Harmonised Shape Grammar in Design Practice

By

Arus Kunkhet

A Thesis Submitted in Partial Fulfilment of the Requirements of Staffordshire University for a Degree of

Doctor of Philosophy

Faculty of Computing, Engineering and Sciences

April 2015

Abstract

The aim of this thesis is to address the contextual and harmony issues in shape grammar (SG) by applying knowledge from the field of natural language processing (NLP). Currently shape grammars are designed for static models (Ilčík et al., 2010), limited domain (Chau et al., 2004), time-consuming process (Halatsch, 2008), high user skills (Lee and Tang, 2009), and cannot guarantee aesthetic results (Huang et al., 2009). The current approaches to shape grammar produce infinite design and often meaningless shapes. This thesis addresses this problem by proposing a harmonised shape grammar framework which involves applying five levels of analysis namely morphological, lexical, syntactic, semantic, and pragmatic levels to enhance the overall design process. In satisfying these semantically well-formed and pragmatically well-formed shapes, the generated shapes can be contextual and harmonious. The semantic analysis level focuses on the character's anatomy, body function, and habitat in order to produce meaningful design whereas the pragmatic level achieves harmony in design by selecting relevant character's attributes, characteristics, and behaviour. In order to test the framework, this research applies the five natural language processing levels to a set of 3D humanoid characters. To validate this framework, a set of criteria related to aesthetic requisites has been applied to generate humanoid characters; these include the principles of design (i.e. contrast, emphasis, balance, unity, pattern, and rhythm) and aspects of human perception in design (i.e. visceral, behavioural and reflective). The framework has ensured that the interrelationships between each design part are mutually beneficial and all elements of the humanoid characters are combined to accentuate their similarities and bind the picture parts into a whole.

Acknowledgements

I would like to express my appreciation and gratitude to the following individuals who dedicated themselves to the successful completion of this dissertation. Professor Bernadette Sharp and Dr Len Noriega, my first and my second supervisors, especially Professor Bernadette Sharp who gave me professional guidance and kind support throughout my research. Without their support, the completion of this dissertation would not have been possible.

I also would like to gratefully thank all my colleagues and friends at Staffordshire University for creating such a wonderful environment throughout my research. I thank the Faculty of Computing, Engineering and Sciences at Staffordshire University for the financial support. I thank all staffs at Staffordshire University especially at Stafford campus for your support and events. These help building up a wonderful atmosphere for both local and international students. I would like to thank Chiangmai University for the support before I came to study at Staffordshire University.

Last but not the least, I would like to thank sincerely my family, especially my parents and my girlfriend for believing and supporting in whatever I do. I would not grow up to be a man I am without their love, care, understanding, and support.

Table of Content

List of Figures

Chapter 1

Chapter 2

Chapter 3

Chapter 4

Chapter 5

List of Tables

Chapter 2

Chapter 3

Chapter 4

Chapter 5

Chapter 1: Introduction

1.1 Introduction

Design is defined as the process of creating new structures characterised by new parameters, aimed at satisfying specific requirements. It consists of several phases, namely the conceptual design, detailed design, evaluation and iterative redesign (Renner and Ekrárt, 2003; Bentley and Wakefield, 1997). The design generation of harmonious and efficient characters for successful computer games and movies is very important. Manual generation of such characters is expensive and requires highly skilled designers (Ilčík *et al*., 2010). Computational approaches have been used extensively for all these stages of design except the creative conceptual design phase. This phase of design has been regarded as a black art locked in a time warp of platitudes, vague design procedures and problem specific design rules (Goldberg and Rzevski, 1991). Consequently, the theory of shape grammars, developed by Stiny and Gips (1972), has provided a methodology to formalise the design process based on the use of primitive shapes and transformation rules.

For the past three decades, shape grammars have been used mainly to study architectural design (Stiny and Mitchell, 1978), paintings (Stiny and Gips, 1972) and product design (Trescak *et al*., 2009). In the last decade, shape grammars began to play an important role in application areas such as computer graphics (e.g. simulations, computer games and movies) (Preda *et al*., 2005). However, most current shape grammar implementations have operated in limited experimental domains, have provided little support for real designs and focused mainly on two-dimensional space (Chau *et al*., 2004).

Shape grammars consist of rules limited to manipulation and transformation of geometric elements and are unable to handle organisational and contextual information. Currently, they are designed for static models only and confined to geometric and spatial derivation rules (Ilčík *et al*., 2010). Despite some of their design principles and transformational rules shape grammars cannot guarantee aesthetic results (Huang *et al*., 2009).

1.2 Background of the Research

The shape grammar, developed by Stiny and Gips (1972), had a vocabulary of shapes and spatial relations between shapes. The vocabulary elements of shapes consisted of points, lines, planes and volumes. Krishnamurti and Earl (1992) define shape as a finite set of maximal straight lines of finite, nonzero length, where each line is specified by the coordinates of its end points. A shape grammar also includes a set of transformation rules to capture the knowledge of how practical design methods may be applied to manipulate and combine various physical components (Lee, 2007). A shape is generated by an initial shape and recursively applying various shape operations of addition, subtraction and spatial transformations such as shifting, mirroring and rotating. Rules specify which and how the particular shapes may be replaced by applying a series of transformations that permit one shape to be part of another.

Since the inception of shape grammars researchers have focused on developing shape grammar interpreters to automate the application of shape rules and to generate networks of designs (Krishnamurti and Giraud 1986; Krishnamurti 1982; Chase 1999; Tapia 1999). Recent work by Jowers and Earl (2010) have extended shape grammar interpreters by developing a method for defining shapes as compositions of parametric curve segments in 2D and 3D space. Riemenschneider *et al.* (2012) and Martinovic and Van Gool (2013) defined an extended shape grammar to generate parsing of façade imagery. A review of shape grammar implementations is discussed in chapter 2.

1.2.1 Shape Grammar Issues

Shape grammars are useful for generating a large variety of designs. However, most current shape grammar implementations still operate in limited experimental domains and fall short in support for real designs (Chau *et al.*, 2004). They use only rectilinear basic elements and are mostly limited to 2D space or primitive 3D shapes. They provide little support for high quality designs such as complex 3D geometry, thus confining successful designs to a very limited area. They have been proved effective in the field of architecture and engineering as well as computer-aided design tools. However, in the last decade, many researchers have sought the possibilities of extending shape grammars to other fields such as computer graphics, computer games and animation (Huang *et al*., 2009). To create a set of a large variety of buildings and characters in game design or animation, a massive amount of effort and skill is required. Shape grammars could increase the possibility of design variations. However, this would result in time-consuming tasks when operated with common planning methods (Halatsch *et al.*, 2008). There is a need for a framework to reduce the unnecessary time-consuming design process and allow designers to concentrate on their design activities, such as evaluation of designs and making design decisions (Lee and Tang, 2009).

The traditional way of modelling is mainly under human guidance which involves creating a geometric representation of the design object surface. Therefore, it may be more effective to create a new model rather than to modify a similar one (Ilčík *et al*., 2010). In computer games, a humanoid character, for example, requires a careful and skilled modeller, depending on the contextual constraints linked to the morphology. Shape grammars appear to be able to parameterise several types of designs which could then be used to semi-randomly generate cityscapes for computer games and animation (Chau *et al*., 2004). The 3D design of humanoid characters in computer games is very complex. The characters must reflect their physiological and sociological profile relevant to the story's environment. Design concepts of the generation of shape grammars involve human's subjective selections and evaluations in controlling modification of the shapes and shape grammar rules (Lee, 2007). The designers are required to be technically competent to write shape grammars that can be parsed and compiled by the application (Huang *et al*., 2009). Creating an effective design requires an expert trained in creation and usage of shape grammars (Orsborn *et al*., 2006). In addition, computational representations of products enable the automatic generation of designs. However, not all designs would be considered valuable. A valuable design may be conceived as a set of problem spaces described by a set of criteria or variables. The aim is to reduce the distance between the problem spaces and the solutions (Merrick *et al*., 2013). To find the balance between stylistic consistency and innovation in a computational framework is also one of the critical issues highlighted by Lee and Tang (2009). In many cases, users have found that some or most of the design generation has not been feasible for the purposes of the designs nor for the designs' environment. It would have required further evaluation of schemes or algorithms to filter these results.

1.2.2 Context and Harmony in Design

Stiny (2006) discussed how the grammar of visual mathematical arguments could be used to describe and construct shapes by means of a formal algebra. He compared shape grammars with Noam Chomsky's verbal grammars. In addition to aesthetics and visual appeal, shape grammars, like any grammar, focus on syntax. The shapes are the vocabulary (or lexicon) of a shape grammar and the grammar rules consist of a set of spatial design transformations that are to be applied to produce new shapes. In most applications, it has been left to the user to guide the selection of the rules in order to meet the design objectives. However, there are many choices of rules which may result in several emergent properties corresponding to various conditions and objectives. To resolve these issues, researchers have applied artificial intelligence techniques to control the selection of the rules. Shea and Cagan (1999) have applied shape annealing to produce optimally directed designs. Explicit domain knowledge has been placed within a grammar as rules and syntax whilst design interpretation has been used to select forms that fulfil functional and visual goals. O'Neil *et al,* (2009) have combined genetic algorithms with a shape grammar to encode human domain knowledge in order to rediscover known benchmark target structures.

Although these approaches can handle complex 3D objects, they are better at generating free-forms or unpredictable shapes than at fulfilling specific harmonised objects. These approaches may be acceptable to engineering design and product design; however, in computer games and movie applications they cannot provide a rich semantic and harmonised design set of characters. The generation engine of a shape grammar requires a deeper level of analysis. It has to go beyond the syntactic level and consider both semantics and context in order to generate a harmonised set of characters. The harmony will help bring a realistic feeling to the audience suggesting that characters come from the same world or same story. Characters should be compatible and exhibit an agreeable set of features and personalities. Any non-harmonised character should be considered as an error in the design set and should be rejected. In order to achieve harmony, all characters should exhibit not only agreeable but also compatible characteristics with their function and role within their world space.

This thesis is to address the problem of harmony. Consequently, it is to investigate the field of natural language processing and to extend current shape grammars in order to achieve harmony in design. Thus, the main research question addressed in this thesis is as follows: *Can the integration of two natural language processing levels (i.e. the semantic and pragmatic levels) to the current shape grammars help bring context and harmony into design, and, in particular, to character design?*

1.3 Aims and Objectives

The main aim of this thesis is to develop a novel shape grammar framework to generate harmonious and contextual based design objects. It is to address the need to bring harmony and context into the design of objects, an important gap in shape grammars. This is to be achieved by extending current shape grammars to include two new levels, semantic and pragmatic, to capture context and achieve harmony in design.

Most research in shape grammars has focused on geometric properties, architectural design and product design. This thesis is to address harmony and context in design by adding semantic and pragmatic levels inspired by natural language processing (NLP) theories. It is to use the domain of 2D and 3D character design in creating a set of harmonious characters to validate the proposed approach. In order to achieve the foregoing, the following objectives need to be carried out:

- To review current problems and approaches in shape grammar associated with a generative character design which can support the generation of 2D and 3D characters,
- To investigate the theory of natural language processing (NLP) and how it may be applied to provide harmony and context in the design process,
- To extend current shape grammars by applying two further levels (i.e. the semantic and pragmatic levels) inspired by natural language processing

(NLP) in order to generate a novel and harmonious 2D and 3D shape grammar,

- To apply the above harmonised shape grammar using appropriate artificial intelligence techniques (e.g. genetic algorithms) to generate a set of harmonised 2D and 3D humanoid characters,

- To validate and evaluate the novel extended shape grammar.

1.4 Research Methodology

There are two major research approaches which can be adopted by a researcher: qualitative and quantitative research methodology. This is one of the most important factors to consider when conducting research. In order to select the appropriate research methodology the research philosophy embedded in each research methodology should be examined.

The research philosophy helps in understanding the nature of knowledge and how that knowledge can be developed throughout the research. The most common way is to describe it in terms of the research onion, which represents the research methods in six different layers (Saunders *et al*., 2012). The research onion is divided into two sections: the top half and the bottom half. The strategies in the top half of the onion are more suitable for a quantitative approach whereas the bottom half strategies are more suitable for a qualitative approach (Fig.1.1).

The first layer includes, from bottom to top, pragmatism, interpretivism, realism and positivism (Fig.1.1). Positivism uses observation of social reality to achieve the production of credible data as in the natural *sciences.* Realism relates to facts that can be proven scientifically as reality is the truth. Interpretivism deals with how to understand humans as social actors whereas pragmatism is concerned with the interplay between action and knowledge. The second layer includes deductive and inductive methods. In the deductive approach, a conclusion is reached based on a set of premises assumed to be true whereas the inductive reasoning learns from known facts in order to produce general principles. The third layer consists of a series of methods, some of which are primary, used in qualitative research such as archival research, ethnography and ground theory. Others tend to be adopted for a quantitative research approach such as case study, survey and experiment action. Action research is often used in a qualitative rather than in a quantitative approach and may be described as research into a practice undertaken by the researchers with the aim of changing and improving the practice. The last two layers include methods related to observational studies, namely a cross sectional and longitudinal approach and may involve the application of a multiple or single research method.

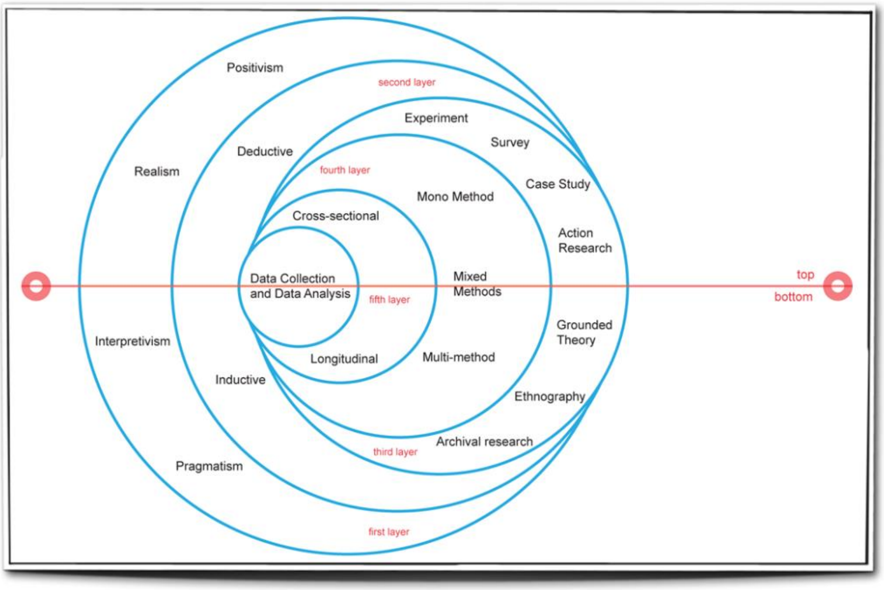


Fig.1.1 *Research Onion Model (Saunders et al., 2012)*

To achieve its objectives, this research considers a case study to test and validate the proposed extended shape grammar. The design of humanoids is a means to apply the novel shape grammar to investigate its various levels and processes required to achieve harmony in design. The concept of harmony can be viewed from many positions, namely art historians, architects, musicians, designers. It can be influenced by culture, experience, and personal satisfaction. To adopt an objective approach, this study defines a harmonious design as being achieved when all elements of design are in unity with each other. These design elements are to include shape, form, colour, texture, environment, proportion, balance, style and rhythm. The stages to be adopted in this research are the following:

- Studying shape grammar theory and reviewing the various approaches, applications and key issues,

- Formulating the research question, aim and objectives of this research project in the light of the literature review of shape grammar,
- Studying the field of natural language processing and investigating areas which could address the research question,
- Developing a new shape grammar,
- Identifying a case study (e.g. humanoid characters) to test the proposed shape grammar.
- Implementing the proposed shape grammar and evaluating the first generated characters with respect to morphological and functional specifications,
- Evaluating the next generation of humanoid characters with respect to the harmonious design criteria,
- Evaluating further the proposed shape grammar.

1.5 Ethical Issues

This research project adheres to the following guidelines of the British Computer Society (BCS) Code of Conduct:

- Take into consideration the wellbeing of others, security, privacy, public health, environments and is accomplished without any reference to sex, age, disability or race. In order to carry out the required professional responsibilities, this research ensures compliance with *current* legislation and maintains competence and integrity.
- Carry out professional responsibilities with due diligence and care in compliance with the relevant authority's requirements whilst exercising professional judgement at all times.
- Accept professional responsibilities for the work of colleagues and respects confidential information as required by legislation.

- This research does not misrepresent or withhold information on the performance of products, systems, services or take advantage of the lack of relevant knowledge or inexperience of others.

- The researcher accepts the personal duty to uphold the reputation of the profession and seeks to improve professional standards through participation in their development, use and enforcement.

1.6 Outline of the Thesis

Chapter 1: *Introduction*

This thesis is divided into six chapters. This chapter begins with a brief introduction to shape grammars and their main issues. Then it describes the aims and objectives of the research and outlines the research methodology to be adopted. The ethical issues related to this project and the thesis structure are also provided.

Chapter 2: *Literature Review*

This chapter reviews shape grammars and describes the two types of shape grammars. It provides a definition of shape rule properties and behaviour and reviews the applications of shape grammars from their introduction in 1971 to the recent developments including their limitations and issues which have limited their implementation in more challenging areas. To address the issues of harmony in character design, this chapter provides a definition of harmony in related art and design fields, and discusses the human perception in design.

Chapter 3: *Natural Language Processing (NLP)*

This chapter describes six levels of NLP and their applications. It also describes the contribution of NLP to shape grammars and explains how these six levels, when included in shape grammars, could bring context and harmony into design. The last two sections explain how NLP is integrated with shape grammars in order to address the harmony issue.

Chapter 4: *Harmonised Shape Grammars*

This chapter describes the proposed harmonised shape grammar framework. It focuses on the key issues of the research describing how to combine a shape grammar and natural language processing to achieve harmony in character design. To this end an overview of character design and object modelling techniques are described. The proposed harmonised shape grammar is to be applied to the design of a set of humanoid characters whose properties and habitat are to be represented ontologically.

Chapter 5: *Experimental Study*

This chapter explains how natural language processing is to be implemented in a shape grammar aimed at generating a set of harmonious humanoid characters. It describes the implementation of each of the six levels of NLP based on an ontological representation of humanoid characters. The implementation is to be carried out using MAYA Embedded Language (MEL-scripts) which generates the first humanoid parents. Genetic algorithms, and their implementation, that are to produce subsequent humanoid generations ensuring that harmonious offspring would be ‘created’, are described. The last section evaluates the proposed shape grammar with respect to criteria related to human perceptions in design.

Chapter 6: *Discussion Conclusion*

This chapter revisits the main objectives of this research and evaluates the overall approach. It outlines the main issues, challenges, and limitations of this novel approach, and proposes future work.

Chapter 2: Literature Review of Shape Grammars

2.1 Introduction

The aim of this research is to address the contextual and harmonisation issues in shape grammars (SG) by applying knowledge from the field of natural language processing (NLP). This chapter reviews the literature related to shape grammars and their applications, and focuses on two important aspects: shape grammars and harmonisation issues. Sections 2.2 and 2.3 introduce the theory of shape grammars, their properties and shape rules. Section 2.4 reviews methods and applications of shape grammars. Section 2.5 discusses current issues in shape grammars and explores the concepts of harmony in design. The principles of harmony in design are crucial to the development of a harmonious and contextual shape grammar. To this end, section 2.6 reviews the principles in design.

2.2 Shape Grammars

Shape grammars originated in 1972 when Stiny and Gips first introduced a shape grammar in order to create a grammatical pattern painting design. A shape grammar comprises a *vocabulary* of shapes (e.g. points, lines, planes, or volumes) and *spatial relations* between shapes (Stiny and Gips, 1972). A shape is generated by starting with an initial shape and recursively applying various transformation *rules* (e.g. shifting, mirroring and rotating) and shape *operations* (e.g. addition, subtraction) (Lee, 2007) Fig.2.1. The foundation of a shape grammar lies in the clear understanding of the diagrammatic and parametric rules. Both types of rules are quite similar in their principles; however, they produce distinct results in different situations.

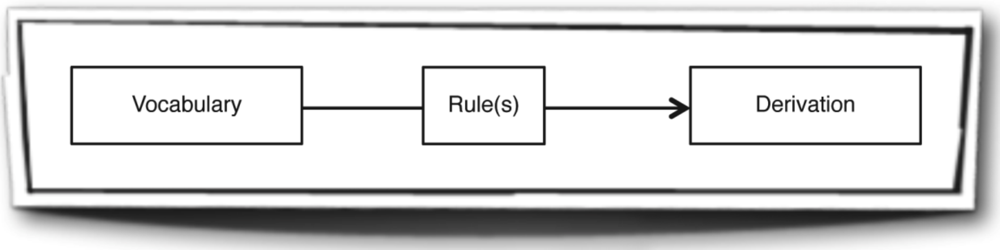


Fig.2.1 *Shape Grammar Model (Lee, 2007)*

2.2.1 Diagrammatic Shape Grammars

Diagrammatical shape grammar rules are based on a generic 2D diagram. The design process starts by applying a rule to a vocabulary. The applied rule(s) can be repeated several times, one rule at a time. The process is simple, as the shapes will be continually formed until the required shapes are produced. Diagrammatical shape grammars are used in applications of pattern design, abstract painting and sculpture, and architecture (Lee, 2007; Stiny and Gips, 1972; Flemming, 1987).

