operations of the system as well as the interactions among the designers and the system. For example, the designers can evaluate the generated designs visually in a 3D virtual environment at each generation of the evolutionary process. In this way, the designers provide their preferences by selecting their favourite design for the system interactively.

- The first control strategy, which is specifically derived to regulate the evolving designs and their corresponding SG rules to obtain regular or symmetric properties, is highlighted and tested. The first experiment applies the first control strategy to generate new designs with regular design characteristics. The generated designs are evaluated with respect to different design requirements and the analysed results are listed in a tabular format.

This demonstration has shown that it is possible to strategically control the evolving SG rules to generate new designs with particularly desired design characteristics.

- The second experiment applies different control strategies to generate new designs with different design characteristics. By evaluating the implementation results visually and numerically, a clear picture of the complex effects produced by the multi-objective functions and control strategies is depicted. Based on the numerical analysis of the implementation results, designers can select appropriate control parameters of the SG rules and control strategies to explore new designs during the evolutionary design process. Graphical illustration of the analysis of implementation results is also provided which assists designers in understanding the effects of the interactions among both designers and system.

In summary, the second prototype presented in this chapter has demonstrated the newly developed integrated genetic and SG representation. Control strategies have been developed to monitor the parameters of the SG through new representations during the evolutionary process. The system has been described and evaluated with real examples of designing the forms of digital cameras.

Part V Conclusions

Part five (Conclusions) consists of one concluding chapter which presents conclusions and future work. Chapter 7 (Conclusions and Future Work) concludes with the findings from this research and identifies the key contributions made from the findings. Possible directions for future research in this area and several related issues are briefly discussed.

Chapter 7

Conclusions and Future Work

7.1 Revisiting the Objectives

This thesis has developed a framework to integrate evolutionary computing techniques with shape grammars, in order to combine the power of the two technologies to provide better support to product design process. The application of this framework to product design as demonstrated by the two prototypes of the framework enhanced the efficacy in deriving new solutions in accelerated speed for generating and evaluating a variety of stylistically consistent designs as defined by the shape grammars. Another significant advantage of this framework is that large varieties of valid designs with similar styles can be generated with the shape grammars, and further enhanced by the evolutionary systems. The advantages of using this integrated framework to produce design required significant work to plan and detail relevant design information. However, there are several key problems affecting the efficient use of a shape grammar based approach to support product design. Without addressing these key problems, the generative capability of the developed framework will be hindered. In this concluding chapter, the original objectives for the research are revisited in order to clearly state how these objectives are achieved by providing solutions to these key problems.

7.1.1 Key Problems

The key problems of applying a shape grammar based approach to product design identified in this thesis mainly include, how to derive shape grammars for a product with complex features and forms, and how to refine the shape grammars and extend the knowledge base to support new designs with different design requirements. These problems concern themselves with a fundamental issue of designing, that is, how to define a design space that inherits the features of the existing products or design ideas, but with constrained solution space for new designs.

For the first issue (problems in deriving the shape grammars), the first step is to develop a systematic method for deriving shape grammar rules for the generation of designs which fulfil specific design requirements. For this issue, neither theories nor empirical skills from the experts can be easily obtained. It is a time consuming process to derive grammars generic enough for all kinds of products by means of case studies. In the meantime, the practical skills of the experts are qualitative in nature and therefore hard to quantify for computational purposes. Therefore, most of the shape grammar rules have to be derived through analysis of existing designs.

However, some existing designs are stylistically inconsistent and therefore have diversified features, which in turn have discrete geometric attributes with complex spatial constraints. If these stylistically inconsistent designs are used for analysis, then they result in configuration conflicts and increase the complexity of the shape grammars. Besides, most of the features of the existing designs have different evaluation criteria from different perspectives. For example, a product designed with round fillet features around its exterior is expensive to make but has better aesthetic visual appeal. This makes it hard to find consensus on how features should be encoded in the shape grammar rules from an existing product without contextual design information other than geometric or spatial features.

For the second issue (problems in refining shape grammars and extending the knowledge base of shape grammars for new design), it is necessary to introduce flexibilities in the shape grammars. Normally, a specific shape grammar is limited to generating the designs within a confined design solution space. This is because a specific shape grammar is usually derived to solve one particular design problem. The knowledge encapsulated in the specific shape grammars is often fixed. However, if there are variations in design requirements, then these specifically derived shape grammars cannot generate satisfactory solutions. Therefore, it is necessary to derive a systematic approach which is capable of refining the shape grammars and extending the knowledge base such that the specific shape grammars can be modified to adapt to a new situation to solve new and more generic design problems. Therefore, a key objective is to develop a combined parametric 3D shape grammar generator and interpreter such that new rules can be explored and refined in a generative product design support environment.

7.1.2 Integration of Shape Grammars and Evolutionary Algorithms

For these reasons, the formulation of the complex development process of the shape grammars and the efficient manipulation of them becomes an inevitable necessity. It is an objective of this thesis that the development of a computational framework which facilitates the use of design information, and utilises the shape grammar based techniques as well as designers' knowledge to develop the shape grammars should ultimately support an important step towards the automation of the shape grammar development and generation processes in the future. This requires considerable research on the approaches to articulate a design strategy to foster quality development of shape grammars through a systematic plan in organising the design information and the creation of the computational framework to automate that design information to derive alternative shape grammar rules and generating new designs.

With these objectives, this research focuses on utilising the power of shape grammar and evolutionary computing to evolve shape grammar rules and derive new rules by the integration of an evolutionary architecture within the implemented IGBDS, targeted mainly for product design applications. In summary, the developed evolutionary IGBDS addressed the following problems, as specified in Chapter one:

- utilizing the power of genetic and shape grammar representation in the evolutionary shape grammar framework,
- introducing control strategies for evolving a set of stylistically inconsistent shape grammars which are gradually modified to generate stylistically consistent designs,
- resolving configuration conflicts for different features by constraining the maximum boundaries of the features and embedding collision avoidance criteria in the objective functions,
- adopting weighting methods for determining the evaluation values of designs,
- developing multi-objective functions for evaluation from many different perspectives, and
- allowing designers to alter the existing sets of shape grammar rules by modifying the control parameters of the objective functions and selecting appropriated control strategies.

To achieve the above objectives, this research addresses the key problems and issues in the development of a systematic approach and a comprehensive computational framework. The systematic approach describes a set of main tasks to be performed by the designers. The tasks are specified with the procedures for the development of shape grammars and the evolutionary architectures. For example, one task is to construct an information network of shape grammars, and its corresponding procedure is to define how the relevant design information should be organised. Once the shape grammars and evolutionary architecture are developed, these two computational techniques can be integrated into an evolutionary IGBDS. The evolutionary IGBDS consists of two main subsystems. These are the knowledge base of shape grammars for designs and the evolutionary computing architecture as a generative tool.

7.2 Revisiting the Research Methodologies

The solution insight to use the systematic approach in evolving SG in order to solve the problems in the development of SG did not come at the very beginning of this research for advancing the SG approach to product design. It has gone through five research stages non-linearly and repeatedly, and has touched every aspect of the activities as described in the above sections before the conceptual ideas became solidified. The five main stages of the research process were: 1) Preparation (Problem Finding), 2) Incubation, 3) Insight, 4) Evaluation and 5) Elaboration. The research process gone through in this research can be compared and referenced to descriptions by Nunamaker et al. (1991). Nunamaker et al. (1991) describe a general research process which consists of five key research stages: 1) Constructing a conceptual framework, 2) Developing a system architecture, 3) Analyzing and designing the system, 4) Building the (prototype) system, and 5) Observing and evaluating the system. The systematic approach encompasses four elements: 1) The information network of SG, 2) Core Variant Model, 3) Parametric SG, and 4) Evolutionary architecture.