In diagrammatical shape grammar formalism, the transformation process starts from the left hand side (LHS) with an initial vocabulary of shapes. Through the emergence and spatial transformations, shape rules are continually applied by the designer until the derived shapes are completed or are no longer available for further modification.

The example, given in Fig.2.2, shows how a shape grammar is used to generate a derived representation based on a Mondrian painting (Schnier and Gero, 1998). It starts with the rectangle as the initial shape and applies a set of rules recursively as follows:

1. New rectangles can be formed by drawing a horizontal or a perpendicular line dividing the original rectangle.

2. The rule is repeated to create new rectangles.

3. The created rectangles are considered as new rectangles and are available for further rule applications.

4. Any created rectangle that becomes a square is marked in blue.

5. Any created rectangle that loses the square properties from the previous step must have the colour unmarked.

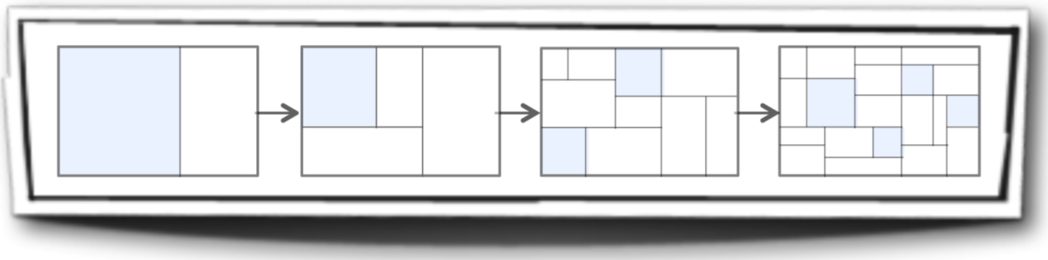


Fig.2.2 *Diagrammatic Shape Grammar Rules in 2D Drawing (Schnier and Gero, 1998)*

2.2.2 Parametric Shape Grammars

A parametric shape grammar is an advanced form of shape grammar which allows variations by use of parameters, for example, changes in lines and angles of shapes (Agarwal and Cagan, 1998). The new vocabulary created by the rules is defined by parameters employing the parameter concept to all design elements. Being parametric, a greater variety of forms can be created. Derivations can be used as a new vocabulary, and the process is repeated again to generate a new shape or form.

Parametric shape grammars were first introduced in 1992 (Krishnamurti and Earl, 1992). Each shape rule, A → B, can be used to obtain a family of shape rules *g*(A) → *g*(B) where *g* is a parameterised function. In manual mode, the user of a shape grammar implementation provides parameters required by a parameterised function *g*. Users are required to input detailed information. For each mode, possible parameterisations are generated by a computer in a fashion similar to generation of shape transformations.

The example, given in Fig.2.3, shows the parametric shape grammar rules applied in 3D workspace. The vocabulary of the initial shape consists of four arms arranged to form a cross shape and a set of rules is applied as follows:

1. A new vocabulary is obtained by rotating the previous vocabulary clockwise or counter-clockwise along X, Y and Z axes, where the rotation origin is at the centre of the previous vocabulary.

2. A set of new vocabulary is generated by mirroring horizontally or vertically the previous vocabulary.

3. The mirror origin must be at the centre of that vocabulary.

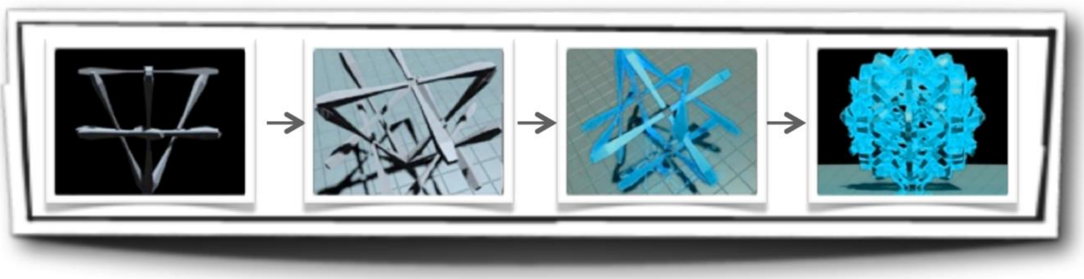


Fig.2.3 *Parametric Shape Grammar Rules in 3D Drawing*

Parametric shape grammars offer more flexibility in modifying shapes compared with diagrammatical shape grammars. They are used widely in applications such as product design, industrial designs, architecture, urban design and engineering applications (Lee, 2007). However, parametric shape grammars can be difficult to implement because of the increase in complexity of local design decisions and the increase in the number of elements to which attention must be paid for task completion (Woodbury, 2010).

2.3 Properties and Behaviour of Shape Rules

The shape rules constitute an algorithmic system for creating and understanding designs directly through computations with shapes, rather than indirectly through computations with text or symbols (Knight, 2000). In shape grammar formalism, rules are spatial and emergent (Stiny, 2006). The spatial properties are used to define shape transformation arrangements, whereas the emergent properties allow shape rules to be applied to create sub-shapes. The shape rules include two parts separated by an arrow pointing from left to right. The left hand side of an arrow is termed as the left-hand side (LHS) and the right side is termed as the right-hand side (RHS) (Stiny, 2006). LHS depicts a given state of a shape in the vocabulary and a marker representing the shape orientation. RHS depicts how the shape on LHS would be transformed or rearranged (Hoisl and Shea, 2009). Fig.2.4 shows a simple example of a spatial arrangement rule in a shape grammar. The shape rule is defined as A → B, when A defines a shape on LHS and B defines the resulting shape on RHS when the spatial rule is applied.

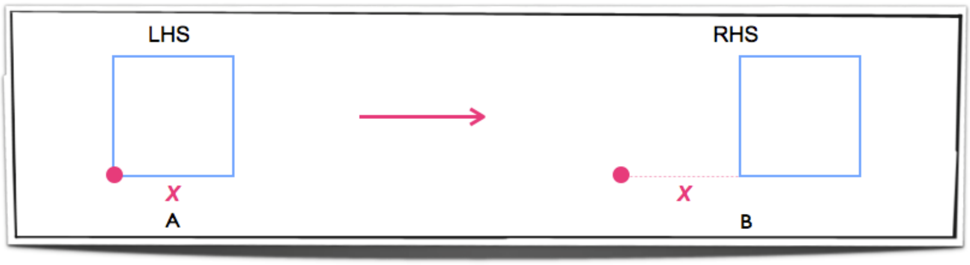


Fig.2.4 *Example of Shape Spatial Rule*

Shape rules include a number of properties. Shape emergence is one of the elemental properties that distinguish the use of shape grammars from other computer-aided design tools (Lee, 2007). The benefit of shape rules is that they reveal more design creations. Shape rules calculate shapes using recursion and embedding techniques. The emergent properties of shape grammars involve the formulation of sub-shape concepts, where a shape may be a sub-shape of another.

In most computer-aided design tools, rules are not allowed to be applied to the sub-shape. The boundary between the parts of shape is made when the shape is initially created. In contrast, shape emergence allows rules to be applied to both basic elements and sub-shapes. For example, by using the same rule applied in Fig.2.4, one can generate overlapping rectangles, *a*, *b*, and *c* (Fig.2.5). The overlap between shape *a* and shape *b* creates a sub-shape *ab*, and the overlap between shape *b* and shape *c* creates a sub-shape *bc*. The rule is applied to sub-shape *bc*, and again to sub-shape *ab*, merging three rectangles into a single complex shape.

The shape's emergent properties allow numerous creative possibilities based on a set of generative rules that are applied. The transformational rules of shape grammars can be executed flexibly whenever the condition of either shapes or sub-shapes satisfies the rule (Lee, 2007). This property of shape rules, relying on the concept of a sub-shape, forms part of the emergent properties of shape grammars. This demonstrates a certain type of emergent behaviour (Stiny, 1994).

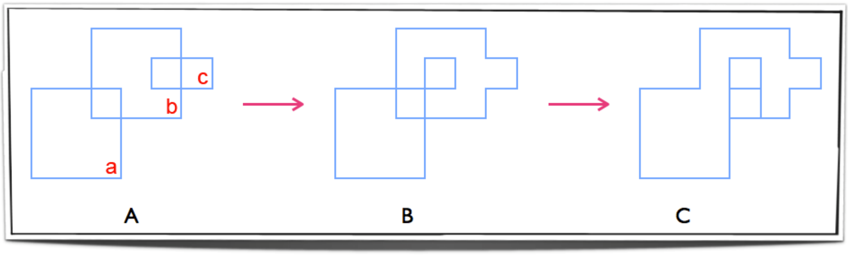


Fig.2.5 *Shape Rules Example*

2.4 Applications of Shape Grammars

Shape is an inexhaustible source of creative ideas. To understand shapes is a useful way to understand what is possible in design (Stiny, 2006). Design is calculating. Shape grammars provide ways to calculate shapes and understand their properties. Shape grammars have been applied in many design areas such as architecture (Stiny and Mitchell, 1978; Flemming, 1987; Duarte and Correia, 2006), engineering design

(Brown, 1997), product design (Lee and Tang, 2004), painting and sculpture (Stiny and Gips, 1972), design education (Chen *et al*., 2009), and computer graphics (Zhang and Lin, 2008).

In the field of architecture, shape grammars are used widely to study architectural planning design and planning layout. For example, the recreation of Villa Malcontena floor plan (Stiny and Mitchell, 1978) demonstrates how a number of plans can be generated in the style of Palladio using a parametric shape grammar. Flemming (1987) generated houses in the Queen Anne style, which dominated domestic architecture in the United States of America in the 1880s in areas such Pittsburgh's historic Shadyside district, specified by a shape grammar. Separate grammars were given for the generation of plans and for the articulation of plans in 3D. Aksamija *et al* (Aksamija, 2010) have integrated knowledge bases and shape grammars to study the character of existing row-house and high-rise apartment buildings in Baltimore. Academic researches such as Li's thesis (Li. 2001) developed Yingzao Fashi Grammar to study the building and construction architecture of Chinese state buildings during the mid-Song Dynasty of China. Duarte's thesis (Duarte, 2005) used a shape grammar to generate novel designs for customising Alvaro Siza's mass houses at Malagueira, and in his more recent work he has developed a parametric urban shape grammar for the Zaouiat Lakhdar quarter of the Medina of Marrakech in Morocco (Duarte *et al*., 2007). Teboul *et al* employed shape grammar parsing for façade segmentation using Reinforcement Learning (RL). The research used shape parsing which entailed simultaneous geometry optimisation and the façade topology (such as number of floors of the façade) (Teboul *et al*., 2011). An alternative Palladian shape grammar (Benros *et al*., 2012) described a way to use a shape grammar to recreate Palladio's villas with different parametric shape rules and methodology from the previous version proposed by Stiny and Mitchell (Stiny and Mitchell, 1978). The aim was to test the hypothesis that different grammars can generate the same corpus of designs, which received positive results. GRAMATICA (Correia *et al*., 2012) discussed the shape representation, shape generation, and shape control aimed at using a 3D shape grammar interpreter to target mass-customised housing. The implementation used several shape grammars, including the design module of a specific shape grammar called DESIGNA. Santos *et al* (2012) proposed a multi-agent rule-based architecture for a computational system to support generic work with shape grammars. The system used symbolic knowledge representation and reasoning, rule-based systems and multi-agent systems as the core. Riemenschneider (Riemenschneider, 2012) used a shape grammar as a method to express hierarchical spatial relationships and represent constraints for semantic façade interpretation. The framework provided feasible generic façade reconstruction by combining low-level classifiers with mid-level object detectors to infer an irregular lattice. Recently, Noghani *et al*. (Noghani, 2013) used a shape grammar to generate classical Roman architecture aimed at providing a shape grammar based on the writings of the ancient Roman architect Vitruvius, encoding rules for procedurally defining the make-up of Roman settlements. Martinovic and Van Gool (Martinovic, 2013) developed Bayesian grammar learning for inverse procedural modelling for architectural parsing façade imagery generation by using Haussmannian buildings in Paris as a case study.

In the last two decades, shape grammars were extended to the engineering field. Engineering shape grammar design was reviewed comprehensively by Cagan (Cagan, 2001) and was described as a formal engineering design synthesis. Engineering shape grammars were used to help the designer generate and evaluate ideas and concepts during the conceptual phase of the design process (Brown, 1997). In the following years, Agarwal *et al* developed Coffee Maker Grammar (Agarwal, 1998) and MEMS Grammar (Agarwal, 2000) using Java and LISP. The frameworks proposed the use of shape grammars for engineering expert systems in order to represent knowledge on both the functionality and the form of the product.

In computer-aided design, shape grammar has been developed for use in many different purposes. For example, Krishnamurti developed a shape grammar interpreter (Krishnamurti, 1982) and a shape generation system (Krishnamurti, 1982) to use as a design generating application, followed up with GRAIL (Krishnamurti, 1992) and a parametric shape grammar interpreter as design analysis tools. Shea developed EifForm (Shea, 1997), SGMP (Shea and Ertelt, 2009), and an interactive 3D spatial grammar system (Hoisl and Shea, 2010) for design study purposes. Shape grammars have been developed using many different programming languages, including PROLOG (Chase, 1989; 2002), C++ and CLP to analyse aircraft design, (Heisserman, 1991), and AutoCAD and Auto LISP to develop a shape grammar editor (Shelden, 1996).

Many researchers have developed a shape grammar as a design approach application. For example, Piazzalunga and Fitzhorn (Piazzalunga, 1998) developed a 3D shape grammar interpreter using ACIS and LISP in order to extend the use of a shape grammar into 3D implementation. Chien (Chien, 1998) developed a system called SG-Clips aimed at supporting the automatic generation of design form grammars. Chau (Chau, 2002) developed a 3D Interpreter aimed at improving shape emergence, parametric shape rules, and the ability to work with complex 3D geometry. Jowers (Jowers, 2006) introduced Qi as a shape grammar curve-based design tool to work with curve properties. This system aimed at extending the applicability of a shape grammar to enable computation with free-form shapes. McKay *et al* (McKay, 2008) introduced a shape synthesis system based on a shape grammar formalism. The research aimed at developing a new generation of computer-aided design system to support design exploration as well as the production of product definitions. Trescak *et al* (Trescak, 2009) developed SGI2 to interpret most non-deterministic 2D shape grammars, and to support real-time sub-shape detection and labelling rules. Campbell (Campbell, 2010) developed an application called GraphSynth to use shape grammar working in 2D space in order to generate architectural plan design. Li *et al* (Li, 2010) analysed shape grammar development systems aimed at developing a prototype system to allow a user to edit grammars, test them, and be able to switch between the two types of activity.

Since 2005, shape grammars have been applied to product design as an approach to design methodology. In product design, shape grammars are expressed in terms of a grammatical design which considers relationships between each product's components and the way they are assembled together to achieve their final design. Designs of products are studied and explained with their grammatical calculation. So every aspect of design is calculated and must be grammatically correct. A designer can understand the grammar of an existing product in order to create similar products in their product line. For example, Chau *et al.* (Chau, 2004) developed a U_{13} shape grammar to support rectilinear and curvilinear basic elements in 3D space which was tested on two case studies: a Coca-Cola bottle grammar and a Head & Shoulders bottle grammar. The research could identify grammars of the two product lines and the relationship in terms of the development on their bottle designs. Lee and Tang (Lee, 2004) have developed an interactive system that uses parametric 2D and 3D shape grammars for digital camera design, incorporating an evolutionary algorithm for exploring product forms at the early stage of the design process. In the vehicle-design industry, shape grammar implementations were used to analyse current vehicle parts in order to apply the same grammar to create a new vehicle design. For example, McCormack and Cagan (McCormack, 2002) developed a parametric shape grammar in vehicle design study to design a car inner hood panel. Pugliese and Cagan (Pugliese, 2002) used a shape grammar to capture a motorcycle brand identity design. Orsborn *et al* (Orsborn, 2006) developed a vehicle grammar to create a new cross-over vehicle by defining and combining different vehicle classes.

Shape grammars have been also used to generate paintings and sculptures, and have been applied in many systems for design studies (Stiny and Gips, 1972). By applying appropriate rules and grammars, the generative specification of design can be achieved. Gips introduced a simple interpreter (Gips, 1975), and a syntax-directed program which performed a perceptual task (Gips, 1974) which worked in both 2D and 3D spaces. Zhang and Lin (Zhang, 2008) used a shape grammar to analyse the art of Tibetan Tangka paintings in order to generate a similar painting style. Chen *et* al. (Chen, 2009) used a shape grammar and tangible augmented reality to assist in collaborative design learning. The SG interpreter (Chase, 1987) translated a mathematical model to a computer implementation and a model for user interaction in grammar-based design systems, to use in design evaluation, such as GEdit, a visual implementation of a shape grammar system (Tapia, 1996), 3D Shaper, automatic generation and fabrication of design (Wang, 1998), Shaper 2D, and visual software for learning shape grammars (McGill, 2001).

In recent years, shape grammars have also played an important role in the field of computer graphics, computer games and animation. Fiedler and Ilčík have extended the application to the procedural modelling of humanoids (Fiedler & Ilčík, 2009) and procedural skeletons for kinematic extension (Fieldler & Ilčík, 2010). Huang *et al* (2009) used a shape grammar to generate a large variety of buildings useful in computer games. A shape grammar approach to computational creativity and procedural content generation in massively multiplayer online role-playing games was developed (Merrick, 2013) aimed at developing tools that can aid in or automate game design processes. The system used computational models of creativity as a part of the generative process to capture the usefulness and value of an existing human design while introducing variations through the model of interest.

Most early work in shape grammars was carried out manually (Gip, 1999). Later, many researchers developed computational algorithms to implement shape grammars leading to a generation of shape grammar interpreters in 2D and 3D spaces (Chau, 2004; Lee and Tang, 2009; Trescak, 2009; Krishamurti, 2010). Many of these interpreters have focused on design, and either generated shapes in the language or were guided by the user who selected the rule to be applied and the shape to apply it to. A few interpreters have been more analytical in their approach in determining whether the shape is generated in the language by the shape grammar or whether the sequence of rules is generated to produce the shape. Other types of interpreters have applied a grammatical inference strategy to generate shapes in the same style from a given a set of shapes. Table 2.1 compiles a list of shape grammar implementations, as provided by Chau (Gips, 1999), using a variety of tools to support 2D and 3D character designs. This table has been extended to include recent implementations.

Table 2.1*: Shape Grammar Implementations*[1]

No	Name	Author(s) and Year	Purpose	Type
1	Shape Grammars and the Generative Specification of Painting and Sculpture	Stiny and Gips, 1972	Geometric painting and sculptures	2D, 3D
2	Shepard-Metzler Analysis	Gips, 1974	Design application: SAIL	2D
3	Simple Interpreter	Gips, 1975	Design Application: SAIL	2D, 3D
4	Palladio Style	Stiny and Mitchell, 1978	Architectural design study	2D
5	Shape Grammar Interpreter	Krishnamurti, 1982	Design approach application	2D
6	Shape Generation System	Krishnamurti and Giraud, 1986	Design application: PROLOG	2D
7	Queen Anne Houses	Flemming, 1987	Architectural design study	2D

[1]*Extended from Chau et al (Chau, 2004), and Design Computing and Cognition workshop (Chase, 2010)*

8	SG Interpreter	Chase, 1987	Computer implementation	2D
9	Grammar Interaction	Chase, 1987	Computer implementation	N/A
10	Shape Grammar System	Chase, 1989	Design application: PROLOG Mac	2D
11	Genesis (CMU)	Heisserman, 1991	Design application: C, CLP	3D
12	Genesis-Boeing	Heisserman, 1991	Design application: C++, CLP	2D, 3D
13	GRAIL	Krishnamurti, 1992	Design approach application	2D
14	GRAMMATICA	Carlson, 1993	Design approach application	N/A
15	GEdit	Tapia, 1996	Design application: LISP, Mac	2D
16	Shape Grammar Editor	Shelden, 1996	Design application: AutoCAD, Auto LISP	2D
17	Implementation of Basic Grammar	Duarte and Simondetti, 1997	Design application: AutoCAD, Auto LISP	3D
18	EifForm	Shea, 1997	Design study	3D
19	Shape Grammar Interpreter	Piazzalunga and Fitzhorn, 1998	Design Application: ACIS, LISP	3D
20	SG-Clips	Chien, 1998	Design application: CLIPS	2D, 3D
21	3D Architecture from Synthesiz	Wang, 1998	Design application: Java, Open Inventor	3D
22	Coffee Maker Grammar	Agarwal *et al*., 1998	Design application: Java	2D, 3D
23	MEMS Grammar	Agarwal *et al*., 2000	Design application: LISP	2D
24	Shaper 2D	McGill, 2001	Design application: Java	2D
25	Engineering esign Synthesis	Cagan, 2001	Design approach review	2D
26	Motorcycle Brand Identity Grammar	Pugliese and Cagan, 2002	Brand design study	2D
27	Yingzao Fashi Grammar	Li, 2001	Design approach application	2D
28	3D Interpreter	Chau, 2002	Design approach application	3D

29	Grammar Use and Interaction	Chase, 2002	Design approach application	N/A
30	Shape Grammar to Design the Inner Car Hood Panel	McCormack and Cagan, 2002	Vehicle design study	2D, 3D
31	U_{13} Shape Grammar Implementation	Chau *et al*., 2004	Design application: Perl	3D
32	Parametric Shape Grammar for Digital Camera Design	Lee and Tang, 2004	Product design study	2D, 3D
33	Shape Design V2	Wong *et al*., 2004	Design approach application	3D
34	PROGRAMA (The first module of MALAG)	Duarte and Correia, 2006	To capture Siza's Malagueira Houses in Evora, Portugal	3D
35	Quad Interpreter (Qi)	Jowers, 2006	Curve design application	2D
36	Cross-over Vehicle Grammar	Orsborn *et al*., 2006	Vehicle grammar study	2D, 3D
37	Shape Synthesis System	McKay *et al*., 2008	Design approach application	2D
38	Zaouiat Lakhda of the Medina of Marrakech	Duarte *et al*., 2007	Architectural design study	2D
39	The Art of Tibetan Tangka Painting Analyse	Zhang and Lin, 2008	Painting study	2D
40	SG and Tangible Augmented Reality	Chen *et al*., 2009	Collaborative design learning	N/A
41	Large Variety of Building Generation	Huang *et al*., 2009	Urban design study	3D
42	SGMP	Ertelt and Shea, 2009	Design approach application	3D
43	SGI (2)	Trescak *et al*., 2009	Design approach application	2D
44	The Procedural Modelling of Humanoid	Fiedler and Ilčík, 2009	Design study	3D
45	DESIGNA (MALAG II)	Correia *et al*., 2010	The second design module of MALAG	3D
46	Parametric SG Interpreter	Krishnamurti, 2010	Design analysis	2D
47	Baltimore Row-house	Aksamija *et al*., 2010	Architectural design study	2D
48	GraphSynth	Campbell, 2010	Design approach application	2D
49	SG Development System	Li *et al*., 2010	Prototype system design application	2D, 3D
50	The Procedural Skeletons for Kinematic Extension	Fiedler and Ilčík, 2010	Design study	3D
51	Interactive 3D Spatial Grammar System	Hoisl and Shea, 2010	Design application	3D

52	Shape Grammar Parsing via Reinforcement Learning	Teboul *el al*., 2011	A recursive binary split grammar	2D
53	An Alternative Palladian Shape Grammar	Benros *et al*., 2012	Architecture design study	2D
54	GRAMATICA	Correia *et al*., 2012	Architecture design study	3D
55	Multi Agent Expert System Shell	Santos *et al*., 2012	Urban and architecture design study	2D
56	SG Façade Parsing	Riemenschneider, 2012	Architectural façade parsing	2D
57	SG for Online RPG Games Content Generation	Merrick *et al*., 2013	Procedural content generation	N/A
58	Vitruvian SG to Generate Classical Roman Architecture	Noghani *et al*., 2013	Architecture design study	3D
59	Bayesian Grammar	Martinovic and Van Gool, 2013	Parsing façade imagery	2D

2.5 Current Issues in Shape Grammars

Shape grammars are used particularly for applications in computer-aided architectural design applications, as they provide two levels of language analysis including vocabulary, known as shapes, and grammar, known as shape rules, in order to create new syntactically well-formed shapes.

Although shape grammars are useful for generating a large variety of designs, they still operate in limited experimental domains and fall short in support for real designs (Chau *et al*., 2004). They are designed for static models (Ilčík *et al*., 2010) and demand high user skills (Lee and Tang, 2009). They use only rectilinear basic elements and are mostly limited to 2D space or primitive 3D shapes. They also lack support for high quality design such as complex 3D geometry and cannot guarantee aesthetic results (Huang *et al*., 2009). Current research approaches are primarily focused on the need to reduce the time-consuming design process and to allow designers to concentrate on their design activities, such as evaluation of designs and making design decisions (Lee and Tang, 2009). However there is a greater need for a framework to support the aesthetic aspects of design, ensuring that the final design products are harmonious and contextually relevant to the technical requirements.

These issues have led this research to investigate the concept and principles of harmony in design.

2.6 Principles of Harmony

The term harmony originates from the Greek word harmonia which means joining, agreement, and concord. The term harmony is used mostly to describe the relationship between tones and notes in music and between hue, value and saturation in colour. In the Oxford English Dictionary, harmony is defined as the combination of simultaneously sounded musical notes to produce a pleasing effect (Oxford English Dictionary, 2013). However, the term can be interpreted in various types of art to describe the cohesion and coherence between design basic elements such as shape, form or spaces. This section aims at exploring the term harmony from different perspectives.

2.6.1 Principles of Harmony in Colour Design

Forms, colours, and their arrangement are the foundation elements of design (Holtzschue, 2011). Each colour has its own meaning and can bring different feeling to viewers. Viewers interpret the experience of colour in different and personal ways. From the way the human brain is constructed, one colour is perceived according to the surrounding colours. In order to identify harmony in colour design, one must understand colour not only individually (colour meaning), but also as relationships between colours (colour scheme), and the attributes of colour, including hue (shade of the colour), value (relative brightness of a colour), and saturation (hue intensity of a colours, i.e. ranging between dull and vivid).

Colours are composed from a vocabulary of colours. It is similar to the way the vocabulary of languages and the vocabulary of shapes work. Each colour in a vocabulary expresses its own meaning. However its meaning can be altered depending on its surrounding colours. This means that the identity of a colour does not only reside in colour itself but it also established by relationships between the colour and its neighbouring colours (Arnheim, 1969). In order to achieve a semantically well-formed colour design, the colour vocabulary and the colour grammar must be identified.

A grammar of colour can be described as a relationship and rhythm of one colour to its surrounding colours to be sensed as a whole either in order to construct a mathematical ordering arrangement to form such a pattern or in an abstract free form randomising representation as an abstract painting. A colour grammar can be described as a colour composition. For example, the way one arranges a set of colours into a group of a design. It is similar to the way words are put together to form a sentence or a music component such as different notes and tones that can be arranged to create a song. Form, colours, and their arrangement have equal importance in design (Holtzschue, 2011).

Colour meanings are individual and personal. Each person perceives the meaning of colours differently according to their personal experiences. This is the description of an understanding of each person toward a colour. However in a colour relation, when colours are put together in such a grammatical way, i.e. in correct forms and orders, it can bring a particular meaning to a viewer. When colours of red, orange, yellow, green, blue, indigo and violet are put together and arranged in this particular order a rainbow is produced. The meaning of each individual colour in the vocabulary is ignored by the viewer but the meaning is their relationship is the context of the composition. The meaning is perceived as a whole relating to the context.

Harmony in colour can be expressed in term of three independent dimensions: hue, value (lightness) and saturation (colour purity). The harmonisation in hue can be achieved by understanding the relationship between colours. For example, a red text can be set in a white or black background but when set in a blue background it creates an optical illusion and becomes very difficult to read. The harmonisation in value can be achieved in three different ways according to whether even intervals have occurred, or the value has occurred in the middle of a range or several colours have equal values. The harmonisation in saturation can be achieved when the overall level of saturation is relatively constant (Holtzschue, 2011).

2.6.2 Principles of Harmony in Music

In music, the term harmony is used to describe the simultaneous pitches including tones and notes, or chords (Malm, 1996). It includes the study of chord construction and chord progressions, and to find the principles of the relationship that govern them (Dahlhaus, 2007). Harmony is achieved by putting two or more notes together to produce coherence in tonal music. The term interval is used to describe the distance between two notes, and harmonic progression describes how a succession of notes is controlled and how one chord is followed by another.

2.6.2.1 Chords

Harmony can be achieved by understanding the relationship of pitch classes known as chords. Creation of a chord, starts with the root, followed by the third and the fifth intervals above the root. The chord members are named according to their interval above the root. A chord with three members in it is called a triad. Several chord qualities can be achieved depending on the size of the intervals being stacked. Chords are named according to their roots plus various term and characters indicating their qualities. For example, a G major triad may have members G (root), B (major third), B♭ (minor third), and D (fifth), called by default as simply a G chord.

2.6.2.2 Intervals

The term interval means the relationship between two separate musical pitches. The four-step interval means music that was composed by combining the four scale notes or seven chromatic notes above the root together. For example, a song that starts with the first note in pitch G may have the second note in pitch D which is at the interval of one fifth. The symmetrical harmonies can be created with a combination of notes with their specific intervals.

2.6.3 Principles of Harmony in Architectural Design

In architectural design, the term harmony refers to the architectural proportions (Vitruvius, 2001). These include geometric ratios, orientation of the buildings, features of the grounds where the building is located, the choice of materials in relation to the orientation of the building, and environmental and climatic conditions.

Many architectural designs have their geometric proportions based on the golden ratio, also known as the golden mean, mean ratio or golden proportion (Livio, 2002; Dunlap, 1997; Sadowski, 1996). The golden ratio occurs if the ratio of the sum of larger and smaller quantities to the larger quantity is equal to the ratio of the larger quantity to the smaller one. The relationship can be described by a mathematical formula as shown below where the Greek letter phi (ϕ) represents the golden ratio.

$$\frac{a+b}{a} = \frac{a}{b} \equiv \phi$$

The value of phi (ϕ) representing the golden ratio is:

$$\phi = \frac{1+\sqrt{5}}{2} = 1.6180339887 \ldots^{2}$$

Fig.2.6 shows the golden ratio represented as a golden rectangle with the larger rectangle A and the smaller rectangle B.

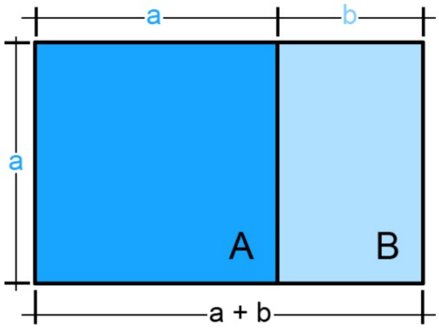


Fig.2.6 *Rectangles using the Golden Ratio of the So-Called Golden Rectangle.*

The golden ratio is widely used by both architects and artists especially as golden rectangles for elevations, plans or drawings. This ratio is believed to be aesthetically pleasing for the designs. The golden ratio properties were also studied by famous mathematicians such as Euclid (323-283 BC.) and Luca Pacioli (1445-1517) in order to achieve pleasing and harmonious proportions in designs, and the term has also been used to analyse the proportions of natural objects (Padovan, 1999). In ancient architectural designs, the golden ratio was used in many well-known architectural designs: the façade of the Parthenon, the Notre Dame cathedral in Paris, and the Taj Mahal in India. The golden ratio was also used in many modern architectural designs such as for the CN Tower in Toronto and the United Nations building in Switzerland.

The golden ratio has also been used in painting and arts. The philosopher, Agrippa (1486-1535), used the relationship of golden ratio proportions to draw a man's body as a pentagram inside a circle. The Renaissance polymath, painter, sculptor and architect, Leonardo da Vinci, used the golden ratio in many of his paintings (Burstein, 2006).

Architects have often used the proportional system principle in order to generate or constrain the forms to create a building plan or elevation design. The system of mathematical relations can be applied in many buildings in terms of grammatical rules which govern the relationship between different aspects of the design. The proportion systems were determined using geometrical methods in order to achieve the goal of producing a sense of coherence and harmony among the building elements and proportions.

2.6.4 Human Perception in Design

The way people understand and appreciate a design aesthetic when exposed to art forms develops from three particular levels of processing: *visceral*, *behavioural*, and *reflective* (Norman, 2004). These three levels work differently and generate different reactions to different parts of a design. The process begins when human eyes receive a sensory input or information. The visceral level processing is the first to act in order to define whether the information is good or bad, safe or dangerous, etc. The visceral processing level may be enhanced by the behavioural processing level, where most of human behaviour is controlled. Furthermore, the behavioural processing level can be also enhanced by the third processing level, the reflective level. When a person sees a design, for example a door handle, each level of the person's brain perceives the information from different perspectives of that particular design. The visceral processing level judges the appearance of the design, i.e. whether the design is attractive, modern or old-fashioned. The behavioural processing level focuses on the function of the design. For example, the pleasure and effectiveness of the door handle, whether it is easy to use, and how easy it is to grip the handle. The reflective processing level reflects a self-image, a personal satisfaction and memories for that particular person. For example, a person may have a preference for a particular material, such as aluminium or a stainless steel, rather than wood.

To achieve satisfaction in design, the three processing levels should be fulfilled. It is difficult to identify which processing level is the most important one. The visceral processing level in a design process, on its own can produce a very attractive product but may lack functionality. Focusing only on the behavioural processing level, the product may be designed with good functionality, but may lack attractiveness, which may result in a negative feeling towards the design. The reflective processing level in a design is very difficult to judge because it is related to a person's culture, personal experiences and self-satisfaction.

2.7 Conclusion

This chapter has introduced the two approaches to shape grammar: diagrammatic and parametric. A grammar-based synthesis framework was developed (Chakrabarti *et al*., 2011) which enables the definition of classes in a production system to generate geometric shapes syntactically in 2D and 3D. The framework was formalised in 1971 (Stiny and Gips, 1971) in order to generate painting patterns. It consists of two important parts: vocabulary and shape rules. The process requires at least one primitive shape as a start vocabulary. It requires shape rules including start rules, transformational rules and termination rules, which are to be applied in order to manipulate and transform the existing vocabulary to new shapes. Shape grammars have been used in a number of applications such as architectural design, engineering design, product design and study design.

The literature review has shown that shape grammars are designed primarily for static modelling (Ilčík *et al*., 2010). Use of a shape grammar is a time-consuming process (Halatsch, 2008), requires high user skills (Lee and Tang, 2009), and cannot guarantee aesthetic results (Huang *et al*., 2009). The current approaches to shape grammars have produced infinite designs with often meaningless shapes. The reason is that most of the existing approaches are limited to two levels of shape grammars for the vocabulary (lexicon) and rules (syntax). The extension of two levels, semantics and pragmatics, would bring meaning, context and harmony into design. Meaningful and contextual design can reduce time-consuming issues. Harmony can guarantee aesthetic results and can be used for non-static modelling. The literature review discusses the need to extend shape grammars to incorporate context and achieve harmony in design. Based on the literature review, these are the main concepts which contribute to our study: colour theory derived from the principles in colour design, unity extracted from the harmony of music, shape proportion based on the golden ration principle in architectural design, and the visceral, behavioural, and reflective of processing associated with the human perception in design.

Chapter 3: Natural Language Processing

3.1 Introduction

In order to address the issue of context and harmony in shape grammar, this thesis explores the field of natural language processing which can offer promising linguistic concepts. Natural languages are the languages that humans learn and use in everyday communication with each other (Harris, 1985). Natural languages include both emotional expressions which are learnt early in life (such as grunts, whines, and cries), and spoken languages (such as English, French, and Japanese) that are learnt through social interaction and later through schools, knowledge and culture exchange. In contrast, artificial languages are languages which are created by humans in order to communicate with machines, such as computer programming languages and mathematical notation. Natural language processing (NLP) is a knowledge-based approach to computer manipulation of natural language (Bird *et al.*, 2009). The traditional approach is to translate utterances into a formal specification that can be processed further by computer. In natural language interaction, NLP involves reasoning, factual data retrieval, and generation of an appropriate tabular, graphic or natural language response. NLP is used in many technologies involving language and communication such as machine translation, question-answer systems, machine learning and phrase structure parsing. Definitions of NLP can be specified differently depending on how it is to be used. In this thesis, NLP is defined in terms of a theory of computational techniques for analysing and representing naturally occurring texts at one or more levels of linguistic analysis in order to achieve human natural language communication for various tasks or applications (Liddy, 2001). NLP requires techniques for dealing with many aspects of language including syntax, semantics, discourse context and pragmatics (McGraw-Hill, 2002).

This chapter describes six analysis levels of natural language processing and explains how they may extend shape grammars to achieve context and harmonisation in design. Section 3.2 discusses the six analysis levels of NLP. Section 3.3 introduces the original contributions of NLP to shape grammars. Section 3.4 reviews the applications of natural language processing. Section 3.5 describes an extended shape grammar with two analysis levels of NLP called harmonised shape grammar.

3.2 The Six Levels of Natural Language Processing

Natural language processing (NLP) is the computerised approach to text analysis by using both a set of theories and a set of technologies (Liddy, 2001). To manage the complexity of language processing, linguists have defined six levels of analysis. The first level is of prosody which focuses on rhythm and intonation of language. The second level is of morphology which is concerned with the components (morphemes) of words and their meaning (Allen, 1995). The third level is of the lexicon which focuses on the meaning of words and their part(s)-of-speech (e.g. determiner, noun and verb). The fourth level is of syntax which is concerned with the analysis of words in a sentence and uncovering its grammatical structure. The output of this level of processing is a representation of the sentence revealing the structural dependency relationships between the words. The fifth level is of semantics which determines the possible meanings of a sentence by focusing on the interactions among word-level meanings in the sentence (Allen, 1995). For example, amongst other meanings, the word 'file' as a noun can mean either a folder for storing papers, or a tool to shape one's fingernails, or a line of individuals in a queue. To disambiguate the meaning of polysemous words requires consideration of the local context. This is provided by the sixth level, analysis of pragmatics. Two identical sentences may be interpreted differently depending on the overall context and other sentences in the surrounding text.

These levels are useful in dealing with ambiguities in language processing. In a language that has a large grammar and lexicon it may be difficult to choose a perfect interpretation for a given sentence (Russell and Norvig, 2010). Ambiguities can be found at many levels of analysis. Lexical ambiguity occurs when a word has more than one meaning. For example, the word "bear" may be a noun (an animal or stock exchange), or a verb (carry, endure, give birth to, or proceed in a specified direction). Syntactic ambiguity occurs with a phrase that requires multiple parses. The interpretation of the same sentence can be different depending on its syntactic structure. For example "John loves you and Mary too" can either mean John loves you and he also loves Mary, or both John and Mary love you. Semantic ambiguity, on the other hand, occurs when the syntactic structures remain the same but the individual words are interpreted differently. For example, let us consider these two sentences: 'Time flies like an arrow' and 'Fruit flies like bananas'. In the first sentence, the lexical term, flies, is a verb and in the second sentence it is a noun. The ambiguity lies in the fact 'flies' may be a verb or a noun. In order to decide which interpretation is correct, the overall meaning must be considered at the pragmatic level of analysis. However, ambiguities may also occur at the pragmatic level when the statement is not specific and the communication happens between two agents who do not share the same context. For example, "Do you want a cup of tea?" means either a question ("Do you feel a desire to have a cup of tea?"), or an indirect offer ("I can make you a cup of tea"). In order to understand indirectness, the relationship between the speakers, the place and situation, and the moment in time have to be considered (Harris, 1985).

3.3 Applications of Natural Language Processing

Originally, NLP was used for machine translation (MT) by taking the simplistic view that the only differences between languages reside in their vocabularies and the permitted word orders (Liddy, 2001). In 1954, the Georgetown-IBM experiment was the first public demonstration of machine translation (Hutchins, 1954). It involved automatic translation of Russian sentences into English. The project was judged to be successful, however the translations were limited to a number of sentences. Following in the same direction, SYSTRAN was created with more success in translating Russian to English for the United States Air Force during the Cold War (Toma, 1968). These were two of the oldest machine translation systems which were limited to translating between Russian and English texts. Decades later, machine translation systems were extended to more languages. Eurotra achieved translation between European languages (Debrock and Van Eynde, 1982). Anusaraka was able to handle translation between several Indian languages (Bharati *et al*., 1978). Mu, a system introduced by Japanese researchers, was one of the oldest machine translation programs to translate between Japanese and English (Nakamura *et al*., 1984) whereas KBMT has translated English to Japanese by using a knowledge-based approach (Nirenburg, 1989). In the past decade, statistical machine translation systems were greatly improved by using various statistical alignment models, called GIZA++, to train IBM models and Hidden Markov models (HMM) word alignment (Och and Ney, 2003). Another machine translation alignment was a phrasal inversion transduction grammar alignment technique, called Pialign, developed by Neubig *et al*., (2011).

As a machine translation decoder, Moses was a system that allowed users to automatically train translation models for any language pair (Koehn *et al*., 2007). This system could be trained with a collection of translated text and would use a search algorithm to find the most likely answer among all possibilities. Joshua was a parsing-based machine translation toolkit that implemented algorithms required for synchronous context-free grammar (SCFGs), aimed at translating between French and English (Li *et al*., 2009). An open source framework called Cdec was used for decoding, aligning with, and training a number of statistical machine translation models. The system separated model-specific translation logic from general rescoring, pruning, and inference algorithms by using a single unified internal representation for a translation forest (Dyer *et al*., 2010). A tree-to-string statistical machine translation system called Travatar was designed to be a syntax-based translation decoder (Neubig, 2013). It was most effective in translating between language pairs that required large amounts of recoding. A parsing algorithm for translating English to an Indian language was designed to be a machine translation system using tree adjoining grammar formalism (Nath and Joshi, 2014). This hybrid approach aimed at improving the performance or Early-type TAG by increasing the use of constraints and applying heuristic rules.

An automatic evaluation of machine translation such as RIBES aimed at developing high quality translation systems was developed by Isozaki *et al*., (2010). It proposed a rank correlation method to overcome issues in translating distant language pairs such as Japanese and English. Meteor used new metric features including text normalisation improved, precision paraphrase matching, and discrimination between content and function words (Denkowski and Lavie, 2011). The translation provided a high correlation with human judgments. Multeval was created as a statistical hypothesis testing for machine translation (Clark *et al*., 2011). The aims were to mitigate some of the possibilities of using unstable optimisers and to help in evaluating the impact of in-house experimental variations on translation quality.

In the early years, pattern matching techniques were used in natural language processing to create chatterbot which could talk like humans and exhibit intelligent behaviour. Systems such as ELIZA (Hutchins, 1954), PARRY (Colby, 1972), and Jabberwacky (Carpenter, 1997) were subject to the Turing Test to determıne whether users thought they were communicating with a human or to a machine. In fact, chatterbot did not understand the meaning of dialogues. However by using pattern matching techniques, the system could simulate convincing human-like conversations most of the time by asking more questions than it answered. The authors of ELIZA and PARRY focused on what answers machines should give to these questions.