7.2.1 Revisiting the Research Proposition

The primary goal of this research proposition is to explore a systematic approach to develop an integrated SG and evolutionary algorithm framework. This framework can be used to derive alternative or new SG rules to generate new designs. The insight from this research idea is cultivated from observation of two important facts: 1) Understanding the use of SG rules to explore design is a way to understand design process, and 2) It is of the same importance to understand the reasons and methods in deriving the SG rules. The systematic approach is therefore established to strategically apply all relevant design information with a clear understanding on why and how to apply such information to develop high quality SG and the evolutionary architecture. Once the evolutionary IGBDS is constructed with the SG and the evolutionary architecture, the designers can participate in the design process throughout the system. The advantages of this approach include integrating the designers' experience in the evolutionary IGBDS to evolve SG rules to generate new designs.

7.2.2 Organising Relevant Information

At the beginning of this research, a system application of water tap design using a Core Variant Model approach has been reviewed (Lee et al., 2000). With this case study, the problems and difficulties arising from processing design information necessary for product designs were identified. These problems include organising the complex features of components which are tightly coupled together to perform specific functions. After a long incubation period, an insight has been gained into explicitly specify the conditions for the control of the parameters for the complex features of the components.

After this review process, the development of an information network of SG was identified as a fundamental issue for organisation of design information related to the development of SG. The design information includes the abstracted forms of design requirements as well as the interrelationships among shapes and all other related attributes. Further elaboration of the ways to use such an information network of SG followed. Since the information contained in the information network of SG is excessive beyond what is absolutely required for the construction of SG for specific applications, a Core Variant Model was used to abstract the necessary information from the information network of SG to effectively access the relevant design information. Another issue in constructing the Core Variant Model was the methodologies to explicitly control of the parameters of the shapes in the SG rules.

In constructing the Core Variant Model, the systematic approach defined methodologies to organise all the relevant design information with clear understanding of why and how to apply such information to derive SG rules. This includes understanding the interrelationships among each component of the products. Once the model was built, it should be capable of identifying and analysing the design information associated with complex form creation. Based on the analysis results, this model can be used for the classification of product components, the definition of design spaces in component configuration, the specification of design constraints, the spatial relationships among components, and the explicit specification of the conditions in reasoning shapes. As a result, this model can be used by the IGBDS in systematically analysing the generated designs and organising the analysed results in several ways for further processing in the development of SG.

7.2.3 Development of Shape Grammars

The third element of the systematic approach was to develop SG based on the information specified in the Core Variant Model. The development of SG should be considered with the relevant issues in specific applications such as what types of products are selected to be designed, what algebras of SG are required, and what requirements of the designs are specified. The basic strategies of the systematic approach in developing SG are summarised as follows:

- to classify the components of existing designs from different perspectives,
- to define the maximum and minimum boundaries of geometry for each component,

- to specify the constraints of spatial relationships between any two components, and

- to identify evaluation criteria from different perspectives.

When there are existing SG which have been developed for specific applications, the development time in deriving new SG for new requirements can be reduced. The new SG can be derived by modifying the existing SG and therefore avoiding deriving the SG from scratch. This leads to the investigation of the methodologies to redefine the specific SG as illustrated in the experimental case study for generic pattern designs (see Chapter two). The solution insight was sought from two directions: 1) From the technical point of view to seek out all the possibilities in redefining the specific SG which included the modification of the parameters of shapes and the shapes themselves in the SG rules, the control of rule sequences and etc., and 2) From the analytical point of view to redefine the SG from many different perspectives.

A) Classification of Shape Grammars

In this thesis, two types of design activities in product design domain are classified as: 1) The generation of components and 2) The configuration of the generated components in an assembly. The SG rules can be classified based on these two types of design activities: Construction and Configuration SG rules. The Construction SG rules are used for the generation of each component whereas the Configuration SG rules are used for the allocation of components in an assembly. The Construction and Configuration SG rules are defined by encoding the knowledge described in handbooks, literatures, analysis results of the existing designs, the manuals and specifications for a particular type of products, and or results obtained from interview with design experts. After all the components are generated by the Construction SG rules, the components are configured in an assembly using the Configuration SG rules.

The Construction SG rules can be classified into different groups in accordance to the functional decomposition of the components or geometric properties. Labels are used to control the execution orders of the SG rules. The development of the Construction SG rules can further be divided into two types: 2D and 3D SG. The reasons for developing these two types of SG are that most of the components are standardised for ease of industrial manufacture. Some form features of the components can therefore be generated by standard mechanical methods such as extrusion and sweeping of 2D profiles. On the other hand, some form features require 3D free forms.

The formulation of Configuration SG rules is based on the typical arrangement of components located at the main body. The Configuration SG rules use the boundaries of the designs as configuration constraints for the allocation of the components, and use labels to maintain proper generation sequence.

B) Parametric 2D Shape Grammars

Parametric 2D and 3D SG with labels have been developed in this research for applications in the product design domain. Each of these two SG can be applied individually or together for applications in specific situations. A set of parametric 2D SG rules with labels can be derived with reference to the Core Variant Model. The SG rules are derived to generate the 2D geometric profiles of components. The 2D profiles of components are chosen because they can be further manipulated with different methods such as coiling, extrusion, lofting, revolving and sweeping to create three-dimensional (3D) objects. For example, after a 2D profile is created by the application of the Construction SG rules, it is then extruded with a thickness to form a 3D object. For the generation of a curved 3D object, lofting operations between two profiles are required. Boolean operations on the 3D objects are also required to generate more complex geometric features of the components. In this way, the implementation time can be reduced in generating the form features of the components while longer implementation time is required for the generation of the free form exterior main body.

C) Parametric 3D Shape Grammars

In the development of parametric 3D SG with labels, the SG rules are derived to generate the 3D geometric profiles of components. Parametric 3D SG with labels is developed for the generation of product forms which comprise common engineering shapes. The common engineering shapes are the vocabularies of SG. The common engineering shapes can be classified by their geometric properties like free-form shapes and primitive shapes such as blocks, cylinders, cones, spheres, torus, etc. and their combinations. Non-uniform rational B-spline (NURBS) surfaces are constructed to represent free-form objects in a virtual 3D spatial environment. Labels are used to associate with the control points of the NURBS surfaces and primitive shapes, and the design objects. The labelled control points are used for the identification of NURBS surfaces and primitive shapes. Labels of the design objects are used as functional symbolic notations for the control of generation sequence. Both the labelled control points and the labels of design objects have values to specify their XYZ geometric coordinates.

7.2.4 Evolutionary Architecture Integration

The knowledge base of the information network of SG must be evolved and updated to allow new knowledge to be input to the network. The main reason for this requirement is to extend the generative capability of SG, so that new SG can be derived to generate new designs for new design requirements. However, difficulties exist in adding new knowledge to expand the existing knowledge base of the information network of SG. This is due to the huge amount of work needed to prepare revisions for the information network of SG. An insight to develop a strategic plan to add new knowledge to the existing information network of SG, and to make that knowledge fully effective for future usage was developed. The strategic plan suggested that an evolutionary architecture needed to be integrated with an IGBDS to evolve alternative or new SG rules.