LUNAR (Woods, 1977) was one of the oldest question-answer systems, designed to answer specific questions about moon rocks using knowledge structures, proposed by Schank and Abelson (1977). The knowledge structures of question-answer systems captured the coherent point of a narrative including thematic organisation packets (TOPs), story points, plot units, thematic abstraction units (TAUs), and planning advice themes (PATs) (Mueller, 2002). Wilensky (1982) created UNIX Consultant (UC), which allowed users to be able to communicate with the UNIX operation system in ordinary English. Instead of remembering the syntax of the commands, users could request the system in English to execute commands such as delete, move and create files, or browse system directories.

Much natural language processing was accomplished by using machine learning techniques. Trados was a successful computer-assisted translation system (Hummel and Knyphausen, 1984) and achieved a major success in 1997 when it was used for internal localisation by Microsoft (Ignacio, 2005). SVM-light was developed at the University of Dortmund by Joachims (2008) using supported vector machines with a fast optimisation algorithm. LIBLINEAR (Fan *et al.*, 2007) was created at the National Taiwan University for learning support vector machines and large linear classifications. These applications supported classification and regression analysis and were successful as open source machine learning libraries. CRF++ was a Japanese Morphological Analysis implementation created by Kudo (2004). It was designed for generic purpose that could be applied to various tasks of natural language processing. CRF suit (2007) and Classias (2009) created by Okazaki were fast training and tagging machine learning implementations and collections of machine learning algorithms for classifications. MALLET was created at University of Massachusetts Amherst by McCallum (2002), using hidden and maximum entropy Markov models, and conditional random field algorithms.

Sekine and Collins (1997) developed Evalb, an evaluating parsing accuracy tool using phrase structure parsing techniques. It was a bracket scoring program that reported precision, recalled, F-measure, non-crossing and tagging accuracy for any given data. Charniak (2000) developed the Charniak PCFG parser, a discriminative CFG parser for English. Socher *et al.* (2013) developed the Stanford parser (Socher *et al*., 2013), an application that worked out the grammatical structure of sentences. Although it still made some mistakes it was reliable in most cases. A token-order-specific naïve Bayes-based machine learning system (Rap TAT) was developed and evaluated by Gobbel *et al*. (2014). The aim was to create a machine learning system for concept mapping of phrases for medical narratives to predict associations between phrases and concepts. This approach improved performance in terms of accuracy and speed accomplished by the tool.

In language modelling, SRILM was developed by Stolcke *et al*. (1999) working as a statistical language models builder. It was used in speech recognition, statistical tagging and segmentation, and machine translation. The system was still under development and could be used on both UNIX and Windows platforms. IRSTLM was developed by Federico *et al*. (2008) and had many features including algorithms and data structures. The application was compatible with other language modelling tools such as SRILM. The Kyoto language modelling toolkit was developed by Neubig and Yao (2009) known as Kylm. It was capable of comparing the effectiveness of various types of language models including those for Japanese. KenLM was developed by Heafield (2011), one of the recent language modelling toolkits. The application had many technical improvements including executing faster and lower memory requirements compared to SRILM and IRSTLM, allowing multi-core processors, and on-disk estimation with user specified RAM. Brychcin and Konopik (2014) improved language modelling by building semantic spaces. The uses of semantic spaces and clustering to create a class-based language model with a standard n-gram model could create an effective language model. Their results achieved the reduction of the complexity and the improvement of the n-gram language model accuracy without the requirement of any additional information.

In speech recognition, Lee *et al*. (2001) introduced Julius that could perform large vocabulary automatic speech recognition with a large vocabulary. Moore *et al*. (2006) developed Juicer, a weighted finite state transducer based automatic speech recognition decoder.

As regards finite state models, Riley *et al*. (2007) developed OpenFst as a library to construct, combine, optimise, and search weighted finite-state transducer (FSTs). Neubig developed Kyfd (2009), the Kyoto Fst Decoder. It was a decoder for text-processing systems building. The system also used weighted finite-state transducers. It was used in many applications such as statistical machine translation, speaking style transformation, and speech segmentation.

As regards general natural language processing libraries, Bird *et al*. (2009) developed NLTK, a NLP implemented in Python to work with human language data. It included more than 50 corpora and lexicon resources and text processing libraries suitable for classification, tokenisation, stemming, tagging, parsing, and semantic reasoning. Kottmann *et al*. (2010) developed Apache OpenNLP library, a machine learning based toolkit to process natural language text. It supported most natural language processing tasks and also included maximum entropy and perception based machine learning. Manning *et al*. (2010), in the same year, developed Stanford CoreNLP, a library including many natural language processing tools developed at Stanford University. It was capable of giving the base words forms, parts of speech, and normalised date, time and numeric quantities from raw English text.

For over half a century, research has been conducted in several other areas of natural language processing and mostly focused on texts and phrases; in particular SHRDLU for natural language understanding (Winogard, 1971), LIFER system for natural language processing interfaces to databases (Hendrix, 1977), Racter for template system text generation (Chamberlain and Etter, 1983), MaltParser for data-driven dependency parsing (Nivre *et al*., 2006), MSTParser for dependency parsing (McDonald *el al*., 2006), Kytea for morphological analysis (Neubig *et al*., 2013), and Mpaligner for pronunciation estimation (Kubo *et al*., 2013).

The uses of natural language processing were not limited to just texts and phrases generation. Probabilistic grammars for music were introduced by Bod (2001). The research investigated the use of probabilistic parsing techniques in order to create musical parsing. Natural language processing of lyrics was introduced by Mahedero *et al*. (2005) suggesting the possibility of using NLP to generate song lyrics. However, this literature review has not been able to identify any natural language processing systems that design or implement shape grammars.

The major NLP implementations are summarised in Table 3.1 below.

Table 3.1: *Natural Language Processing Applications*

No	Name	Author(s) and Year	Purpose	Type
1	Georgetown-IBM Experiment	Hutchins, 1954	Translation of Russian sentences into English	Machine translation
2	SYSTRAN	Toma, 1968	Machine translation system	Machine translation
3	Anusaraka	Bharati *et al.*, 1978	Translation between several Indian languages	Machine translation
4	Eurotra	Debrock and Van Eynde, 1982	Translation between nine European languages	Machine translation
5	Mu	Nakamura *et al.*, 1984	Machine translation system between English and Japanese	Machine translation
6	KBMT	Nirenburg, 1989	Knowledge in machine translation	Machine translation
7	GIZA++	Och and Ney, 2003	A standard tool for creating word alignments using the IBM models	Machine translation alignment
8	Pialign	Neubig *et al.*, 2011	A phrase aligner based on inversion transduction grammars that can create compact but effective translation models	Machine translation alignment
9	Moses	Koehn *et al.*, 2007	A popular statistical machine translation decoder that supports phrase-based and tree-based models	Machine translation decoder
10	Joshua	Li *et al.*, 2009	A decoder implementing syntax-based translation	Machine translation decoder
11	Cdec	Dyer *et al.*, 2010	A parsing-based decoder implementing tree and forest translation	Machine translation decoder

12	Travatar	Neubig, 2013	A tree-to-string decoder for syntax-based translation	Machine translation decoder
13	RIBES	Isozaki *et al.*, 2010	An evaluation measure the accuracy of word recording	Machine translation evaluation
14	Meteor	Denkowski and Lavie, 2011	A tool for the Meteor metric, which performs accurate evaluation using a number of methods such as synonym regularisation, stemming, and considering recordering	Machine translation evaluation
15	Multeval	Clark *et al.*, 2011	A tool for evaluating machine translation results which considers the statistical significance of results for several evaluation measures	Machine translation evaluation
16	A Parsing Algorithm for English to an Indian Language	Nath and Joshi, 2014	A machine translation system using tree adjoining grammar formalism	Machine translation
17	ELIZA	Weizenbaum, 1966	A simulation of a Rogerian psychotherapist	Pattern matching
18	PARRY	Colby, 1972	Chatterbot	Pattern matching
19	Computing Machinery and Intelligence	Turing, 1950	Computing machinery and intelligence	Turing test (TT)
20	Jabberwacky	Carpenter, 1997	Computing machinery and intelligence	Turing test (TT)
21	LUNAR	Woods, 1977	Answering specific questions about moon rocks	Question-answer system
22	The Knowledge Structures	Schank and Abelson, 1977	Story understanding	Question-answer system
23	UNIX	Wilensky, 1982	Communication with users in a dialogue to provide suggestions	Question-answer system
24	Trados	Hummel and Knyphausen, 1984	Develop and market translation memory technology	Machine learning
25	LIBSVM	Chang and Lin, 2000	A full-featured package for learning support vector machines	Machine learning

26	MALLET	McCallum, 2002	A machine learning package for use in natural language processing. It implements hidden Markov models, maximum entropy Markov models, and conditional random fields. Written in Java	Machine learning
27	CRF++	Kudo, 2004	An implementation of conditional random fields, a standard sequence prediction methord. Can be customised with feature templates	Machine learning
28	CRF suite	Okazaki, 2007	A very fast implementation of conditional random fields	Machine learning
29	LIBLINEAR	Fan *et al*., 2007	A library implementing linear support vector machines and logistic regression. Training is extremely fast.	Machine learning
30	SVM-light	Joachims, 2008	An efficient SVM library	Machine learning
31	Classias	Okazaki, 2009	A library implementing many different kinds of classifier algorithms	Machine Learning
32	AROW++	Dredze *et al*., 2012	An implementation of adaptive regularisation of weight vectors, an online learning algorithm	Machine learning
33	A token-order-specific naïve Bayes-based machine learning system (Rap TAT)	Gobbel *et al*., 2014	A machine learning system for concept mapping of phrases for medical narratives to predict associations between phrases and concepts	Machine Learning
34	Evalb	Sekine and Collins, 1997	A tool for evaluating parsing accuracy	Phrase structure parsing
35	Charniak PCFG Parser	Charniak, 2000	A discriminative CFG parser for English	Phrase structure parsing
36	Stanford Parser	Socher *et al*., 2013	A parser that can output both CFG parses and dependencies. Can parse English, Chinese, Arabic, French, and German	Phrase structure parsing

37	SRILM	Stolcke *et* al., 1999	An efficient n-gram language modelling toolkit with a variety of features. A variety of smoothing techniques (including Kneser-Ney), class based models, model merging, etc.	Language modelling
38	IRSTLM	Federico *et al.*, 2008	A toolkit for training and storing language models	Language modelling
39	Kylm	Neubig and Yao, 2009	A language modelling toolkit that allows for weighted finite state transducer output and modelling of unknown words	Language modelling
40	KenLM	Heafield, 2011	A tool for memory and time-efficient storage of language models	Language modelling
41	Semantic Spaces for Improving Language Modelling	Brychcin and Konopik (2014)	The use of semantic spaces and clustering to create a class-based language model with a standard n-gram model	Language modelling
42	Julius	Lee *et al.*, 2001	An open-source decoder for large vocabulary automatic speech recognition	Speech recognition
43	Juicer	Moore *et al.*, 2006	A WFST-based speech recognition decoder	Speech recognition
44	OpenFst	Riley *et al.*, 2007	A library implementing many operations over weighted finite state transducers (WFSTs) to allow for easy building of finite state models	Finite state models
45	Kyfd	Neubig, 2009	A decoder for text-processing systems build using weighted finite state transducers	Finite state models
46	NLTK	Bird *et al.*, 2009	A general library for NLP written in Python	General NLP libraries
47	Apache OpenNLP	Kottmann *et al.*, 2010	A library written in Java that implements many different NLP tools	General NLP libraries
48	Stanford CoreNLP	Manning *et al.*, 2010	A library including many of the NLP tools developed at Stanford University	General NLP libraries
49	SHRDLU	Winograd, 1971	Working in restricted blocks worlds	Natural language understand-ing

50	LIFER system	Hendrix, 1977	Adapting existing computational linguistic technology to practical applications	Natural language interfaces to databases
51	Racter (raconteur)	Chamberlain and Etter, 1983	Generated English language prose at random	Template system text generation
52	MaltParser	Nivre *et al.*, 2006	A parser based on the shift-reduce method. Inducing a parsing model from Treebank data and to parse new data using an induced medel	Data-driven dependency parsing
53	MSTParser	McDonald *et al.*, 2006	A tool for dependency parsing based on maximum spanning trees	Dependency parser
54	Kytea	Neubig *et al*., 2013	A tool for word segmentation and morphological analysis that is relatively robust for unknown words and easily domain adaptable	Morphologic al analysis
55	Mpaligner	Kubo *et al*., 2013	A program for aligning graphemes to phonemes for training pronunciation systems, mainly for use with Japanese	Pronunciatio n estimation
56	Probabilistic Grammars for Music	Bod, 2001	Investigating the use of probabilistic parsing techniques for musical parsing	Probabilistic parsing
57	Natural Language Processing of Lyric	Mahedero *et al*., 2005	Using standard NLP tools for analysis of musical lyrics	Lyrics processing

3.4 Natural Language Processing Contribution to Shape Grammar

Natural language processing provides both a theoretical underpinning and a basis for system implementation for a range of applications. In fact, any application that utilises text is a candidate for it (Liddy, 2001). The capability mostly limits the researchers to language analysis, language recognition, and text utilisation. However NLP can be extended to a new area of application namely shape grammar. One can draw a parallel between the shape grammar structure and the natural language analysis levels. Shape grammars use points, lines, and planes. Words can be combined by natural language grammar rules to form a well-formed sentence and similarly shapes can be combined using spatial and transformational rules to produce complex shapes. NLP can provide a promising theoretical underpinning to shape grammar, and can help generate meaningful shapes, contextually relevant and harmoniously designed.

This thesis proposes a harmonised shape grammar paradigm inspired from the field of natural language processing making use of its important levels of analysis: lexical, syntactical, semantic, and pragmatic levels (Fig.3.1). These levels may be mapped to a shape grammar where the smallest unit of a shape consists of points, lines, and planes. These primitive shapes may be combined to produce a lexicon of complex shapes (e.g. cubes, cylinders, and spheres). Syntax rules define the transformations to assemble these shapes into more complex shape. Semantics give a meaning to these shapes and provide a context and a direction to the final design. Finally, context and harmony can be achieved through pragmatics. Shapes do not have individual meaning but have to be evaluated according to the overall design specifications and goals.

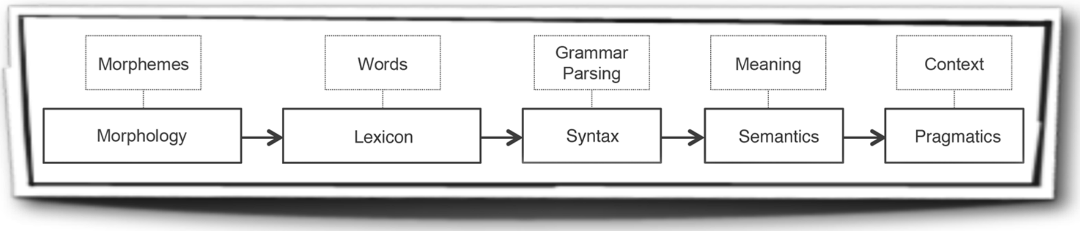


Fig.3.1 *Natural Language Processing Model*

The proposed shape grammar framework will extend the traditional shape grammar by adding *morphology, context,* and *harmony* to its two original levels: *vocabulary,* and *shape rules*. This framework will be validated by applying it to produce a set of harmonised humanoid[1] characters. To achieve harmony in character design, the five levels are to be embedded in the generation engine. In Fig.3.2, the *shape basic element is* the smallest unit used to create shapes such as points, lines, and planes. The *vocabulary* or lexicon is made up of these shape elements and subjected to a set of syntactic rules to constrain the possible spatial and functional transformations specific to the design object; the humanoid body characters in our case study. These transformations will be interpreted by a semantic model embedded in *context* to

[1] *A humanoid is a being having human form or human characteristics.*

ensure legitimacy, consistency, and compatibility in character design. To achieve *harmony*, legitimate shapes and elements of the objects must adhere to certain contextual properties and purposes of the design; for example, each generated character should be consistent and compatible with its siblings and parents, in terms of physiognomy, functionality, habitat, and personality.

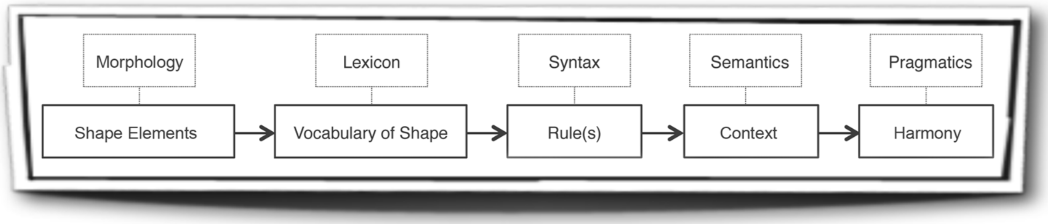


Fig.3.2 *The Integration of Shape Grammar and Natural Language Processing*

Early work in shape grammars took a top-down approach because CAD systems offered primary top-down support. However, the main advantage of a bottom-up design which is adopted in this research, is the ability to reflect the designer's thinking in generating a new design which is context sensitive and adaptive (Talbott, 2007). This approach can also augment the designer's visual tie to produce an evolving character.

This research will further explore the field of ontology to achieve context and harmony in character design by capturing the anatomy, function and organisation of the humanoid world as well as the hierarchical and contextual relationships among the characters. For example, to create a humanoid family, shape rules will apply a set of transformations on the primitive vocabulary (e.g. points, lines, planes) to create components of a body (e.g. head, body, arms, legs using basic shapes) consistent with the ontological representation of a humanoid body. The assembling of these components are to be governed by a semantic model which dictates their spatial relations, size, weight and height, and habitat (e.g. terrestrial, subaquatic, celestial). Context is to check that the humanoid character is harmonious with other members of the humanoid family, in terms of its morphological and functional structure. Context is to also examine the cohesion and coherence of humanoid features, and pragmatics is to focus on the harmony issue and on the combination of various elements to emphasise their similarities with other characters and bind the picture parts into a whole. These aspects, namely spatial relations, morphology, cohesion, coherence, and harmony, are to be used as measuring criteria to validate and evaluate the proposed framework. The ontological representation of harmonised humanoid characters is explained in details in chapter 4.

3.5 Summary

This thesis discusses the integration of shape grammar with natural language processing in order to bring harmonisation, contextual information, and meaning to design shapes by validating this novel approach using the humanoid design characters as a case study. The novel harmonised shape grammar framework is a novel application of NLP and extends the existing shape grammar.

Table 3.2: *Shape Grammar, Natural Language Processing, and Harmonised Shape Grammar Levels of Analysis*

Shape Grammar	Natural Language Processing	Harmonised Shape Grammar
X	Morphological level	Shape elements
Vocabulary of shape	Lexical level	Vocabulary of shapes
Shape rules	Syntactic level	Shape rules
X	Semantic level	Context
X	Pragmatic level	Harmony

The novel shape grammar framework focuses on five analysis levels of natural language processing, namely the morphological, lexical, syntactic, semantic, and pragmatic level in order to generate harmonious character design objects. The morphological level contains the smallest unit of shapes, the lexical level includes vocabulary of shapes, the syntactic level consists of rules to generate complex shapes, the semantic level generates meaningful and contextual characters, and the pragmatic level delivers harmonisation (Table.3.2).

To validate the new shape grammar framework, a set of humanoid characters is to be defined and an ontological representation of what constitutes a humanoid character is to be developed to capture morphological, lexical, syntactical, semantic and pragmatic levels of analysis. The goal is to achieve context and harmony by capturing the essential features of the humanoid world. Consequently, the morphological level is to consist of the smallest units of shapes (points, lines, and planes), and the lexical level is to produce primitive geometrical shapes (e.g. polygon cube, cylinder, and sphere) to defines the anatomy/physiognomy of humanoid characters. The syntactic level is to apply spatial rules (translating X, Y, and Z) and transformational rules (scaling, extruding, and merging) and domain grammar to manipulate these shapes in agreement with the ontological definition of a humanoid body. For example, the syntactic rules are to manipulate the first polygon vocabulary to design the first component (head) and develop the second vocabulary to create the second, third and fourth components (body, arms and legs), the assembling of these design components are then to be refined by the semantic level, acting as the derivation phase, and dictating the humanoid anatomy, body function, and habitat. The generated humanoid character has to be harmonious with other members of the humanoid family, in terms of its design attributes, characteristic, and behaviour. This is to be achieved at the pragmatic level which ensures that the final humanoid character design meets aesthetic criteria, context, and harmony in agreement with the design principles. For example, context focuses on cohesion and coherence of humanoid features whereas harmony is to combine various elements so as to emphasise similarities with other humanoid characters and bind the picture parts into a whole. Further description of the application of the proposed new shape grammar is included in chapter 4.

Chapter 4: Harmonised Shape Grammar

4.1 Introduction

The main contribution of this thesis is the development of a novel shape grammar framework which is able to generate harmonious and contextual-based design objects. The previous chapters highlighted the limitation of shape grammars; in particular, they were designed for static models only and confined to geometric and spatial derivation rules (Ilčík *et al*., 2010). Despite some of the design principles and transformational rules, shape grammars cannot guarantee aesthetic results (Huang *et al*., 2009). By adapting natural language processing levels of analysis to shape grammars they can be extended to achieve context and harmony.