According to this strategic plan, the fourth element of the systematic approach is therefore defined to develop an evolutionary architecture and integrate it to an IGBDS. The purposes of the integration of an evolutionary architecture to IGBDS are to enhance the generative capability of the SG, and to automate some of the development processes in the systematic approach. The evolutionary architecture was developed to explore new SG rules which in turn generate new designs for new design requirements. This required substantial investigation on all related issues in the modification of the SG rules. This exploitation process of SG rules directly influenced the generative capability of IGBDS. As a result, the importance and necessity to enhance the generative capability of IGBDS was emphasised in the exploration of designs. Two evolutionary architectures were developed in this research with the first using standard GA as the core evolutionary algorithm and the second using GP.

The development of an evolutionary architecture for an IGBDS involved several technical problems which included issues such as the integration of two types of computational techniques: evolutionary computing and shape grammars, the representation issue in utilising the power of genetic and parametric 2D and 3D SG representations with labels, and the manipulation in evolving the SG rules.

A) The First Prototype System

Based on this strategic plan, a software prototype (the first prototype system) was built and tested to validate the theoretical approach to integrate an evolutionary architecture to an IGBDS to facilitate the development of SG. The first prototype system provided a new paradigm to enhance the generative capability of SG. The key feature of this system was to allow designers to participate in the development process of SG. The designers can subjectively choose their favourite generated designs during the evolutionary design process. In particular, the generated results were evaluated by convergent and divergent testing to assure that the system has the optimisation and exploration capabilities. This was achieved by adjusting the control parameters of the evolutionary architecture like mutation and crossover rates. After the evaluation of the results with different testing plans in modifying the control parameters of the SG rules, the results were found satisfactory.

B) The Second Prototype System

However, a further elaboration of the first prototype system indicated that the generative capability of the system was limited due to the parametric 2D SG and the problems of random modification to stylistically consistent designs generated by the SG. These two technical problems hindered the real power of the generative capability of the evolutionary IGBDS. Usually, insight for solutions is achieved at the time the problems are identified. Better solution insight for these two problems was gained by discussions among peers, reviewing the relevant historical and updated literatures, experimentally testing the systems and through discussions with designers.

After a reasonable incubation period, the objectives were defined to tackle these two technical problems: 1) Deriving 3D parametric SG for free form generation, 2) Utilising the power of genetic and SG representations, and 3) Developing control strategies for the control of modification of the stylistically consistent designs. The second prototype system was then developed based on these strategies. In order to further strengthen the capability of the second prototype system, multi-objective functions were used to evaluate the generated designs and the corresponding SG rules from many different perspectives. The results were evaluated and found satisfactory to fulfil different design requirements.

For the first objective related to solving the problems of free form designs, the parametric 3D SG rules with labels were created with 3D labelled shapes. In developing the 3D SG, a set of existing products in a particular product design domain was first analysed to derive shape features in the form of SG rules using the systematic approach.

For the second objective related to solving the problems of the representation issues for the integrated SG and evolutionary algorithm approach, a new representation was developed to utilise the power of genetic and parametric 2D and 3D SG representations with labels. The new representation named “GP-GA-SG” was developed and illustrated in the second prototype system. An evolutionary algorithm was applied to evolve the genetic SG rules for the generation of new designs. Both product component designs as well as product configurations were supported in this second prototype system.

For the third objective related to solving the problems of random modification on stylistically consistent designs, control strategies were developed. The control strategies should therefore control the modification on shapes during the evolutionary process and be capable of finer control of styling modification. Multi-objective functions were also developed to cooperate with the control strategies to direct the evolving SG rules to obtain specific design characteristics. The SG rules can therefore be evolved to generate new designs to fulfil new requirements and constraints.

7.3 Summary of the Key Research Activities

The aims of the development of SG were to build a "visual language of design" for designers to understand, reason, generate and interpret designs visually within that language. As the cultivation of the SG languages were progressively generated and revised during the SG generation process, the evolving designs appeared in the languages at a particular state reflecting their current situations. These situations included the current state of understanding by the designers of how and why the designs were derived and generated in the ways specified in the SG rules. Not only can these two activities: "understanding the reasons of derivation and generation of designs by the SG" be reflected at the current state of the evolving designs, but more important activities which go beyond what the SG rules can express, "the reasoning and interpretation of designs by the designers" are also reflected in the current situations.

From the micro point of view, these two activities: "reasoning and interpretation of designs" can only allow designers to make decisions on the choices of particular SG rules or modification of the parameters of the selected SG rules for the next state of SG generation process. However, all decisions have to be made within the current framework of SG. From the macro point of view, the "reasoning and interpretation of designs" can inspire designers to investigate new perspectives in analysing designs, new ways to express designs in SG, new response manners to evaluate designs and new paradigms to redevelop the whole SG framework. The best way to illustrate this observation is to review the key research activities and theoretical issues which were addressed during the research process.

This research developed two prototype systems. Much of research work focused on the development of parametric SG using labels and the evolutionary architecture. From the micro point of view, each prototype system went through all the five main research stages. From the macro point of view, the five main research stages as stated in section 7.2 were gone through separately among the two prototype systems.

For the first stage (Preparation), the problems of the development of SG have been identified. To tackle these problems, the concepts of an information network of SG and the Core Variant Model were initialised as solutions to manage all the relevant information necessary for the development of SG.

For the second stage (Incubation), the problems of specific SG which limit generation of a particular class of designs and for specific requirements were identified. The methodologies to identify all the possible modifiable elements and refine the SG from many different perspectives were developed and illustrated in an experimental case study for generic pattern designs. After a long incubation period, more critical issues related to the development of SG were sought.

For the third stage (Insight), the problems of the heavy loading to redefine the SG for more generic applications and the limitations in expanding the richness of knowledge in the information network of SG were identified. The methodologies to integrate an evolutionary architecture to an IGBDS to evolve new SG was developed and illustrated in the first prototype system for digital camera form designs.

For the fourth stage (Evaluation), the problems of the evaluation in using a single criterion with artificial selection technique and the limitation of the parametric 2D SG in generating free forms were identified in the first prototype system.

For the fifth stage (Elaboration), the overall performance of the first prototype system was analysed in detail and possible strategies to enhance the performance of the evolutionary IGBDS was considered. The methodologies in developing new representation to utilise the power of genetic and SG representations, new control strategies to control the random modification of stylistically consistent designs and multi-objectives functions to evaluate the generated designs were developed and illustrated in the second prototype system for digital camera form designs.

7.4 Contributions

The critical issues in using the SG approach to product design were addressed and discussed in this thesis. These issues included the development of usable SG relevant to particular applications, the representation and manipulation issues of SG, how to apply the SG in different situations, what elements of SG have to be controlled and which control strategies have to be applied in what conditions and in what manner, etc.

For the development of SG, the major tasks were to derive a set of SG rules which are useful for particular applications. The SG rules implicitly captured the knowledge of design and their quality is the key factor in determining the success of the SG applications. In the development process of SG, difficulties arose from acquisition of design knowledge either by deriving design theories or identifying empirical skills from experts.