In his thesis, Lee (2006) explained that shape grammars have been used to generate stylistically consistent and novel designs in computer aided design (CAD) systems. However, to achieve this consistency designers were involved in controlling the parametric modifications of the shapes and their grammatical rules. They also contributed to the evaluation of the design. As the generation of shape grammars for specific design requirements is complex and time consuming, he developed a systematic approach to the formulation of shape grammars and applied an evolutionary algorithm for his interactive grammar-based design system. His experiments demonstrated the flexibility of his approach to the design of digital cameras.

Chapter 2 also identified that most research in shape grammars has focused on geometric properties, and architectural and product design. There is a growing need to extend the application of shape grammars to design characters in computer games, which is becoming the fastest growing sector worldwide. This research is an attempt at addressing these issues by focusing on the application of shape grammars to character design. The aim is to extend existing shape grammars to address harmony and context in design, in particular, by extending these grammars to include semantic and pragmatic analysis. The generation of 3D humanoid characters is used to validate the proposed framework. The semantic level of analysis is to allow the generation of meaningful humanoid character designs each with humanoid anatomy appropriate to body function, and design attributes relevant to the contextual environment. To achieve harmonisation between all humanoid characters, aspects such as personality, behaviour and environment should be clearly identified. These can be embedded in the pragmatic level of analysis.

4.2 Context and Harmonious Design

In his book, Stiny (2006) discusses how grammars of visual arguments[1] may be used to describe and construct shapes by means of a formal algebra. He compares shape grammars with Noam Chomsky's verbal grammars. In addition to aesthetics and visual appeal, shape grammars focus on syntax just like other grammars. The basic geometrical shapes represent the vocabulary of shape grammars, and the transformational rules consist of a set of spatial design transformations which are to be applied to produce new shapes. In most applications, it is left to the user to guide the selection of the rules in order to meet the design objectives. However, there are many choices of rules which may create emergent properties corresponding to several different conditions and objectives. To resolve these issues, researchers have applied artificial intelligence techniques to control the selection of the rules. Shea and Cagan (1999) applied shape annealing to produce optimal designs. Explicit domain knowledge was included with the grammatical rules and syntax and design interpretation was used to select forms that fulfil functional and visual goals. O'Neil *et al.* (2009) combined genetic programming with a shape grammar to encode human domain knowledge to rediscover known benchmark target structures.

Although these approaches can handle complex 3D objects, they are better at generating free-forms or unpredictable shapes than at fulfilling specific harmonised objects. These approaches may be acceptable to engineering design and product design; however, in computer games and movies applications, they cannot provide rich semantic and harmonised design characters. The generation engine of a shape grammar requires a deeper level of analysis to combine syntax with both semantics and pragmatics in order to generate a harmonised set of characters. Context and

[1] Visual argument means a way to create an argument using visual image

harmonisation aim to bring a realistic feeling to the audience to suggest that the characters come from the same world or same story. Any non-harmonised character should be considered as an error in the design set and should be rejected. In order to achieve context and harmonisation, all characters should exhibit not only agreeable but also compatible characteristics with their function and role within their world. For example, characters that live in a subaquatic environment should have fins to assist their movement under water and wings should be provided for celestial characters to fly above ground.

4.3 Character Design Overview

Character design plays an important role in many fields including 2D and 3D animation, graphic novels, game design, and toy product design. Humanoid characters are used in many application areas of computer graphics namely simulation, computer games and movies (Preda *et* al., 2006). Due to the growing research interest in character animation and motion capture in recent years, techniques such as motion editing, key frames and motion capture have been developed to provide all aspects of live motion designed to enhance animation (Pullen and Bregler, 2002). In the last decade, the development of digital media technology has advanced significantly and will enhance the design of 3D characters, specifically in 3D photorealistic and 3D animation character creation. The high quality texture, material, and images can bring 3D character and animation scene design one step closer to a realistic result.

4.3.1 Humanoid Character Design

Artists and designers use shapes, sizes, poses and proportions to represent the physiognomy and personality of a character. These also help define the role played by the character. By looking at the design, the audience should be able to understand the characters' context (profile and background) such as its humanoid anatomy, body function, and habitat. The design process starts with defining basic shape and related proportions. Harmonisation is created using design attributes (e.g. colour, skin tones, texture and material of skin), characteristics (body size, body weight), and behaviour (personality and character expression). Designs vary widely across different cultures, media, genres, target groups and design purposes. Hard work and years of learning and practice are required for artists to put appropriate contexts together with physiognomy in order to achieve impactful characterisation for a target audience. With the high demand for animation and computer games, computers are used mostly for animation and shape-modelling. However, the conceptual character design aspect still relies heavily on the skills and experience of human artists and designers. Systems that could abstract character design rules from finished art would thus be a really useful tool to designers (Islam *et al.*, 2010).

The design and generation of humanoid characters involves many modelling aspects, some of which relate to object modelling and manipulation of primitive shapes. These aspects are described in the next sections.

4.3.2 Object Modelling Techniques

In traditional object modelling, only information about the geometry of an object is stored, mainly in the form of mathematical equations of surfaces (Bronsvoort *et al.*, 2010). The positional boundary of points, lines, planes, or curves along the X, Y and Z axes is used to calculate the coordination of the object. In terms of object modelling techniques, the terms refer to both surface modelling and solid modelling.

In surface modelling, known as Non-Uniform Rational B-Spline (NURBS) modelling, only the geometry of surfaces is represented by mathematical equations known. Surface modelling deals primarily with parametric surface patches (Sederberg and Parry, 1986). This modelling technique has encompassed either creating aesthetics or performing functions. The advantage of surface modelling over solid modelling is that it offers much more flexibility in modelling free-form units (Sapidis, 1994). The drawing data is stored as points which form the arbitrarily curved line when there are two or more points to coordinate, and uses the curves data to create a curved surface if possible (Bronsvoort *et al.*, 2010). Compared to solid modelling, surface modelling produces greater smoothness and with fewer points for a curved line for round and cornerless objects. Thus, surface modelling is better for modelling biogenic objects such as plants and animals (Lorensen and Cline, 1987) and soft surface objects such as cloth or liquids. It also has many uses in the field of engineering and product design to model (Fiorentino *et al*., 2002), for example, vehicle parts, gas turbine blades and fluid dynamic engineering components.

In solid modelling, also known as polygon modelling, the geometry of a solid 3D objects is represented by a boundary representation which consists of vertices, edges and faces. It is to be distinguished from related areas of geometric modelling and machine design (LaCourse, 1995). The vertices data are described by their coordinates along the X, Y and Z axes as the implicit representation where the faces and edges are described by their mathematical relationship with the vertices. The boundaries between vertices, edges and faces are stored in a graph structure and can be converted into shapes using polygonisation algorithms. The solid modelling is parametric and feature-based and is associated with geometric attributes including intrinsic geometric parameters (height, weight and depth), position and orientation, geometric tolerances and material properties (Mantyla *et al*., 1996). The principles of solid modelling form the foundation of computer-aided design and are used widely to support the creation of digital models of physical objects including low-detail objects and humanoid characters. Solid modelling can give a full support for UV mapping coordination which is a necessity for character modelling. This thesis focuses on solid modelling techniques because of its UV mapping advantages and the automation of several difficult calculations which are to be carried out as a part of the design process.

4.3.3 Shape Basic Elements

The primitives in shape grammars are shapes rather than symbols, so their relationships are spatial rather than symbolic. Any system, which uses shape grammars or shape algebraic theory, includes a number of concepts which have to be defined including spatial dimension, elements, relations and properties. In addition, other specific concepts such as sub-part relationships and shape boundaries have to be defined explicitly (Emdanat and Vakalo, 1998). Shapes are the boundaries of four basic elements including points, lines, planes and solids, and may be described as vertices, edges, faces and objects in 3D computer-aided design media. The basic elements are readily described by the linear relationships of coordinate geometry. It is not necessary to extend this repertoire to include other curves and exotic surfaces, especially when these are described analytically (Stiny, 2006). As mentioned in Stiny's book (2006), whether further kinds of these basic elements are allowed or not, the results are approximately the same. Our research focuses on the four basic elements to be considered in shape algebras. The key properties of basic elements may be summarised as shown in Table 4.1 (Stiny, 2006). The corresponding attributes namely, dimension, boundary, content, 3D medium and embedding property, are identified by the relationship between each basic element. A Point is the indivisible unit. It has zero dimensions, no boundary, no content, and is the smallest element of the basic elements. Other basic elements are all divisible into smaller discrete units. A line may be cut down into points, a plane may be segmented into lines, and a solid may be split into planes. The number of dimensions, 0, 1, 2 or 3, shows the relationship, boundary or amount of cohesion between them.

Table 4.1: *Properties of Basic Elements (Stiny, 2006)*

Basic Element	Dimension	Boundary	Content	3D Medium	Embedding
Point	0	None	None	Vertex	Identity
Line	1	Two points	Length	Edge	Partial order
Plane	2	Minimum of three lines	Area	Face	Partial order
Solid	3	Minimum of four planes	Volume	Object	Partial order

4.3.4 Shape Algebras

A shape is defined with a non-zero dimension. It may be as simple as a 1D line shape to a high complexity of a 3D solid object, as long as it has dimension(s). The construction of two separate shapes may share one or more but not all their basic elements. Fig.4.1 shows two squares of different sizes sharing two points (point a, and point b), and a line (line ab). They are considered as two separate shapes as long as the shared line ab is included, but as only one shape without the ab line.

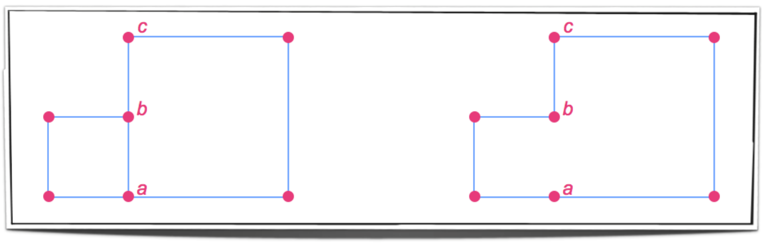


Fig.4.1 *An example of two shapes that share some basic elements together*

Algebras of shapes are defined by three constituent elements: the shapes themselves which contain the properties of basic elements, part relationships for shapes which include Boolean operations, and Euclidean transformations (Stiny, 2006). Table 4.2 shows the formalisation between basic elements and the boundary of shapes and parts. Shapes, normally, have up to three dimensions and may be described in the U_{ij} algebra. Index i represents the dimension of the basic element. For example, U_{0j} refers to point-based shapes, U_{1j} refers to line-based shapes, U_{2j} refers to plane-based shapes and U_{3j} refers to solid shapes. The index j represents the dimension of the shape as a result of combination of the basic elements. For example, U_{i1} allows the basic elements to combine into a shape of one dimension, U_{i2} allows them to combine into a shape of two dimensions, and U_{i3} allows the three dimensions formulation.

Table 4.2: *Properties of Shapes (Stiny, 2006)*

Algebra U_{ij}	**U_{i0}**	**U_{i1}**	**U_{i2}**	**U_{i3}**	**Basic Elements**	**Boundary Shapes**	**Number of Parts**
U_{0j}	U_{00}	U_{01}	U_{02}	U_{03}	Points	None	Finite
U_{1j}		U_{11}	U_{12}	U_{13}	Lines	U_{0j}	Indefinite
U_{2j}			U_{22}	U_{23}	Planes	U_{1j}	Indefinite
U_{3j}				U_{33}	solids	U_{2j}	Indefinite

Table 4.3 shows an example of basic element formalisation using the U_{ij} shape algebra. The minimum of index i can be equal to zero and maximum of three, where the value of index j may not be lower than the value of index i in the same shape. The shape space is based on the allowable dimensions of index j. For example, when the value is zero, there is no coordinate axis. The shape has no dimensions or arrangements thus only a single point is possible. When index j is one, the collinear arrangements of shapes along one axis are possible. This allows the creation of points and lines as long as they are collinear. When index j is two, there may be two coordinate axes. The coplanar arrangements of points, lines and planes are possible. When index j is three, there may be three coordinate axes. The use of three dimensions enables the creation of points, lines, planes, and solids.

Table 4.3: *The U_{ij} Shape Algebra Example*

Algebra U_{ij}	Shape Example	Axis/Axes	Algebra U_{ij}	Shape Example	Axis/Axes
U_{00}		N/A	U_{22}		
U_{01}			U_{03}		
U_{11}			U_{13}		
U_{02}			U_{23}		
U_{12}			U_{33}		

4.4 Proposed Harmonised Shape Grammar Framework

Early work in shape grammars took a top-down approach because a CAD system offered primary top-down support. However, the main advantage of a bottom-up design, which is adopted in this research, is the ability to reflect the designer's thinking in generating a new design that is context sensitive and adaptive (Talbott, 2007). This approach is better at supporting the design of evolving characters.

This thesis advocates the integration of shape grammars and natural language processing in order to bring harmonisation, contextual information and meaning to shape design. The design of humanoid characters will serve as an example. The proposed framework extends Stiny's shape grammar by adding new levels. It is to consist of five analysis levels: morphology, the lexicon, syntax, semantics and pragmatics. These levels help generate harmonious character design objects. Morphology is to consist of shape properties. The lexicon is to comprise a vocabulary of shapes. Syntax is to capture the grammar so as to assemble these shapes and construct a humanoid body. Semantics is to generate meaningful characters. Pragmatics is to deliver a family of harmonised characters with respect to their environment.

The proposed harmonised shape grammar framework is also based on object modelling techniques, described above, and the principles of harmony in design described in chapter 2. The process starts by preparing shape elements and workspace. After the preparation, it focuses initially on creating the primitive shapes which produce a vocabulary of shapes (e.g. cubes, cylinders, spheres). The next level applies a set of spatial rules (translate X, translate Y, translate Z) and transformational rules (scaling, extruding, merging) to manipulate and assemble these shapes to meet specific design goals. From the research, the development of shapes produces not only spatial transformations but also provides semantic information relevant to humanoid character design such as the character anatomy, body function components, and the habitat of humanoids. This level allows shapes to be developed meaningfully. If the resulting shapes are not semantically well-formed, the proposed algorithm is able to backtrack and reapply the semantic rules. The resulting character is processed further in the pragmatics level to achieve harmonisation. Pragmatic information allows vocabularies to be manipulated harmoniously and produce a humanoid with design attributes, characteristics, and behaviour suitable to their environment. This level ensures that appropriate design attributes for the humanoid environment are to be selected, for example, relevant colours, textures and materials. It also checks that appropriate characteristics relevant to the physical states of humanoids (e.g. size, weight), and behaviour associated with the relevant mental states (e.g. aggressive, grumpy, happy) are to be selected. The harmonisation in design is finally achieved when primitive shapes are both semantically and pragmatically well-formed (Fig.4.2).

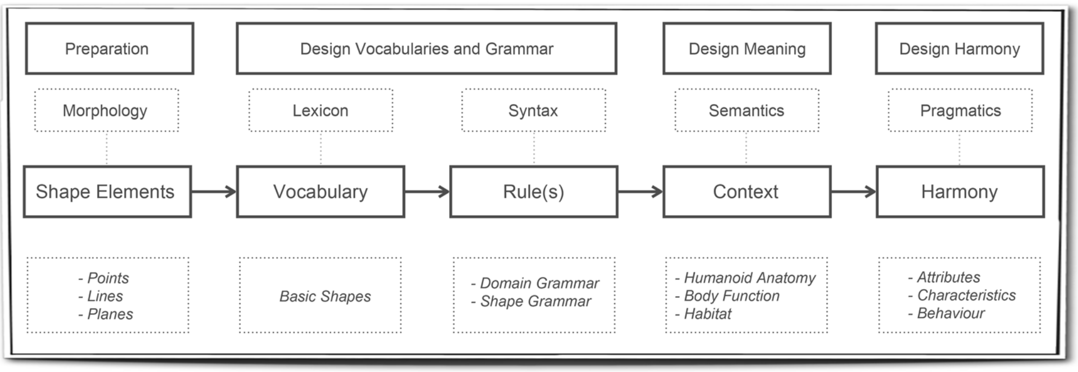


Fig.4.2 *Harmonised Shape Grammar Framework*

4.5 Ontology Based Representation of Humanoid Characters

This section introduces an ontological representation used to capture the complex characteristics of humanoid characters with respect to the five levels of analysis. The term, ontology, has a long history in philosophy, and is basically a description of the concepts and relationships that exists for an agent or community of agents (Gruber, 1993). It is used to support the sharing and reuse of knowledge represented in a formal manner. In this thesis, the definition adopted is that used by the artificial intelligence community which defines ontology as a specification of a representational vocabulary for the domain of humanoid characters expressed in terms of concepts, relations and functions (Fig.4.3). The ontological representation of humanoid characters consists of the following:

- *Morphological level*. This level captures the basic primitive elements of shapes: points (U_{0j}), lines (U_{1j}), and planes (U_{2j}) (see Table 4.2). According to the shape properties and the U_{ij} shape algebra as defined by Stiny (2006), a point (U_{0j}) is a singularity, the smallest component that cannot be broken down further. A single point or multiple points without boundaries between them are not a shape but just a unit which cannot be rendered. In order to create a shape, boundaries are required to associate points and other components. A line (U_{1j}) may be created by defining a boundary between a minimum of two points. It is a path that connects two points together, one as a starting point and another as an end point. Contrary to surface modelling, curve properties cannot be applied to polygon shapes. However, curve shapes can be achieved by creating parabolic sections inside polygons. Planes (U_{2j}) are the boundaries between lines. Lines have to form a closed loop in order to successfully form a plane. For example, two lines cannot form a plane because two lines create a "V" form (U_{12}) which is an open loop. In order to form a plane, a minimum of three lines are required to create a closed loop triangular plane.

- *Lexical level*. This level includes a vocabulary of shapes (U_{3j}): cube, cylinder and sphere. Shape elements from the morphological level are used to create basic 3D shapes (cubes, cylinders and spheres). These basic 3D shapes are considered as root shapes to be used as conceptual design materials in the creation of any desired shapes. It is difficult to create a large database to store all types of possible shapes, but it is feasible to create a small group of root shapes, and then modify them in such a way so as to produce complex shapes using appropriate transformational rules at the next level.

- *Syntactical level*. This level consists of two set of rules: shape grammar and domain grammar rules, which are to be applied to the vocabulary of shapes. In our proposed shape grammar, shape grammar rules focus on the spatial arrangement of the vocabulary of shapes to produce complex shape structures whereas the domain grammar rules apply transformations to produce humanoid components, namely the head, body and limbs. Instead of randomly generating meaningless shapes, the shape grammar rules generate syntactically well-formed shapes with appropriate proportional components for humanoid characters, where each shape has its own individual meaning. For example, a cube gives a feeling of stability, whereas a sphere gives a harmonious and calm feeling. However, when many spheres are assembled together to form a sphere tower, this gives an appearance of instability or insecurity. The appearance is achieved at the individual level and also at the level of a group of assembled shapes. In order to achieve an overall meaningful humanoid character design, these generated components (e.g. head, body, limbs) have to be further processed by the semantic and pragmatic levels.

- *Semantic level*. This level focuses on the context of the design, in particular, the design of character anatomy with respect to its function and habitat. The design of character anatomy focuses on the form and appearance of the design, whereas character body function and habitat focus on the humanoid's environment. Character anatomy should resemble a human in its shape and focus on the relationship between the components (head, body, limbs) generated at the syntactical level. Each component has its sub-members: a head (eyes, nose, ears, and mouth), a body (chest, torso and hip), upper limbs (arm, elbow, forearm, wrist and palm) and lower limbs (thigh, knee, leg, ankle and foot). These are connected by joint components: neck, shoulders and hip. Character body functions define the uses of each component. For example, eyes are to see, so they must be placed on the head; legs are to carry the weight of the body and used for walking if terrestrial, so they must be attached to the character body at the appropriate functional position. Character habitat focuses on three possible living conditions: terrestrial, subaquatic and celestial. Each type of habitat requires a specific set of humanoid anatomy and body functions. A terrestrial character requires legs to carry its body and walk. A subaquatic character requires fins and a tail to swim under water. A celestial character requires wings to fly above the ground.

- *Pragmatics level*. This level focuses on design harmony criteria: humanoid character design attributes, its characteristics, and its behaviour. Character design attributes includes colour, texture and material. Colour theory provides rules to achieve harmony in colour. Humans perceive a colour by comparing one colour to other surrounding colours. For example, blue and purple go well together, whereas blue and red may create colour assimilation and an unpleasant feeling. Textures and materials convey different feelings. Natural texture and material such as leather or cloth give to a design object a totally different feeling from chemical texture and material such as shiny metal or plastic. Character characteristics describe physical states which typically belong to a character, such as size and weight. Character behaviour describes mental states which can be expressed by facial expressions (e.g. neutral, friendly or depressed). To achieve harmony, the three criteria should be met. Colours should create a pleasant feeling and materials should be chosen with the appropriate texture. Characters should express relevant behaviour and characteristics within their environment and with their family members.

The ontological representation defines the design process as five levels of analysis: the morphological, lexical, syntactical, semantic and pragmatic level. By following these five levels of analysis, the contextual design and design harmonisation can be achieved. This ontological representation can be adapted to various purposes of design depending on vocabulary, shape rules, context and harmony criteria that are to be applied. This thesis uses humanoid characters as a case study to test the framework. The application and the evaluation of the proposed harmonised shape grammar are described in the next chapter.