One of the significant contributions in this research was to develop a systematic approach and the computational framework to develop the parametric SG. The SG developed was parametric in 2D and 3D, using labels to maintain proper function to form sequence and grouping individuals from many different perspectives such as functional or geometric points of views, and including algebras of shapes to be applied in designs. The first step of the systematic approach was to analyse the existing products to gain sufficient knowledge for the construction of SG. All the related design information such as geometric features, constraints and spatial relationships were identified from the analysis of the existing designs. The results obtained from the analysis were organised and represented in the information network of SG. A Core Variant Model was constructed by abstracting the necessary information from the information network of SG. A set of parametric SG rules with labels was then derived with reference to the Core Variant Model. The second step of the systematic approach was to apply the evolutionary algorithms to evolve new SG rules which are useful and of high quality, and can be used to generate new designs for specific design applications.

The main contributions of this research in addressing these issues are summarised as follows:

- The critical issues and problems of the formulation of the SG rules were identified in the domain of product design as the key factors to enhance the usefulness and quality of the SG. The application of useful and high quality SG enhances the performance of the IGBDS in which designers can apply the SG efficiently to develop new product designs.

- A systematic approach was developed that goes through every detail of the development of SG starting with the analysis of existing designs from many different perspectives, organising the analysis results and all the relevant information into an information network of SG, abstracting all the necessary information from the information network of SG into an Core Variant Model for particular design applications, explicitly specifying the conditions for the execution of SG rules for the complex features of components which are tightly coupled together to perform specific functions, and developing an evolutionary architecture for the IGBDS to evolve alternative SG. In this way, the systematic approach was developed to cover as extensively as possible all the technical and practical issues for the successful development of SG going beyond the standard SG approach.

- An evolutionary architecture was developed that is competent to derive alternative SG rules of high quality and can be used to generate new designs for specific design applications. The evolutionary architecture was developed with powerful tools capable of conquering the integration problems of random modification of stylistically consistent designs, utilising the power of genetic and SG representations in generating designs, applying control strategies for the control of manipulation of SG and applying multi-objective functions for the evaluation of the designs. This makes the evolutionary IGBDS more scalable than other general IGBDS.

- The parametric 2D and 3D SG were developed to facilitate the generation of product designs. The parametric SG was developed which is highly customisable and in detail for specific applications in the product design domain.

- Two prototype systems were developed which demonstrate that the systematic approach was capable of deriving high quality and useful SG and generating new designs which can fulfil different design requirements. The evolutionary process illustrated in the two prototype systems demonstrated that the evolutionary IGBDS was capable to evolve alternative SG rules to generate new designs in a controlled environment.

The systematic approach and the integrated SG and evolutionary algorithm framework can be targeted at a community whose primary goal is to develop the evolutionary IGBDS, referred to as SG developer or researchers. A community, whose primary goal is to apply the evolutionary IGBDS for the generation of specific types of new products, is referred to as designers. The ‘SG developer’, ‘researcher’ and ‘designer’ may be the same set of people. This work has been reported in international conferences (Lee and Tang, 2004, 2006) and is referenced by other researchers in the similar field (Ang et al., 2006).

7.5 Future Work

But still, further research can be conducted to enhance the efficacy and usability of the systematic approach for the development of SG in product design domain. Much of the discussions for the successful development of the SG applications can be classified into short-term and long-term perspectives.

7.5.1 Short Term

More comprehensive specifications of the systematic approach can be created by focusing on the key issues for specific applications in product design domain. The specifications define how the information network of SG, Core Variant Model and evolutionary architecture should be built for the development of SG. This elaboration process will highlight problems that may exist in the systematic approach for situations that are unique regarding specific applications in product design domain. The comprehensive specifications can precisely describe the requirements for the various models and evolutionary architecture in the systematic approach being further refined.

7.5.2 Long Term

A) Technical Issues

The parametric 2D and 3D SG with labels developed in this research are set grammars which are a subset of SG. For some applications in product and engineering design domains, they are sufficient to generate new designs to meet customers' needs. However, the performance of the integrated SG and evolutionary algorithm framework can be further enhanced by using the maximal representation of SG. This requires further research on how to utilise the ambiguity and emergent properties of SG in product design domain. A critical problem in this issue is that, in general product design applications, the product forms with long repeating elements of shapes are less common than general shapes. This limitation in turn hinders the power of the emergent property of SG as emergent shapes can appear in the repeating elements of shapes. The potential generative capability of the SG is therefore not fully utilized in product design domain.

A methodology should be derived to reason the generated designs and to capture the emergent forms generated from the SG. The systematic approach can be adapted to cope with such extensions of generative capability of SG since the systematic approach allows room for such extension. Although the relevant information for SG is well organised in the Core Variant Model, the level of abstraction for each design object can vary and the objects can merge during each revision by redefining the systematic approach. In this case, the emergent shapes can appear in specific components or in the overall design which depend on the specifications of the systematic approach.

Another extension of the generative capability of SG can focus on the development of 3D SG in algebra V_{23} , V_{33} or other types of 3D SG. More experiments can be conducted to test the non-deterministic 3D SG to gain insight on how to better refine the systematic approach for applications in product design domain.

All these extensions can be referred to as the final stage "Elaboration" of the research process. The last stage of "Elaboration" encompasses the observation and evaluation on the operations of the existing systematic approach. This elaboration process will result in new theories and models of how the systematic approach could be improved for the development of SG. As a result, new computational systems could be continuously developed in this cyclical research process and the quality of SG generated can be improved.

B) Application Issues

One potential area of future research is to provide background support for the development of SG. For example, the cultural elements should be studied: "In terms of product design, we should emphasise the uniqueness of eastern culture, such as implicitness, calmness, lightness, as well as the pursuit for natural and humanistic harmonies, in order to stimulate an emotional echo from the users." (Leung, 2006) The application of the systematic approach and the evolutionary IGBDS requires further elaboration of key aspects such as:

- embedding the elements of the culture in the SG in an elegant manner,
- understanding customer behaviours in choosing the products with their preferences, and
- exploring market trends in the development of new products for specific regions like China, USA and etc, or for global marketing.

Since the systematic approach developed in this research allows room for continuous refinement and further development, the "quality" and "usefulness" of the SG can be assured in the development of the SG with specific domain knowledge and designers' personal experience. In the long term, such a systematic approach and the integrated SG and evolutionary algorithm framework would fulfil the market's needs to generate new product designs.

7.6 Conclusions

In this thesis, two major design problems related to product design exploration are addressed: the first major problem relates to issues such as the balance between stylistic consideration and technical innovation, and the second major problem relates to the issue such as the control of product design exploration under multi-dimensional requirements. The balance of stylistic consideration and technical innovation is more easily attained with a system that has generative ability.

To address these two problems, an explanation of the term of product design exploration has to be sought in advance. The explanation has been given with the illustration of a picture of the product realization process. The exploration of designs is not only categorised as a problem solving activity but also as a problem finding activity. These two activities link with each other in an interactive design environment during the design process. A scenario is envisaged when describing the interactions among the professionals from various departments including engineers, designers and shape grammar developer, and the system within a unified framework of shape grammars.

A strategy is developed to address the issues related to the two problems for supporting design activities. The first part of the strategy describes the role of professionals in various departments of an enterprise, and the computational framework. The second part of the strategy applies a scenario to describe how the professionals and the computational framework can cooperate in an interactive design environment. The third part of the strategy addresses the issues of the construction of the computational framework.

The significance of deriving these three parts of the overall strategy is to help to explain more clearly how a computational framework can be developed for supporting product design exploration. The strategy brings out a model of exploration which works in an interactive way. With the term of interactivity and exploration, it is clearly defined what the system does, and what is the work done by the designers, how the two go together and when it is expected for the designers to stop or for the system to be stopped. With the illustration of the scenario, the strategy shows a clear picture of how the computational framework is used to support various design activities and benefit the professionals from various departments.