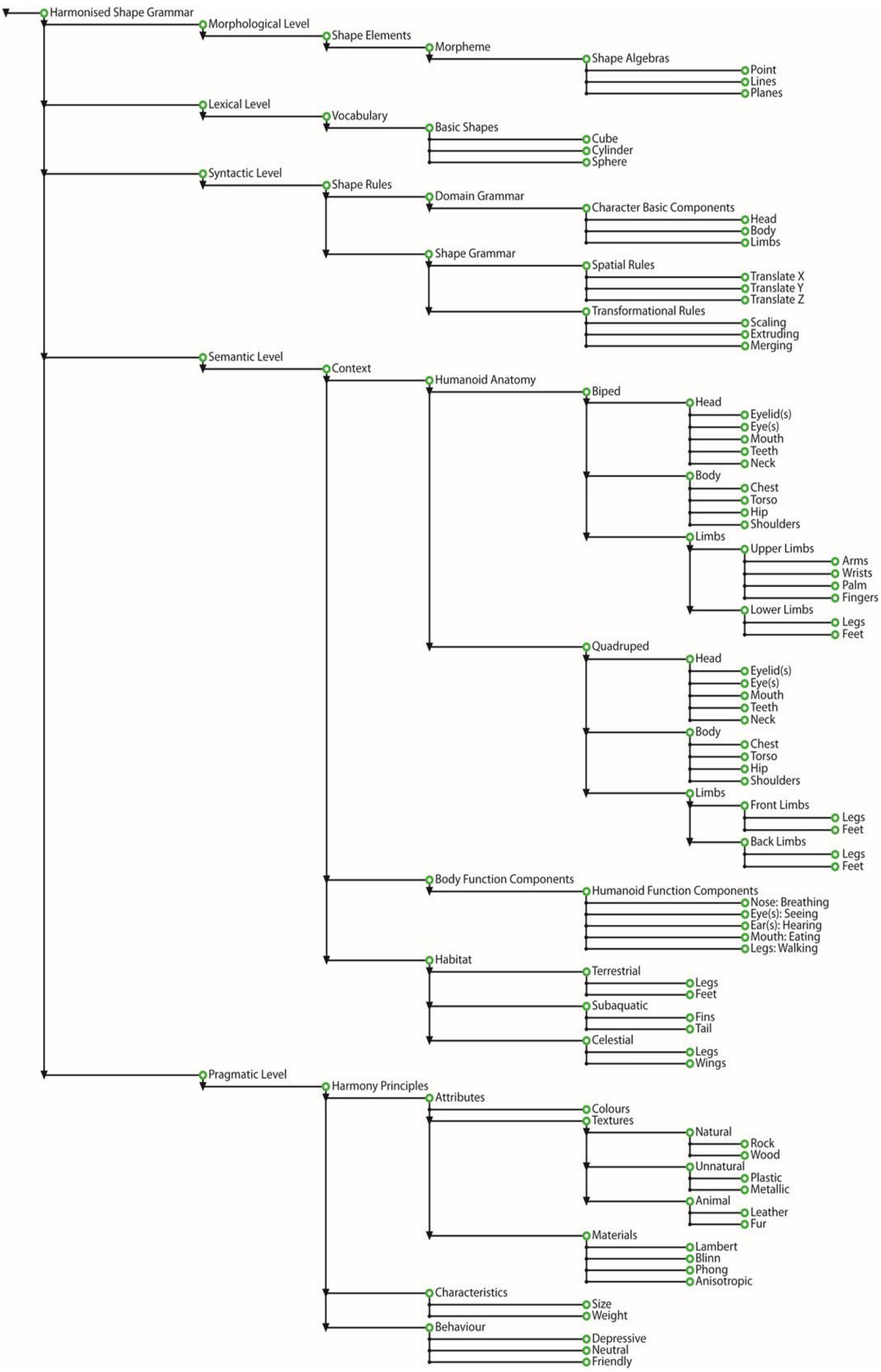


Fig.4.3 *The Ontological representation of Humanoid Characters*

4.6 Conclusion

Shape grammars have been used to generate stylistically consistent and novel designs in computer-aided design (CAD) systems by employing a vocabulary of shapes and shape rules to generate new shapes. Shape grammars are capable of creating new shapes effectively by using limited vocabularies. Most research in shape grammar has focused on geometric properties and architectural design. Recently, there are many research projects that aimed at extending the uses of shape grammars for other purposes (Halatsch, 2008; Lee and Tang, 2009; Huang *et al*., 2009; Ilčík *et al*., 2010; Noghani *et al*., 2013); however, there are some serious limitations as highlighted below:

- They are designed for static models.
- They are confined to geometric and spatial derivation rules.
- They cannot guarantee an aesthetic result.
- They lack context and harmony in design

To overcome these issues, the proposed shape grammar is to include two further levels of analysis: semantics and pragmatics. The semantic level focuses on the context of the design: design anatomy, design function and design habitat. This allows the generation of meaningful design to be created with a design anatomy which is appropriate to the design function, and attributes relevant to the contextual environment. The pragmatic level focuses on design harmony criteria: design attributes, characteristics, and behaviour. This allows the consideration of colours, textures, materials and expression to be applied in the design. This thesis focuses on the application of harmonised shape grammars to create character designs. To verify the proposed framework, an existing shape grammar is to be extended to address the harmony and contextual design issues. A set of 3D humanoid characters are to be created as a case study. The final proposed harmonised shape grammar is to comprise five levels of analysis: the morphological, lexical, syntactical, semantic and pragmatic levels. The morphological and the lexical levels redefine the shape grammar in term of its primitive shapes and vocabulary of shapes. The semantic and pragmatic levels are to specify, using rules, the design of a set of meaningful and harmonious characters.

Chapter 5: Experimental Study

5.1 Introduction

Harmonised shape grammar (HSG) is the basis for a framework which may be applied to different types of designs, from a simple creation of a slide show to something more complex such as generating stylish architectural building designs or computer games characters. Recently, the growth of digital entertainment industries has been increasing and character design engineers find themselves in high demand. However, modelling a character is a time-consuming process and requires a huge financial budget. HSG can help address the demand of high quality design and the duplication and rescaling of a character to build up a set of similar characters, physically and behaviourally. This chapter describes the implementation of HSG to produce a set of harmonious humanoid characters.

5.2 Case Study of Humanoid Characters

The design of humanoid characters is used as a case study to evaluate the proposed HSG. The term humanoid refers to a being having a human form or human characteristics. In this research, a humanoid character is described in terms of a human anatomy including three basic components: head, body and limbs. This case study aims at creating a 3D humanoid character. All shape elements must be handled in a 3D workspace. To this end, HSG applies the five levels of analysis derived from natural language processing: morphology, lexicon, syntax, semantics and pragmatics. Each level focuses on achieving specific harmonised design requirements. Fig.5.1 describes the system overview of HSG used to generate harmonised humanoid character.

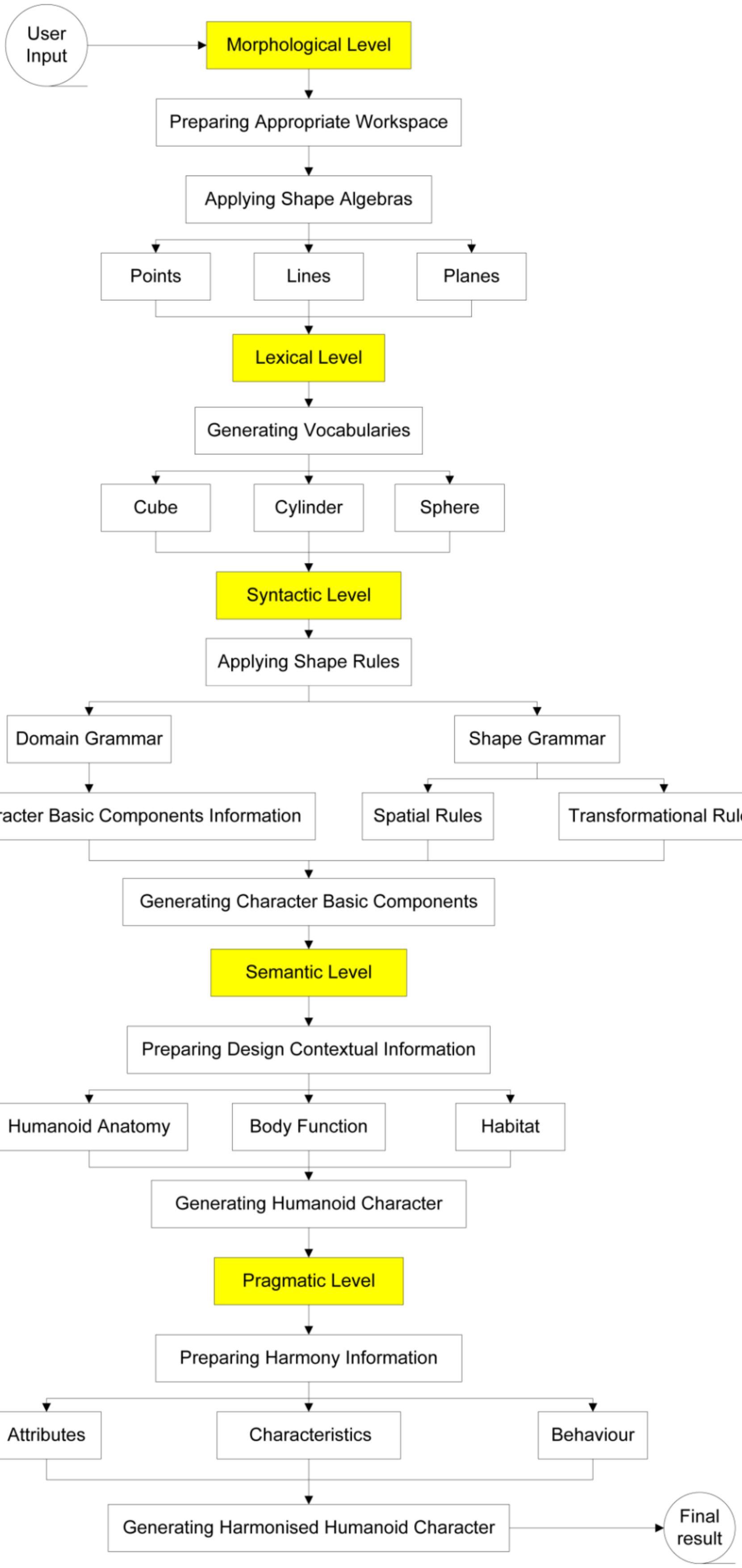


Fig.5.1 *The System Overview of HSG to Generate Harmonised Humanoid Character*

- **Morphological Level**

The implementation is based on the ontological representation of humanoid characters described in Fig.4.3. It starts with defining the appropriate shape elements at the morphological level. These include points, lines and planes which are the very basic elements of solid modelling. A point, line and plane may be referred to as a vertex, edge and face in modelling technical terms. However, to prevent confusion, this research adheres to the common terms as point, line and plane. There are two main differences in preparing 2D and 3D workspaces. The first difference is the measurement of coordinates. The coordinates of a 2D workspace have X and Y axes to represent height and width of shapes whereas 3D workspace coordinates have X, Y and Z axes. The extra axis allows the depth of shapes to be represented. The second difference are shape properties. A point has zero dimensions. It is a singularity and has no dimensional properties. A line has one dimension. All one dimensional shapes have length as their only dimensional property. A plane has two dimensions. All two dimensional shapes have length and breadth as their dimensional properties. Two dimensional properties can be represented as many different shapes (triangles, rectangles, pentagons, hexagons, etc.). Three dimensional shapes have length, height and depth as their dimensional properties. In 3D modelling term, length is usually referred to as width which is the term that is used in this research. All three dimensional shapes are solids. They can be represented in many different forms (cubes, cylinders, spheres, hemispheres, etc.). In this thesis, the experiments use 3D workspace where points, lines and planes are considered as primitive elements. All three elements are equally important to form a shape. Points are the most fundamental structure of all. Lines represent relationships between two or more points. Planes are relationships between points and lines. For example, a hexagon shape has six lines and six points. Solids are relationships between planes. For example, a solid cube is a combination of six square planes meeting at right-angles that has eight points and twelve lines.

- **Lexical Level**

This level has a strong connection with the morphological level; they cannot be separated. Although shape elements are created at the morphological level, the lexical level is required to define the relationships between each element to form a vocabulary of shapes. This level aims at creating a basic primitive polygonal shape (solid modelling shape), in other words, a shape root. Fig.5.2 shows an example of two basic vocabularies, namely sphere and cube. As basic foundation building blocks, less complex shapes such as cubes, cylinders and spheres can be manipulated or built up to achieve a high level of complexity of any desired shape. The boundary between each element must be defined at this level, so that they are not considered as elements anymore but as members of a particular shape. Each shape carries a number of their own members which cannot be shared with other shapes. This will be important later in the process when shapes become complex and can generate an infinite number of shapes. At this level, it requires information of a minimum of two vocabularies of shape before any rules can be applied in the following analysis levels.

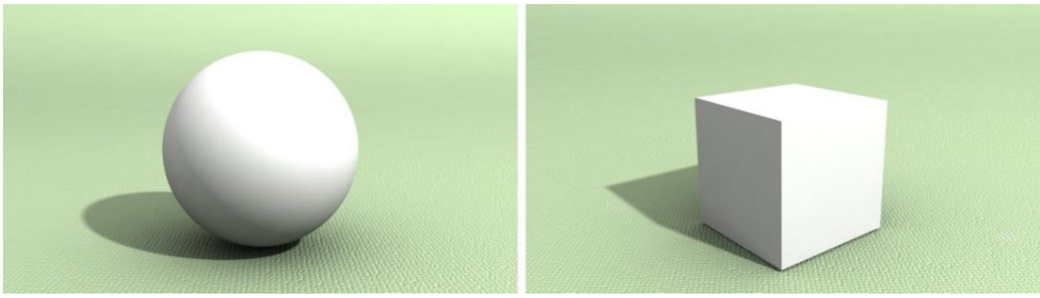


Fig.5.2 *An Example of Vocabulary of Shape A and B in Lexical Level*

- **Syntactic Level**

The syntactical level applies shape rules. At this level, shapes are considered as single units where their members are not isolated. A primitive polygon shape represents a shape root. Modification may be applied through shape rules. Several shapes may be achieved according to their root. For example, a pentagon or an octagon shape can be created by rotating and merging a square cube along one axis. This makes cube a root and pentagon and octagon as its members. Various types of shape can be achieved depending on the structure of shape rules. Most importantly, shape rules govern shapes in a cohesive structure binding them grammatically, in the same way that English grammar rules help create a simple syntactically well-formed active sentence, consisting of a noun phrase (NP) followed by a verb phrase (VP) (Fig.5.3).

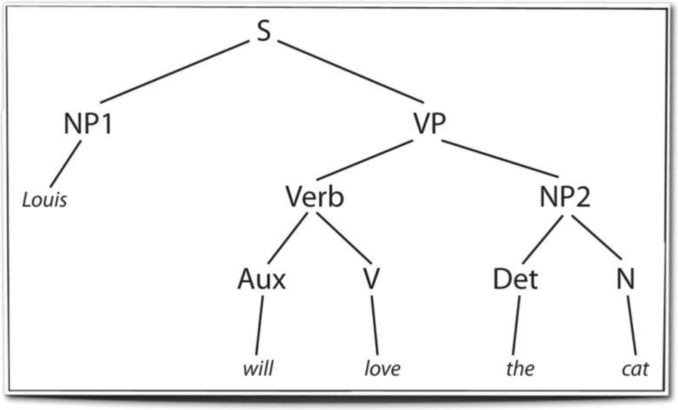


Fig.5.3 *Syntactic Parsing Example*

In this case study, the syntactic level is divided into two parts: a domain grammar and a shape grammar. The domain grammar focuses on the basic components that define a humanoid character consisting of a head, a body and limbs. Shape grammar focuses on the spatial arrangement and transformation of shapes to create the three basic components: head, body and limbs (Fig.5.4). As shape grammar requires the generation of a vocabulary at the lexical level to create the head, the spatial rules and transformational rules are applied first to the head component and using appropriate shape proportion rules, the next two components, body and limbs, are created. The spatial rules define the translation of vocabularies in a 3D workspace. The transformational rules apply object scaling, extruding and merging to modify and create the required basic components. The flexibility of shape transformations and level of modifications can be defined through transformational rules. This level ensures that all basic components are created proportionately (Fig.5.5). However, the contextual environment of the appropriate humanoid anatomy remains unresolved.

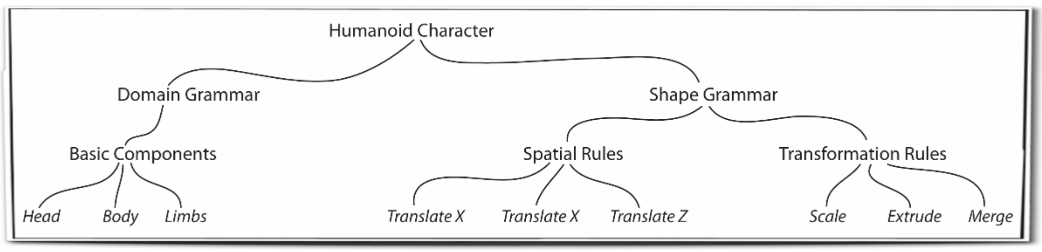


Fig.5.4 *Syntax Level of Harmonised Shape Grammar*

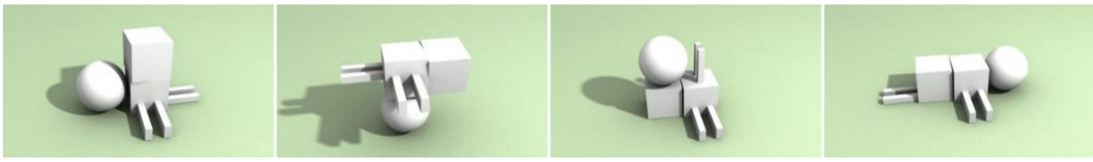


Fig.5.5 *Shape Proportion Rules Applied to Form Humanoid Core Components in Syntactic Level*

- **Semantic Level**

The semantic level brings important contextual considerations to the design process. The previous levels help assemble the primitive shapes to produce the three core humanoid components but fails to produce a meaningful arrangement of the head, body and limbs (Fig.5.5). The way in which these core components are combined must adhere to specific constraints. In order to produce meaningful shapes, two contextual aspects have to be considered: the basic components should be sensitive to the character's habitat and character's body functions which will affect figuration of humanoid anatomy. In order to assemble these three core components, joint components have to be created, in a way sensitive to their environment. The neck is the first component to link the head to the body. Shoulders and hips join the body to its limbs. Wrists and ankles attach hands and feet to limbs. Each core component carries a set of additional features to design: a head with its organs (e.g. eyes, nose, mouth, ears), and upper limbs (arms) and lower limbs (legs). Besides identifying relationships between core components, the design must take into consideration specific spatial arrangement of these components with respect to the habitat of the humanoid as well as its body functions.

The next step is to define the character's habitat, which describes the nature and the arrangement of joint components and core components to suit appropriate environmental conditions. The implementation divides character habitat into three important types: terrestrial, subaquatic and celestial. If necessary, the habitat may override the configuration of the humanoid anatomy. A character that lives in a subaquatic environment requires physiological and anatomical features to cope with aquatic living; for example, fins and tails are necessary for balance and propulsion. A terrestrial character requires a skeletal structure to support the body against gravity and lower limbs to support the trunk when standing. A celestial character requires skeletal components to be strong enough but lightweight to withstand flying. Its anatomy and character habitat affect significantly the body function of a humanoid character. Furthermore, each component carries specific functional properties, for example, eyes function for seeing, nose for breathing, mouth for feeding or communicating, and legs for walking. The design of these types of humanoids must meet criteria compatible with their habitat and body function, and create coherent anatomically meaningful shapes. The resulting anatomical configuration of this semantic level is depicted in Fig.5.6, focusing on legitimacy, consistency and compatibility in the design and arrangement of the body components of humanoids. The first parent for each of the three types of humanoids is generated at the pragmatic level.

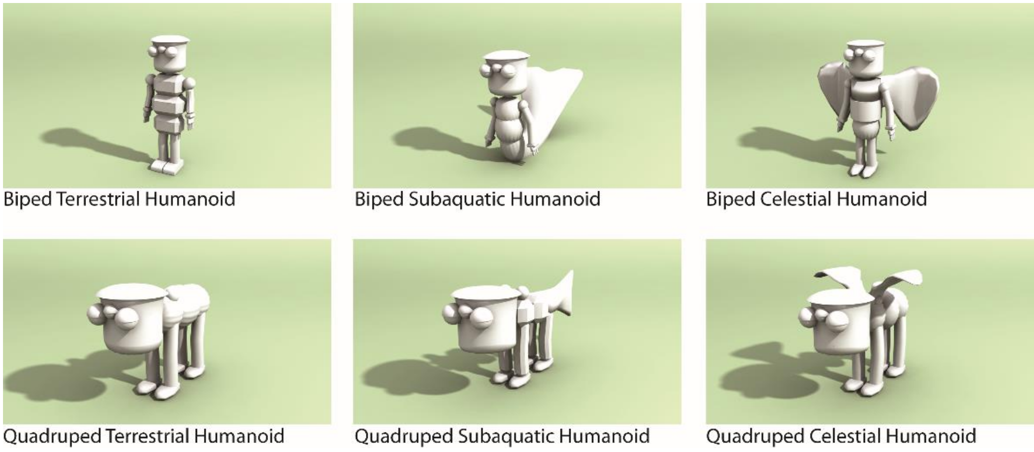


Fig.5.6 *A Set of Humanoid Characters Reflecting Their Habitat and Body Function Requirements*

- **Pragmatic Level**

The pragmatic level aims at achieving harmony among the humanoid members. At this level, a design is not considered at the component level (as an individual's core, joint or specific component) but as a group in the design, in this case, a character. Each generated character must be consistent and compatible with its siblings and parents, in terms of physiognomy, functionality, habitat and personality. Harmony in the humanoid family is achieved by assigning specific attributes, characteristics (physical states), and behaviour (mental states) compatible with their environment. These are described in the following sections.

Character attributes include colour, texture and material. Colour theory is a set of principles used to create harmonious colour combinations. The colour wheel is a visual representation of colour theory, and formulated in terms of primary, secondary and tertiary colours. This thesis divides the colour wheel into twelve slots arranged in clockwise positions. The three primary colours, which include red, yellow, and blue are placed at the first, fifth and ninth slots. The three secondary colours, which include orange, green and violet are placed at the third, seventh and eleventh slots. The last six tertiary colours including red-orange, yellow-orange, yellow-green, blue-green, blue-violet and red-violet are placed at the second, fourth, sixth, eighth, tenth and twelfth slots as shown in Fig.5.7.

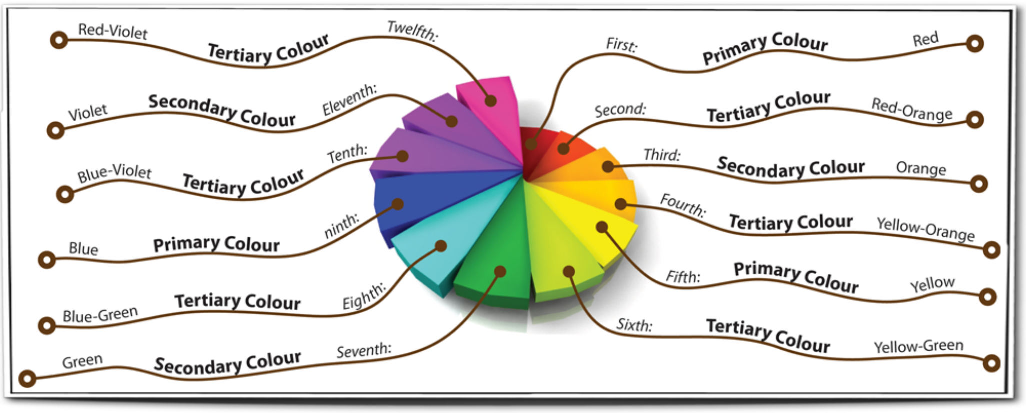


***Fig.5.7** Colour wheel is divided into twelve sections*

The implementation combines both Chevreul's and Rood's theories of colour. Chevreul (1861) explains that the laws of contrast of colour are important in harmony as the contrast between complementary colours produces a pleasing effect on the eye. Rood (1881) states that such an effect may be achieved by creating a contrast of colours in a small space. Both the Chevreul and Rood theories have had a great influence on Neo-impressionists and French impressionists (Westland *et al*, 2007). As result, a design must be constructed with both similar and contrasting colours to fully produce a pleasing and appealing, or in other words, harmonious effect. According to contemporary colour harmony theory, 75% of members of a given group are selected from similar colours, while the other 25% are selected from contrasting colours. Similar colours are colours that sit next to each other in the colour wheel, so- called neighbour colours. For example, red and violet are the neighbour colours of red-violet. Contrasting colours are opposite colours from the opposing colour wheel slots. For example, red is opposite to green, yellow is opposite to violet, and blue is opposite to orange. The application of colour theory to our humanoid characters has led to the following complementing colour arrangement of the humanoid components (Fig.5.8). For example, three sets of colour schemes are applied to biped celestial humanoid characters. The first character has a red scheme, the second character has a green scheme, and the third character has a violet scheme.

Fig.5.8 *A Set of Humanoid Characters Reflecting Their Colour Attributes Requirements*

Texture describes the surface of an object showing how it looks from outside whereas **materials** shows what objects are made of from inside. The implementation groups texture and materials into three types: natural type which includes rocky and wooden physiognomy, unnatural type which includes plastic and metallic physiognomy, and biological type which includes leather, skin, and fur physiognomy. Each physiognomy has its own characteristics with appropriate texture and material. For example, a wooden character (natural type) may be combined with an outfit made from a natural fabric type such as clothes, whereas a metallic robotic character (unnatural type) may be combined with an outfit made from unnatural material such as iron or plastic. To achieve harmony in a given design, the mixture between types should be minimal.

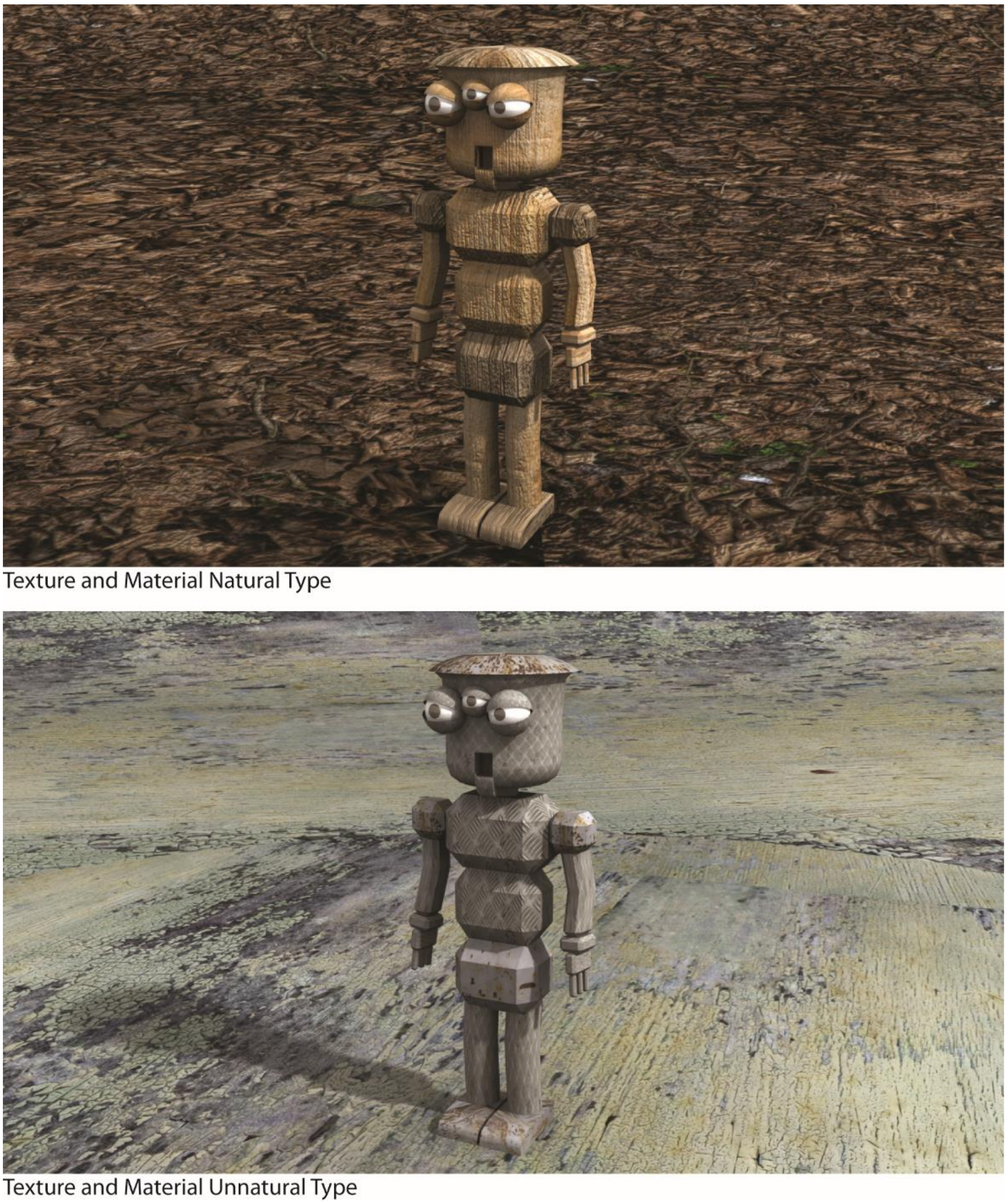


Fig.5.9 *A Pair of Humanoid Characters Reflecting Their Texture and Material Attributes Requirements*

Humanoid characters are implemented using the software, Autodeck MAYA, which divides materials into four types: Lambert, Blinn, Phong and Anisotropic. Each material provides its unique characteristics which may be applied to specific textures. Lambert has the lowest specular property. It has no reflectivity and contains zero specular properties. It is known as a material for matte surfaces, which can be combined with natural and biological texture such as paper, wood or leather. Blinn's specular properties which have 0.3 eccentricities, 0.7 specular roll off and 0.5 reflectivity, is a material compatible with natural and unnatural types such as rock or plastic. Phong has more specular properties than Blinn. Its unique specular properties have a 20 cosine power and 0.5 reflectivity. Phong is suitable in combination with unnatural textures such as metallic or glass. Anisotropic's unique properties are its specular properties controllers: angle, spread (X and Y coordinates), roughness, Fresnel index and reflected colour. However, it does not provide any control over reflectivity. This type of material is suitable with an object which is long and thin in shape and contains curve properties such as animal fur or human hair. To achieve harmony, it is important to apply an appropriate combination of texture and material.

Other physical character characteristics include two properties: size and weight. Size refers to the average body height and body size whereas weight refers to the average weight of characters. To maintain harmony, all characters within a given humanoid type should exhibit a similar size and weight. Fig.5.10 shows the development of humanoids from character basic components to harmonious colour, texture and material characters.

Character behaviour focuses on humanoids' mental states and on their emotional expressions. Three states may be described: normal state, positive state and negative state. A normal state shows minimal emotional expression, whereas a positive state shows a happy and friendly emotional expression and a negative state shows a depressed emotional expression. A harmonious family consists of humanoid members exhibiting a common emotional state. Thus, for example, a happy humanoid character cannot be found among a family of depressed humanoids.

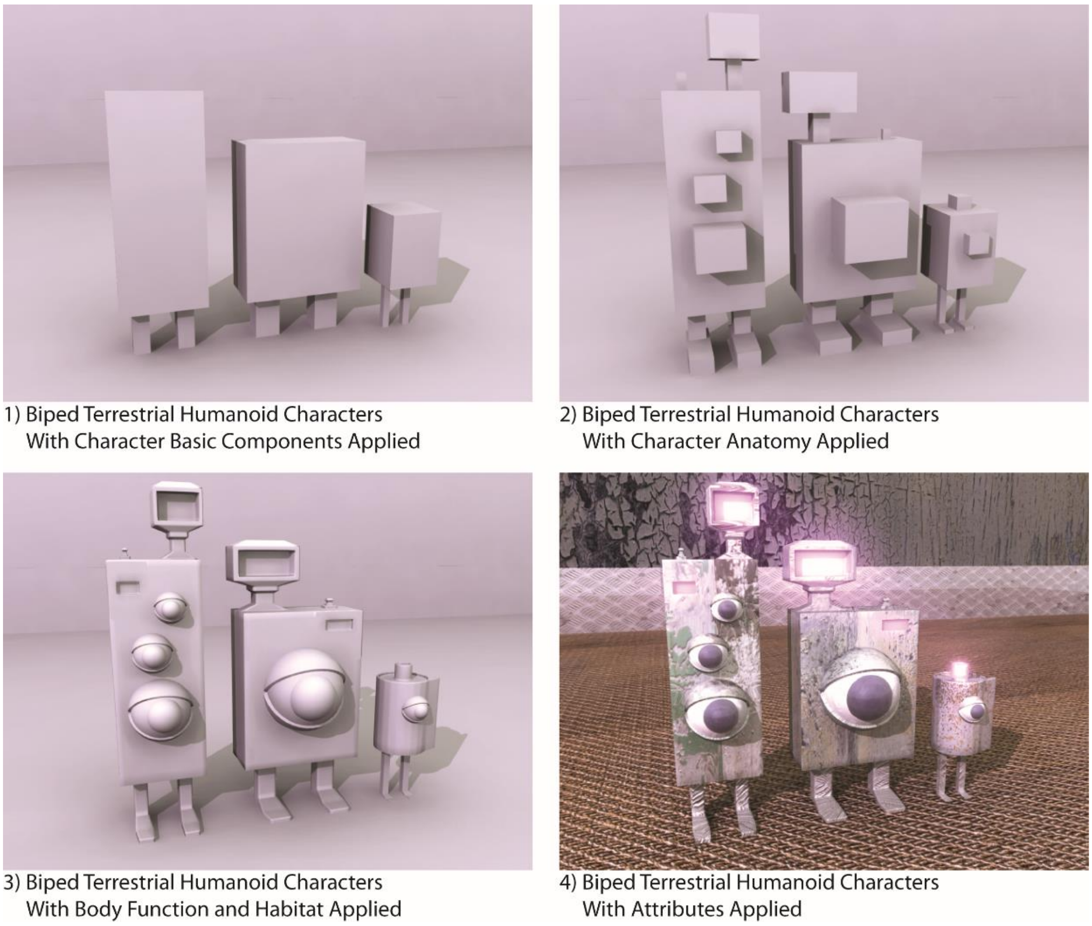


Fig.5.10 *An Example of Humanoid Characters Development*

In summary, Fig.5.11 captures the steps used in generating humanoid characters through the five levels of analysis. The process starts by defining the basic shape elements at the morphological level. A vocabulary of shapes is formed at the lexical level. By applying shape grammar and domain grammar rules, the core components of a humanoid are created at the syntactic level. The semantic level transforms the core components into meaningful forms and arranges them so as to be consistent and compatible to their habitat and body functions. The pragmatic level produces the first generation of humanoid characters by applying principles of character design based on colour, texture, and material harmony, and with appropriate physical and behavioural attributes.

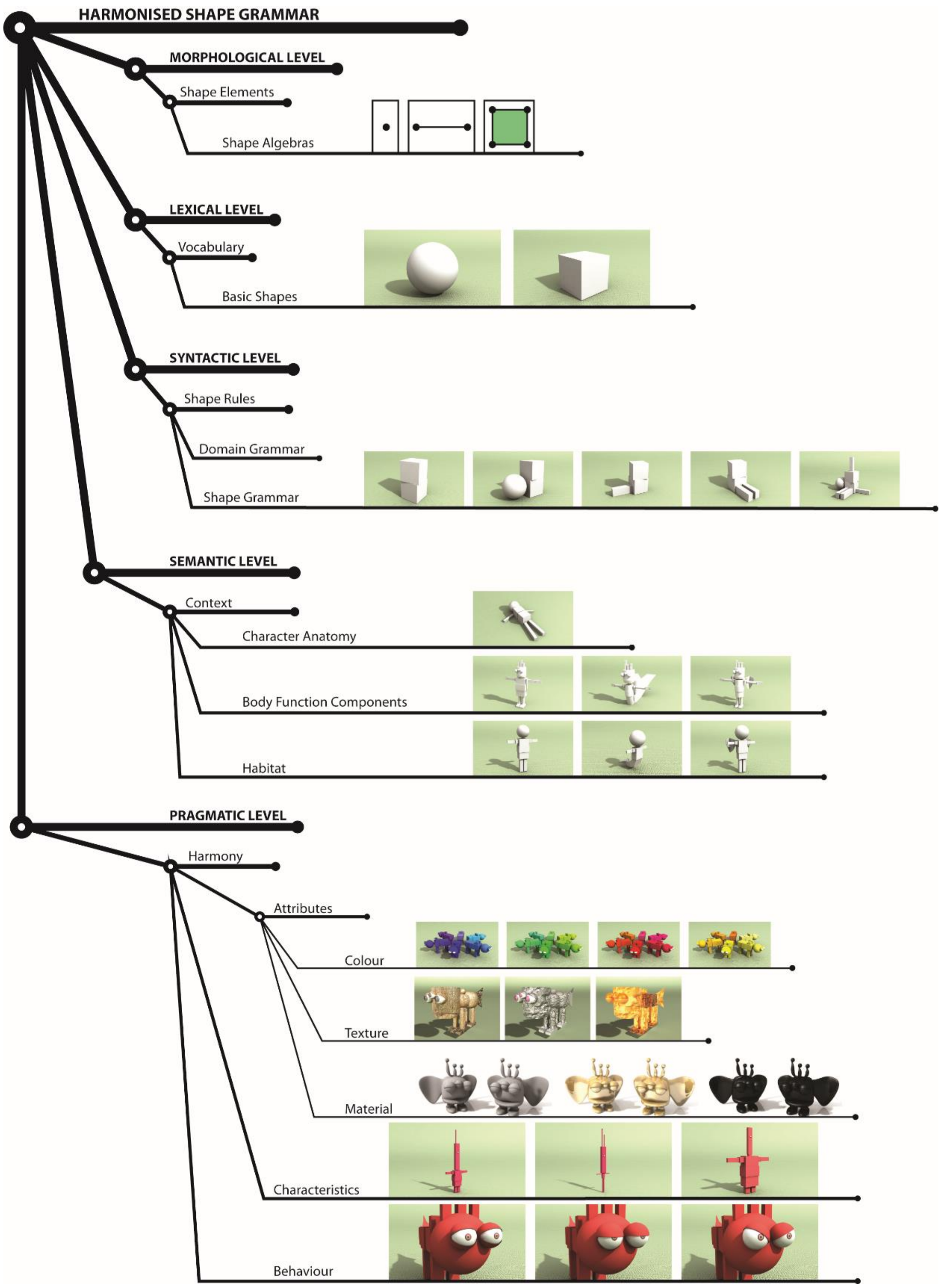


Fig.5.11 *The Implementation Steps of the Five Levels of Analysis*

5.3 The Implementation in MAYA

The generation of the first set of humanoid parents is implemented using Autodesk's 3D graphics software MAYA. The implementation is carried out using a scripting language known as MAYA Embedded Language (MEL). MEL is syntactically similar to Perl and Tcl and provides the dynamic array-allocations and direct access to functions specific to MAYA software. In MAYA, each part of a scene that is used to create an object with history includes 3D geometries, animations, expression relationships, lights, colours, textures, materials, and arguments, which are represented as nodes or attributes (Wilkins and Kazmier, 2005). Fig.5.12 shows the nodes connection hierarchy which generate each character using MEL. All design principle rules are applied using sets of scripting commands in order to generate characters grammatically, contextually and harmoniously.

The implementation is divided into three components. Component A applies the five analysis levels of the harmonised shape grammar. The first level, morphological level, starts with gathering user input information to prepare an appropriate workspace. The input information is used to form a vocabulary of shapes and stored at the lexical level. The user chooses a set of rules to apply at the syntactic level. The shape proportional rules (domain grammar) and transformational rules (shape grammar) are applied to the vocabularies in order to create the head. The head is created along with three data sets, which include head height, width and depth, to be used for proportional rules later on in the generating process. These data sets are references to create other parts of a humanoid character proportionately. The body is the second part to create. By applying proportional rules to head height, width and depth, transformational rules are applied to create body parts. The new data which includes body height, width and depth is created. This data is used to create limbs. Limbs are divided into two parts: arms and legs. Arms and legs are created using the body height, width and depth data. All three parts of a character (head, body and limbs) are achieved by the end of the syntactic level. At the semantic level, the spatial rules are applied to rearrange the three parts following the character anatomy, and transformational rules are applied to transform the character to achieve the character's body function and habitat requirements. Character anatomy has two sets of rules to generate biped and quadruped characters. The biped rules set up the character to be on both legs which result in legs attached to the bottom of its body and arms to each side of the body. The quadruped rules set up the character to be on four legs which require transformation of the arms. The transformational rules modify height, width and depth of the arms to match the legs (and function as legs). The body function transforms body parts into functional body structures. The necessary body components are created to perform tasks functionally. Neck, shoulders, wrists, hip and feet are created to connect the head, body and limbs together contextually. The habitat transforms body parts to match their environment. There are three sets of rules: terrestrial set, subaquatic set and celestial set. The terrestrial set applies transformational rules to modify the character's body parts to suit a terrestrial environment. The subaquatic set applies transformational rules to modify the character’s legs to become fins and tails in the case of biped humanoids. The celestial set applies transformational rules to create wings which are attached to the character's body. The pragmatic level focuses on harmony. There are three sets of rules aimed at achieving harmony based on a selection of appropriate attributes, characteristics and behaviour. The attribute set starts with the selection of primary, tertiary, and complementary colours from the twelve slots colour wheel (Fig5.7) to create a colour scheme set. All colours applied to a character come from this colour scheme. The texture and material are divided into three types: natural, unnatural and biological. Appropriate texture and material are applied to each character design. This requires seven nodes to create a texture by manipulating a set of nodes such as material node, colour node, mapping node, texture mapping coordinate node, bump mapping node, bump value node and bump mapping coordinate node (Fig.5.13). The characteristics set are applied to control the overall physical state of each character design by setting a range of appropriate size and weight values. The behaviour set is applied to create the character's mental state, as shown by its facial expression. There are three types of expressions included in this implementation: depressive, neutral and friendly.