The significant advantages of this research include the enhancement of the generative capability of SG using a systematic approach in which an integrated SG and evolutionary algorithm framework for product design was created. This framework involved both broad theoretical issues and specific computational issues.

The theoretical issues related to the development process of SG, including analysing the existing designs, organising and representing related design information to build a knowledge base of shape grammars for form generation and configurations. The specific computational issues dealing with the implementation of a design system regarding the power of reasoning, generative and adaptive capability of SG and evolutionary computing, were utilised.

This thesis demonstrated how such a framework can be built within a 3D solid modelling environment to deal with form generation and configuration. In order to achieve this objective, sophisticated 2D and 3D reasoning methods were required to identify the key design parameters which represent the product design characteristics for a series of product forms. The reasoning methods were then integrated with evolutionary algorithms to generate alternative designs, in a generative and evolutionary design environment, in which designers can actively participate in the design process through interaction with the system. The feasibility of such a framework was validated via two different prototype systems with the examples of digital camera form design application. The two prototype systems generated useful initial form models which were directly manipulated and further explored in normal 3D solid modelling systems for fine tuning the design.

In this framework, designers play a major role in determining the quality of SG rules through the quality of the output. The system supports the evaluation of the output by providing numerical analysis and visualisation of complex 3D models. However, further investigations on designing a user-friendly interface is required in order to facilitate the designers to master the complexity of control parameters effectively.

In conclusion, the integration of shape grammars with evolutionary computing techniques helps the formulation of design knowledge from existing designs with parametric shape grammars. The application of the formulated parametric shape grammars in an interactive evolutionary system using genetic algorithms derives new shape grammar rules, which in turn can generate new designs outside the scope of the original shape grammars. This thesis provides a foundation with two prototype systems, and developed an analysis process, computational representations, control strategies, spatial reasoning algorithms, and implementation methods for fully exploring the potential of this framework in the future in a product design oriented environment which involves complex form generation and configuration optimisation.

Bibliography

Agarwal, M. and Cagan, J. (1998). A blend of different tastes: The language of coffeemakers. *Environment and Planning B: Planning and Design*, 25:205-226.

Agarwal, M., Cagan, J. and Constantine, K. G. (1999). Influencing generative design through continuous evaluation: Associating costs with the coffeemaker shape grammar. *Artificial Intelligence for Engineering Design, Analysis and Manufacturing (AIEDAM)*, 13:253-275.

Ang, M. C, Chau, H. H, McKay, A. and de Pennington, A. (2006). Combining evolutionary algorithms and shape grammars to generate branded product design. In *Design Computing and Cognition*, pages 521-539.

Bentley, P. J. (1996). *Generic Evolutionary Design of Solid Objects using a Genetic Algorithm*. Doctoral dissertation, Division of Computing and Control Systems, Department of Engineering, University of Huddersfield.

Bentley, P. J. (1998). Aspects of evolutionary design by computers. *Advances in Soft Computing Engineering Design and Manufacturing*, 99-118.

Bentley, P. J. (1999a). An introduction to evolutionary design by computers. In Bentley (1999c), chapter 1, pages 1–73.

Bentley, P. J. (1999b). From coffee tables to hospitals: Generic evolutionary design. In Bentley (1999c), chapter 18, pages 405–423.

Bentley, P. J., editor (1999c). *Evolutionary Design by Computers*. Morgan Kaufman Publishers, San Francisco, CA.

Bentley, P. J. and Wakefield, J. P. (1996a). Generic representation of solid geometry for genetic search. *Microcomputers in Civil Engineering*, 11(3):153-161

Bentley, P. J. and Wakefield, J. P. (1996b). An analysis of multi-objective optimisation within genetic algorithms. In Technical Report ENGPJB96, University of Huddersfield, UK.

Bralla, J. G. (1998). Design for manufacturability handbook, second edition. McGraw-Hill Handbooks; pages 1-17.

Brown, K. N. and Cagan, J. (1997). Optimized process planning by generative simulated annealing. *Artificial Intelligence for Engineering Design, Analysis and Manufacturing (AIEDAM)*, 11:219–235.

Brown, K. N., McMahon, C. A. and Sims, W. J. H. (1995). Features, aka the semantics of a formal language of manufacturing. *Research in Engineering Design*, 7:151–172.

Cagan, J. (2001). Engineering Shape Grammars: Where We Have Been and Where We Are Going. In Antonsson, E. K. and Cagan, J., editors, *Formal engineering design synthesis*, chapter 3, pages 65–92. Cambridge University Press 2001, UK.

Çağdaş, G. (1996). A shape grammar model for designing row-houses. *Design Studies*, 17: 35-51.

Caldas, L. (2001). *An Evolution-Based Generative Design System: Using Adaptation to Shape Architectural Form*. Doctoral dissertation, Massachusetts Institute of Technology.

Caldas, L. and Norford, L. (2001). Architectural constraints in a generative design system: Interpreting energy consumption levels. In *Proceedings of the Seventh International IBPSA Conference*, pages 1397–1404, Rio de Janeiro, Brazil.

Chan, C.-S. (1992). Exploring individual style through Wright's designs. *Journal of Architectural and Planning Research*, 9(3):207-238.

Chan, C.-S. (1995). A cognitive theory of style. *Environment and Planning B: Planning and Design*, 22(4):461-474.

Chan, K. H., Frazer, J. H. and Tang, M. X. (2000). Handling the Evolution and Hierarchy Nature of Designing in Computer-Based Design Support Systems. In *Proceedings of the Third International Conference on Computer-Aided Industrial Design and Conceptual Design (CAID&CD '2000)*, pages 447-454.

Chan, K. H., Frazer, J. H. and Tang, M. X. (2001). Interactive Evolutionary Design in a Hierarchical Manner. In *Generative Art 2001*, Milan, Italy.

Chan, K. H., Frazer, J. H. and Tang, M. X. (2002). An Evolutionary Framework for Enhancing Design. In *Artificial Intelligence in Design '02*, pages 383-403, Cambridge University, UK.

Chan, K. H., Lee, H. C., Fraser, J. H. and Tang, M. X. (1999a). A Hierarchical Design Interface for Collaborative Design. In *The Sixth International Conference on Concurrent Engineering*, pages 181-188, Technomic Publishing Co. Inc., Lancaster, England.

Chan, K. H., Lee, H. C., Frazer, J. H. and Tang, M. X. (1999b). A Hierarchical Design Interface for Integrated CAD Systems. In *The ICYCS99, The Fifth International Conference for Young Computer Scientists*, pages 830-835, International Academic Publisher, Nanjing, PRC.

Chase, S. C. (1989). Shapes and shape grammars: From mathematical model to computer implementation. *Environment and Planning B: Planning and Design*, 16:215-242.

Chase, S. C. (1996). Design modelling with shape algebras and formal logic. In *Design Computation: Collaboration, Reasoning, Pedagogy - Proceedings of ACADIA '96*, pages 99-113, Tucson, Arizona.

Chase, S. C. (2002). A model for user interaction in grammar-based design systems. *Automation in Construction*, 11:161-172.

Chien, S.-F., Donia, M., Synder, J. D. and Tsai, W.-J. (1998). SGCLIPS: A system to support the automatic generation of designs from grammars. In *Proceedings of the Third Conference On Computer Aided Architectural Design Research in Asia (CAADRIA '98)*, pages 445–454.

Coombs, S. and Davis L. (1987). Genetic algorithms and communication link speed design: Constraints and operators. In *Proceedings of The Second International Conference on Genetic Algorithms (ICGA '87)*, pages 257-260, Cambridge, MA, USA.

Davis, L. (1991). *Handbook of genetic algorithms*. Van Nostrand Reinhold, New York.

Davis, L. and Coombs, S. (1989). Optimizing network link sizes with genetic algorithms. In Elzas, M. S., Oren, T. I. and Zeigler, B. P., editors, *Modelling and Simulation Methodology*, Chapter IV(3), pages 317-331, Elsevier Science Publishers B. V., North-Holland.

Dawkins, R. (1983). Universal Darwinism. In Bendall, D. S., editor, *Evolutionary from Molecules to Men*, chapter 20, pages 403-425, Cambridge University Press.

Ding, L. and Gero, J. S. (2001). The emergence of the representation of style in design. *Environment and Planning B: Planning and Design*, 28(5):707–731.

Dixon, J. R. and Poli, C. (1995). *Engineering Design and Design for Manufacturing*, Field Stone Publishers, Conway, MA.

Dorst, K. (2006). Design Problems and Design Paradoxes. *Design Issues*, 22(3): 4-17.

Dorst, C. H. and Cross, N. G. (2001). Creativity in the Design Process: Co-evolution of Problem-solution. *Design Studies*, 22: 425–437.

Duarte, J. P, Ducla-Soares, G., Caldas, L. and Rocha, J. (2006). An urban grammar for the Medina of Marrakech: Towards a tool for urban design in Islamic contexts. In *Design Computing and Cognition*, pages 483-502.

Dym, C. L. and Levitt, R. E. (1991). *Knowledge-Based Systems in Engineering*. McGraw-Hill Inc.

Earl, C. F. (1999). Generated designs: Structure and composition. *Artificial Intelligence for Engineering Design, Analysis and Manufacturing (AIEDAM)*, 13:277-285.

Eggert, R. J. (2005). *Engineering Design*. Pearson Prentice Hall, Pearson Education, Inc. Upper Saddle River, NJ 07458, USA.

Flake, G. W. (1998). *The Computational Beauty of Nature: Computer Explorations of Fractals, Chaos, Complex Systems, and Adaptation.* The MIT Press, Cambridge, MA.

Fogel, D. B. (1988). An evolutionary approach to the traveling salesman problem. *Biological Cybernetics*, 60:139-144.

Fogel, L. J. (1963). *Biotechnology: Concepts and Applications*. Prentice Hall, Englewood Cliffs, NJ.

Fonseca, C. M. and Fleming，P. J. (1995). An overview of evolutionary algorithms in multi-objective optimization. *Evolutionary Computation*，3(1):1-16.

Fonseca, C. M. and Fleming，P. J. (1998). Multi-objective optimization and multiple constraint handling with evolutionary algorithms - Part I: A unified formulation. *IEEE Transactions on Systems, Man, and Cybernetics, Part A: Systems and Humans*, 28(1):26-37.

Frazer, J. H. (1992). Data structures for rule-based and genetic design. In *Proceedings of the Tenth International Conference of the Computer Graphics Society on Visual Computing: Integrating computer graphics with computer vision*, pages 731-744, New York, NY, Springer-Verlag.

Frazer, J. H. (1995). *An evolutionary architecture*. Architectural Association Publications, London, UK.

Frazer, J. H. (2002). Creative design and the generative evolutionary paradigm. In Bentley, P. J. and Corne, D. W., editors, *Creative Evolutionary Systems*, pages 253-274, Morgan Kaufmann Publishers.

Frazer, J. H., Tang, M. X. and Sun, J. (1999). Towards a generative system for intelligent design support. In *Proceedings of the Fourth Conference on Computer Aided Architectural Design Research in Asia (CAADRIA '99)*, pages 285–294, Shanghai Scientific and Technological Publishing House.

French, M. J. (1994). *Invention and Evolution: Design in Nature and Engineering*, Second Edition. Cambridge University Press.

Gero, J. S. (1992). Creativity, emergence and evolution in design. In Gero, J. S. and Sudweeks, F., editors, *Second International Round Table Conference on Computational Models of Creative Design*, pages 1-28, Department of Architectural and Design Science, University of Sydney.

Gero, J. S. and Ding, L. (1997). Exploring style emergence in architectural design. In *Proceedings of the Second Conference On Computer Aided Architectural Design Research in Asia (CAADRIA '97)*, pages 287–296.

Gero, J. S., Louis, S. J. and Kundu, S. (1994). Evolutionary learning of novel grammars for design improvement. *Artificial Intelligence for Engineering Design, Analysis and Manufacturing (AIEDAM)*, 8(2):83-94.

Gero, J. S. and Schnier, T. (1995). Evolving representations of design cases and their use in creative design. *in* J. S. Gero, M. L. Maher and F. Sudweeks (eds), *Preprints Computational Models of Creative Design* , Key Centre of Design Computing, University of Sydney, 343-368.

Gips, J. (1975). *Shape Grammars and Their Uses: Artificial Perception, Shape Generation and Computer Aesthetics*. Birkhäuser, Basel.

Gips, J. (1999). Computer implementation of shape grammars. Report, NSF/MIT Workshop on Shape Computation, Department of Architecture, School of Architecture and Planning, Massachusetts Institute of Technology.

Goldberg, D. E. (1989). *Genetic Algorithms in Search, Optimization and Machine Learning*. Addision-Wesley, Reading, MA.

Goldberg, D. E. (1991). Real-coded genetic algorithms, virtual alphabets and blocking. *Complex Systems*, 5:139-167.

Goldberg, D. E. (2002). *The design of innovation, lessons from and for competent genetic algorithms*. University of Illinois at Urbana-Champaign, Kluwer Academic Publishers.

Goldberg, D. E. and Kuo, C. H. (1987). Genetic algorithms in pipeline optimization. *Journal of Computing in Civil Engineering*, 1(2):128-141.

Graham, P. C., Frazer, J. H., and Hull, M. C. (1993). The application of genetic algorithms to design problems with ill-defined or conflicting criteria. In Glanville, R. and de Zeeuw, G., editors, *Proceedings of Conference on Values and, (In) Variants*, pages 61–75.

Grefenstette, J. J., Gopal, R., Rosmaita, B. J. and van Gucht, D. (1985). Genetic algorithms for the traveling salesman problem. In Grefenstette, J. J., editor, *Proceedings of the First International Conference on Genetic Algorithms and Their Applications*, pages 160-168, Carnegie-Mellon University.

Hinterding, R. and Khan, L. (1995). Genetic algorithms for cutting stock problems: With and without contiguity. In Yao, X., editor, *Progress in Evolutionary Computation*, volume 956 of *Lecture Notes in Artificial Intelligence*, pages 166-186, Springer-Verlag, Heidelberg, Germany.

Holland, J. H. (1975). *Adaptation in Natural and Artificial Systems*. University of Michigan Press, Ann Arbor.

Holland, J. H. (1992). Genetic algorithms. *Scientific American*, 66-72.

Horn, J. and Nafpliotis N. (1993). Multi-objective optimisation using the niched pareto genetic algorithm. In Technical Report 93005, Illinois Genetic Algorithms Laboratory (IlliGAL), University of Illinois at Urbana-Champaign.

Hsiao, S. W. and Chen, C. H. (1997). A Semantic and Shape Grammar Based Approach for Product Design. *Design Studies*, 18(3):275-296.

IDSA (Industrial Design Society of America). Definition of Industrial Design. http://www.idsa.org.

Janssen, P. (2004). *A design method and computational architecture for generating and evolving building designs*. Doctoral dissertation, School of Design, The Hong Kong Polytechnic University.

Janssen, P., Frazer, J. H. and Tang, M. X. (2002). Evolutionary Design Systems and Generative Processes. *The International Journal of Artificial Intelligence, Neural Networks, and Complex Problem-Solving Technologies*, 16(2):119–128.

Kanal, L. and Kumar, V. (1988). *Search in Artificial Intelligence*. Springer-Verlag, Berlin.

Kim, J.-H. and Myung, H. (1997). Evolutionary programming techniques for constrained optimization problems. *IEEE Transactions on Evolutionary Computation*, 1(2):129-140.

Knight, T. (1994a). *Transformations in Design: A formal approach to stylistic change and innovation in the visual arts*. Cambridge University Press, Cambridge, UK.

Knight, T. (1994b). Shape grammars and colour grammars in design. *Environment and Planning B: Planning and Design*, 21:705-735.

Knight, T. (1999a). Shape Grammars: Six types. *Environment and Planning B: Planning and Design*, 26:15-31.

Knight, T. (1999b). Shape Grammars: Five questions. *Environment and Planning B: Planning and Design*, 26:477-501.

Knight, T. (1999c). Applications in architectural design, and education and practice. Report, NSF/MIT Workshop on Shape Computation, Department of Architecture, School of Architecture and Planning, Massachusetts Institute of Technology.

Knight, T. (2003). Computing with emergence. *Environment and Planning B: Planning and Design*, 30:125-155.

Koning, H. and Eizenberg, J. (1981). The languages of the prairie: Frank Lloyd Wright's prairie houses. *Environment and Planning B: Planning and Design*, 8:295-323.

Koza, J. (1992). *Genetic programming: On the programming of computers by means of natural selection*. The MIT Press, Cambridge, MA.

Koza, J. R., Keane, M. A., Streeter, M. J., Mydlowec, W., Yu, J. and Lanza, G. (2003). *Genetic programming IV: Routine human-competitive machine intelligence*. Kluwer Academic Publishers.

Kundu, S. and Hellgardt M. (1996). A networks approach for representation and evolution of shape grammars. *Artificial Intelligence in Design '96*, 291-310.

Lee, H. C., Chan, K. H. and Tang, M. X. (2000). Automatic Generation of Machine Components for Variant-oriented Product Design. In *Proceedings of the Third International Conference on Computer-Aided Industrial Design and Conceptual Design (CAID&CD '2000)*, pages 190-202.

Lee, H. C. and Tang, M. X. (2004). Evolutionary Shape Grammars for Product Design. In *Proceedings of the Seventh International Conference on Generative Art*, pages 309-322, Politecnico di Milano University.

Lee, H. C. and Tang, M. X. (2006). Generating Stylistically Consistent Product Form Designs Using Interactive Evolutionary Parametric Shape Grammars. In *Proceedings of the Seventh International Conference on Computer-Aided Industrial Design and Conceptual Design (CAID&CD '2006)*, pages 374 –379.

Lee, H. C. and Tang, M. X. (2007). Evolving Product Form Designs Using Parametric Shape Grammars Integrated with Genetic Programming. Submitted to *Artificial Intelligence for Engineering Design, Analysis and Manufacturing (AIEDAM).*

Leung, T. P. (2006). Towards a knowledge economy of Hong Kong based on creative and innovative design. *The Journal of Designing in China*, 2:42-55.

Li, A. I. (2004). Styles, grammars, authors and users. *Design Computing and Cognition*, pages 197-215.

Liang, K. –H., Yao, X., Newton, C. S. and Hoffman, D. (1998). Solving cutting stock problems by evolutionary programming. In Porto, V. W., Saravanan, N., Waagen, D. and Eiben, A. E., editors, *Evolutionary Programming VII: Proceedings of the Seventh Annual Conference on Evolutionary Programming*, volume 1447 of *Lecture Notes in Computer Science*, pages 291-300, Springer-Verlag, Berlin.

Lienig, J. and Thulasiraman, K. (1995). GABOR: A genetic algorithm for switchbox routing in integrated circuits. In Yao. X., editor, *Progress in Evolutionary Computation*, volume 956 of *Lecture Notes in Artificial Intelligence*, pages 187-200, Springer-Verlag, Heidelberg, Germany.

Lintermann, B. and Deussen, O. (1999). Interactive modeling of plants. *IEEE Computer Graphics and Applications*, 19(1):2–11.

Luan, F. and Yao, X. (1994). Solving real-world lecture room assignment problems by genetic algorithms. *Complex Systems - From Local Interactions to Global Phenomena*, pages 148-160, IOS press, Amsterdam.

Mackenzie, C. A. (1989). Inferring relational design grammars. *Environment and Planning B: Planning and Design*, 16:253-287.

McCormack, J. P. and Cagan, J. (2002). Designing Inner Hood Panels through a Shape Grammar Based Framework. *Artificial Intelligence for Engineering Design, Analysis and Manufacturing (AIEDAM)*, 16(4):273-290.

Michalewicz, Z. (1992). *Genetic algorithms + data structures = evolution programs*. Sringer-Verlag, Berlin, Germany.

Michalewicz, Z. (1996). *Genetic algorithms + data structures = evolution programs*, third edition. Springer-Verlag, London, UK.

Michalewicz, Z. and Schoenauer, M. (1996). Evolutionary algorithms for constrained parameter optimization problems. *Evolutionary Computation*, 4(1):1-32.

Monedero, J. (2000). Parametric design: A review and some experiences. *Automation in Construction*, 9:369–377.

Nunamaker, J. F., Chen, M. and Purdin, T. D. M. (1991). Systems development in information systems research. *Journal of Management Information Systems*, 7(3):89–106.

Orsborn, S., Cagan, J., Pawlicki, R., and Smith, R.C. (2006). Creating cross-over vehicles: Defining and combining vehicle classes using shape grammars. *Artificial Intelligence for Engineering Design, Analysis and Manufacturing (AIEDAM)*, 20:217-246.

Ozkaya, I. and Akin, O. (2006). Requirement-driven design: assistance for information traceability in design computing. *Design Studies*, 27:381-398.

Paton, R. (1994). Enhancing evolutionary computation using analogues of biological mechanisms. In *AISB Workshop on Evolutionary Computing*, pages 51-64, Springer-Verlag, Germany.

Pham, D. T. and Yang, Y. (1993). A genetic algorithm based preliminary design system. *Journal of Automobile Engineers*, 207(D2):127-133.

Pugliese, M. J. and Cagan, J. (2002). Capturing a Rebel: Modelling the Harley-Davidson Brand through a Motorcycle Shape Grammar. *Research in Engineering Design-Theory Applications and Concurrent Engineering*, 13(3):139-156.

Radford, A. D. and Gero, J. S. (1988). *Design by optimisation in architecture and building*. Van Nostrand Reinhold, New York.

Rahmani, A. T. and Ono, N. (1993). A genetic algorithm for channel routing problem. In Forrest, S., editor, *Proceedings of ICGA '93*, pages 494-498, Morgan Kaufmann, San Mateo, Ca.

Rasheed, K. M. (1998). *GADO: A Genetic Algorithm for Continuous Design Optimization*. Doctoral dissertation, Department of Computer Science, Rutgers University, New Brunswick, NJ. Technical Report DCS-TR-352.

Rasheed, K. and Davison, B. D. (1999). Effect of global parallelism on the behavior of a steady state genetic algorithm for design optimization. In Angeline, P. J., Michalewicz, Z., Schoenauer, M., Yao, X. and Zalzala, A., editors, *Proceedings of the Congress on Evolutionary Computation (CEC '99)*, volume 1, pages 534–541. IEEE Press.

Rechenberg, I. (1973). *Evolutionstrategie: Optimierung technisher systeme nach prinzipien der biologischen evolution*. Frommann Holzboog Verlag, Stuttgart.

Rosenman, M. A. (1996a). A growth model for form generation using a hierarchical evolutionary approach. *Microcomputers in Civil Engineering*, 11(3):163-174.

Rosenman, M. A. (1996b). An exploration into evolutionary models for non-routine design. In *AID '96 Workshop on Evolutionary Systems in Design*, pages 33–38.

Rosenman, M. A. (1996c). The generation of form using an evolutionary approach. In Gero, J. S. and Sudweeks, F., editors, *Proceedings of the Artificial Intelligence in Design Conference (AID '96),* pages 643–662.

Rosenman, M. A. (2000). Case-based evolutionary design. *Artificial Intelligence for Engineering Design, Analysis and Manufacturing (AIEDAM),* 14:17–29.

Rosenman, M. A. and Gero, J. S. (1999). Evolving designs by generating useful complex gene structures. In Bentley (1999c), pages 345–364.

Schmidt, L. C. and Cagan J. (1998). Optimal configuration design: An integrated approach using grammars. *Journal of Mechanical Design*, 120(1):2–9.

Schnier, T. and Gero, J. S. (1996). Learning genetic representations as alternative to hand-coded shape grammars. In Gero, J. S. and Sudweks, F., editors, *Artificial Intelligence in Design '96*, pages 39-57, Kluwer, Dordrecht.

Shea, K. (1997). *Essays of Discrete Structures: Purposeful Design of Grammatical Structures by Directed Stochastic Search*. Doctoral dissertation, Carnegie Mellon University, Pittsburgh, PA.

Shea, K. (2001). An approach to multi-objective optimisation for parametric synthesis. In *13th International Conference on Engineering Design (ICED '01) - Design Methods for Performance and Sustainability, WDK 28*, pages 203–210.

Shea, K. (2002). Creating synthesis partners. *Architectural Design: Contemporary Techniques in Architecture*, 72(1):42–45.

Shea, K. (2004). Directed randomness. In Leach, N., Turnbull, D. and Williams, C., editors, *Digital Tectonics*, pages 10–23, Academy Press.

Shea, K. and Cagan, J. (1999). Languages and semantics of grammatical discrete structures. *Artificial Intelligence for Engineering Design, Analysis and Manufacturing (AIEDAM)*, 13:241–251.

Simon, H. A. (1969). *The Sciences of the Artificial*. The MIT Press.

Simon, H. A. (1984). The structure of ill-structured problems. In Nigel, C., editor, *Developments in Design Methodology*, John Wiley & Sons, New York.

Simon, H. A. (1990). *The Sciences of the Artificial (2nd edition)*. The MIT Press, Cambridge.

Sims, K. (1994a). Evolving virtual creatures. In Proceedings of *Computer Graphics, Annual Conference Series (SIGGRAPH '94),* pages15-22.

Sims, K. (1994b). Evolving 3D morphology and behaviour by competition. In Brooks, R. and Maes, P., editors, *Proceedings of Artificial Life IV*, pages 28-39. The MIT Press.

Soddu, C. (1994). The design of morphogenesis. An experimental research about the logical procedures in design processes. DEMETRA Magazine. (Internet location: http://soddu2.dst.polimi.it).

Soddu, C. and Colabella, E. (1997). Argenic design. In *Contextual Design / Design in Context Conference*, The European Academy of Design, Stockholm.

Sourav, K. and Michael, H. (1996). A networks approach for representation and evolution of shape grammars. In *Artificial Intelligence in Design '96*, pages 291-310, Kluwer Academic Publishers, Netherlands.

Stiny, G. (1975). *Pictorial and Formal Aspects of Shape and Shape Grammars: On the Computer Generation of Aesthetic Objects*. Birkhäuser, Basel.

Stiny, G. (1977). Ice-ray: A note on the generation of Chinese lattice designs. *Environment and Planning B: Planning and Design*, 4:89-98.

Stiny, G. (1980a). Introduction to shape and shape grammars. *Environment and Planning B: Planning and Design*, 7:343–351.

Stiny, G. (1980b). Kindergarten Grammars: Designing with Froebel's building gifts. *Environment and Planning B: Planning and Design*, 7:409–462.

Stiny, G. (1994). Shape rules: Closure, continuity and emergence. *Environment and Planning B: Planning and Design*, 21:49–78.

Stiny, G. (2006). *Shape, talking about seeing and doing*. The MIT Press, Cambridge, Massachusetts, London, England.

Stiny, G. and Gips, J. (1972). Shape grammars and the generative specification of painting and sculpture. *Information Processing*, 71:1460–1465.

Sun, J. (2002). *A framework for supporting generative product design using genetic algorithms*. Doctoral dissertation, School of Design, The Hong Kong Polytechnic University.

Sun, J., Frazer, J. H. and Tang, M. X. (1999). Application of evolutionary techniques in design for manufacturability. In *Proceedings of the Fifth International Conference on Computer-Aided Conceptual Design (CACD '99).*

Taura, T. and Nagasaka, I. (1999). Adaptive-growth-type 3D representation for configuration design. *Artificial Intelligence for Engineering Design, Analysis and Manufacturing (AIEDAM)*, 13:171-184.

Todd, S. and Latham, W. (1992). *Evolutionary Art and Computers*. Academic Press, London, UK.

Todd, S. and Latham, W. (1999). The mutation and growth of art by computers. In Bentley (1999c), chapter 9, pages 221–250.

Vignaux, G. A. and Michalewicz, Z. (1991). A genetic algorithm for the linear transportation problem. IEEE Transaction on Systems, Man and Cybernetics, 21(2):445-452.

Woods, W. A. (1970). Transition Network Grammars for Natural Language Analysis. *Communications of the ACM (CACM)*, 13(10):591-606.

Woods, W. A. (1973). An experimental parsing system for transition network grammars. In Rustin, R., editor, *Natural Language Processing*, pages 111-154, Algorithmics Press, New York.

Yao, X. (1993). An empirical study of genetic operators in genetic algorithms. *Microprocessing and Microprogramming,* 38:707-714.

Yao, X. (1999). *Evolutionary computation theory and applications*. World Scientific Publishing Co. Pte. Ltd.

Yao, X. and Liu, Y. (1996). Fast evolutionary programming. In Fogel, L. J., Angeline, P. J. and Bäck, T., editors, *Evolutionary Programming V: Proceedings of the Fifth Annual Conference on Evolutionary Programming*, pages 451-460, The MIT Press, Cambridge, MA.

Yao, X. and Liu, Y. (1997). Fast evolution strategies. *Control and Cybernetics*, 26(3):467-496.