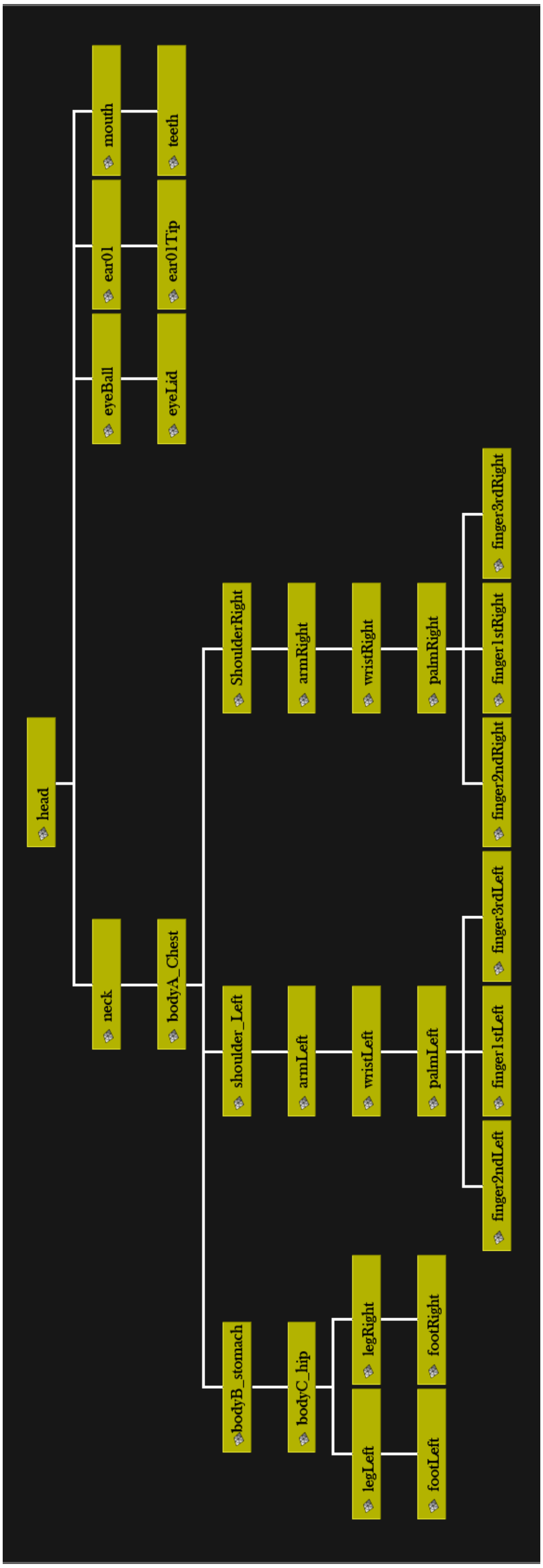


Fig.5.12 *An Example of Nodes Connection Hierarchy Represented a Character*

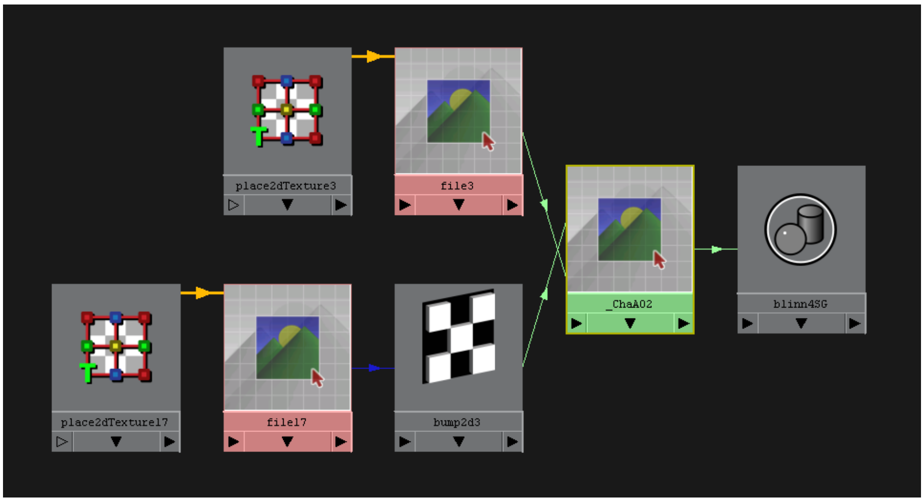


Fig.5.13 *An Example of Nodes Connection Hierarchy Represented a Texture*

The behaviour set applies the same type of expression to all characters in a given set. The final output of component A produces the parents (i.e. the Adam and Eve parents for each type of humanoid character) as a 3D geometrical character (solid modelling). To produce the next generation for each type of humanoid character this research project investigates the use of a genetic algorithm depicted by component B in Fig. 5.13; this is described in the next section. Component C converts the humanoid characteristics data obtained from component B into MAYA parameters in order to produce the final design of a set of harmonious humanoid families. A summary of the steps involved in these components is given in Fig.5.14.

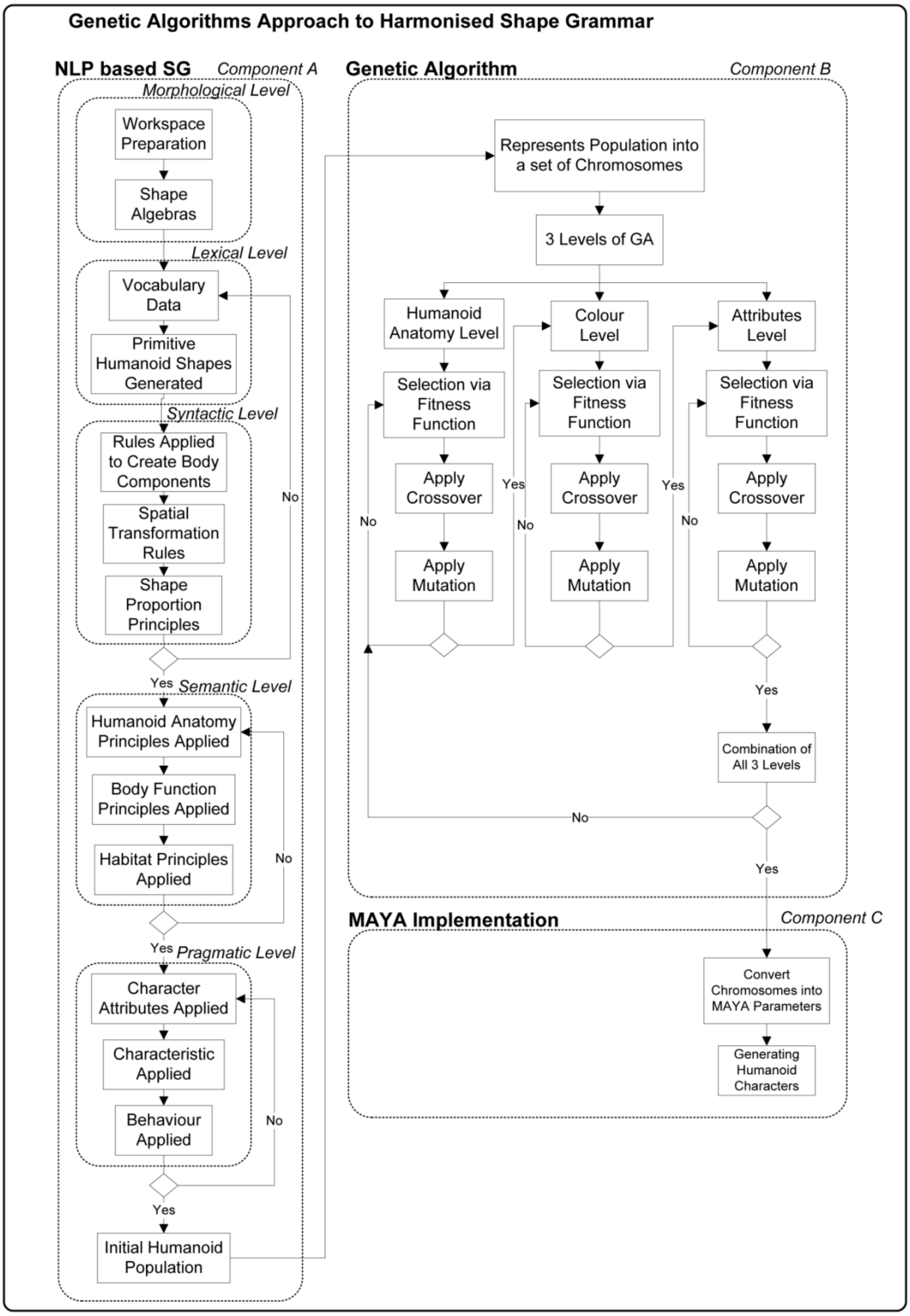


Fig.5.14 *Genetic Algorithms Approach to Shape Grammar*

5.4 Generation of Harmonious Humanoid Family Members

5.4.1 Introduction to Genetic Algorithms

The previous section explains how the first generation for each of the three types of humanoid characters are generated. However, when designing hundreds of character designs, one faces a great challenge in maintaining consistency among these characters. One often spends long and tedious hours of revisions and iterations, eventually achieving a satisfactory result. To aid in this process this research advocates the use of genetic algorithms as these may not only achieve consistency but also diversity yet maintaining harmony among the humanoid family members.

Genetic algorithms were introduced by Holland in the 1970s (Tsoukalas and Uhring, 1997) and belong to a class of evolutionary algorithms aimed at producing well-defined efficient solutions. They simulate the survival of the fittest among individuals over consecutive generate ons in problem solving. Each individual represents a point in a search space and a possible solution. Each generation consists of a population of character strings, analogous to the genetic structure and behaviour of chromosomes in our DNA. A chromosome consists of genes which encode a trait (e.g. eyes colour, nose shape). By combining information from the chromosomes genetic algorithms apply selective `breeding' of the solutions to produce strong offspring from the parents.

Genetic algorithms are applied for solving optimisation problems where randomness is involved (Kumar *et al*., 2010) and have been used in numerous domains. A few recent examples are given below for illustration purposes. A survey of applications of genetic algorithms is available from Kumar *et al*., 2010.

In robotics, Yasuda and Takai (2001) used a genetic algorithm to create a sensor-based path planning and intelligent steering control of non-holonomic mobile robots. The aim was to use a genetic algorithm to plan robots paths in an unstructured environment in real-time. In image processing, Minglun and Yee-Hong (2001) applied genetic algorithms to create a multi-resolution stereo matching by using an intensity-based approach. This application was able to increase the accuracy of the disparity map by removing the mismatches from occlusions and false targets. In gaming, Sandstrom and Norstrom, (2002) used genetic algorithms to manage complex temporal requirements in a real-time control system for a game. This game program automatically selected monsters that fared the best against the player. In engineering, as a real-time system, Madureira *et al*., (2002) created a coordination mechanism for a real world scheduling problems. This real-time system applied genetic algorithms to assign task priorities and offsets to guarantee that real time timing constraints. In social science, as a decision making system, George *et al*. (2012) used a genetic algorithm to create an airline booking terminal open and close decision system. This application aimed at maximising the revenue of an airline, optimising flight bookings, and deciding the transportation terminal open and close system. The application observed data of a particular booking terminal's historical booking to optimise solutions. In computer science, Sharma *et al*. (2013) used a genetic algorithm for software testing. The aim was to improve the software testing issues such as effective generation and prioritisation of test cases. Girgis *et al*. (2013) applied a genetic algorithm to solve the routing and capacity assignment problem in a computer network. The aim was to achieve a considerable memory saving and a better utilisation of the network. The authors suggested that by combining genetic algorithms and simulated annealing the issues can be solved. In crime investigation, Frowd *et al*. (2004) used a genetic algorithm to reconstruct the perpetrator's likeness to that of criminal faces for an eye-witness. This application could generate a full composited face. As matter of fact, faces were not remembered as separate components. This approach could help eye-witnesses to get a better idea of what faces of criminals looked like.

5.4.2 Genetic Algorithms Operators

A genetic algorithm is one of a class of algorithms that is based on population genetics used in computing to find optimal solutions by searching through a *solution space*. A solution space is represented in term of a collection of candidate solutions called a *population* (Thede, 2004). A single solution is referred to as an *individual*. Each individual is defined by its chromosomes as a string of genes (encoded in, for example, binary). The search for an optimal solution is based on the concept of evolution. The population evolves over generations so new individual are "born" and "die". The algorithm measures the *fitness value* of each individual and selects the fittest individual for “breeding”, by a process of crossover or recombination. Crossover is the process of interchanging chromosomes from the parents to create an offspring using either as a one-point crossover (Fig.5.15A) or n-point crossover (Fig.5.15B). A higher life-chance quality of an individual has a better chance to be selected and evolved. The evolution of the population is carried out by this *crossover* process. A *mutation* may also be applied randomly to alter the characteristics of an individual. The value of the mutation should be considered carefully: too small a value could result in a negligible development in the solution quality, whereas too large a value could change outcomes of the entire population dramatically. The mutation process involves a random selection of 10 percent of the population and a random selection of chromosomes from the two parents (Fig.5.15C). At each epoch, fitness evaluation, crossover, mutation and selection are performed.

In summary, the process starts with a population of individuals and the algorithm is repeated until specific conditions are satisfied. A summary of the algorithm steps is given in Fig.5.16A and Fig.5.16B.

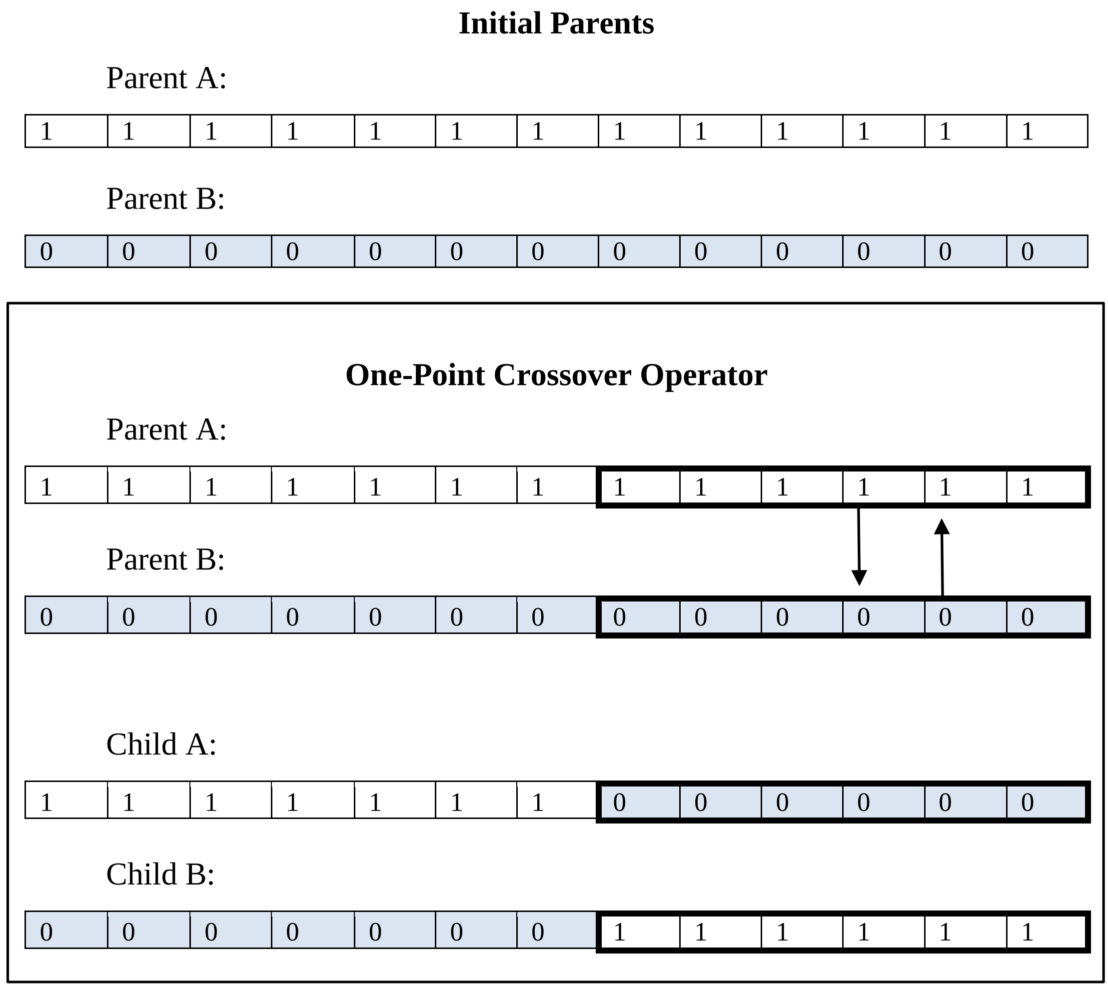


Fig.5.15A *An Example of One-Point Crossover*

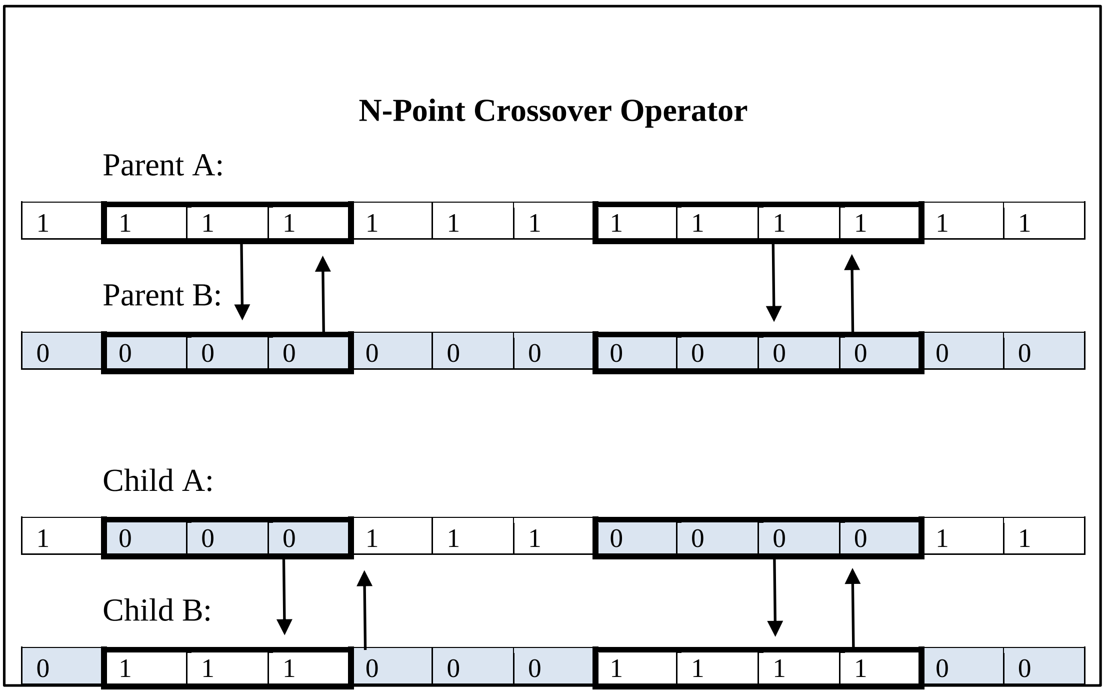


__Fig.5.15B__ An Example of N-Point Crossover

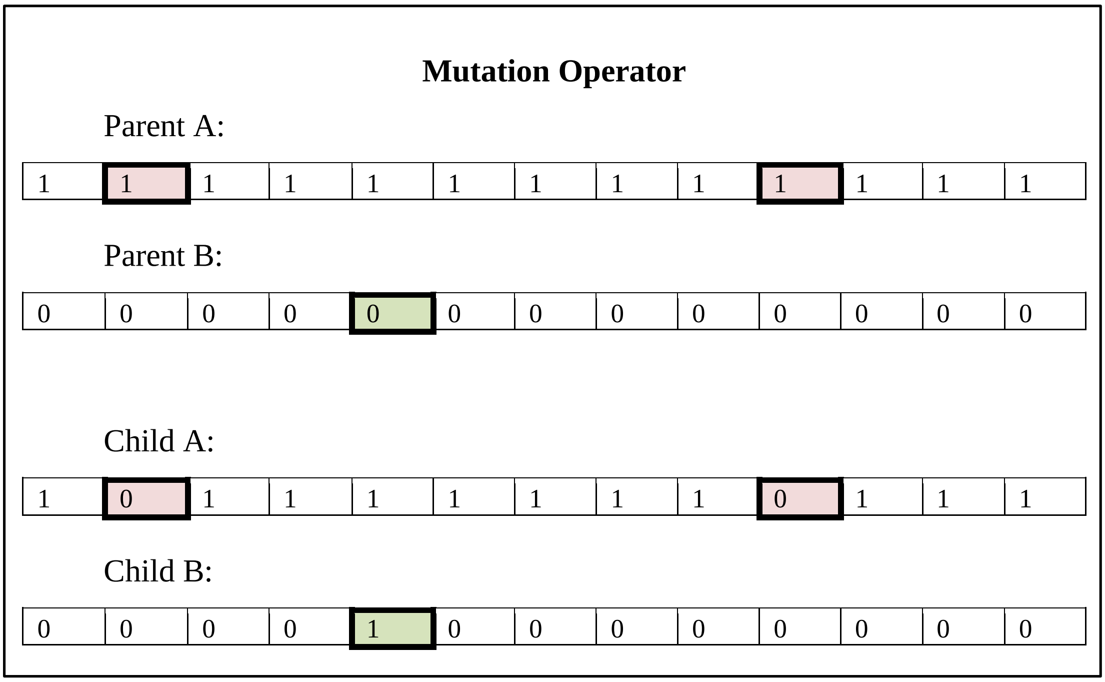


__Fig.5.15C__ An Example of Mutation

1. Create a **population** of random candidate solutions called *population-A*.
2. Until the algorithm termination conditions are met, do the following (each iteration is called a generation):
    a. Create an empty population called *population-B*.
    b. While *population-B* is not full, do the following:
        i. **Select** two **individuals** at random from *population-A* so that individuals which are more **fit** are more likely to be selected.
        ii. **Crossover** the two individuals to produce two new individuals.
    c. Let each individual in *population-B* have a random chance to **mutate**.
    d. Replace *population-A* with *population-B*.
3. Select the individual from *population-B* with the highest **fitness** as the solution to the problem.

Fig.5.16A *Genetic Algorithm*

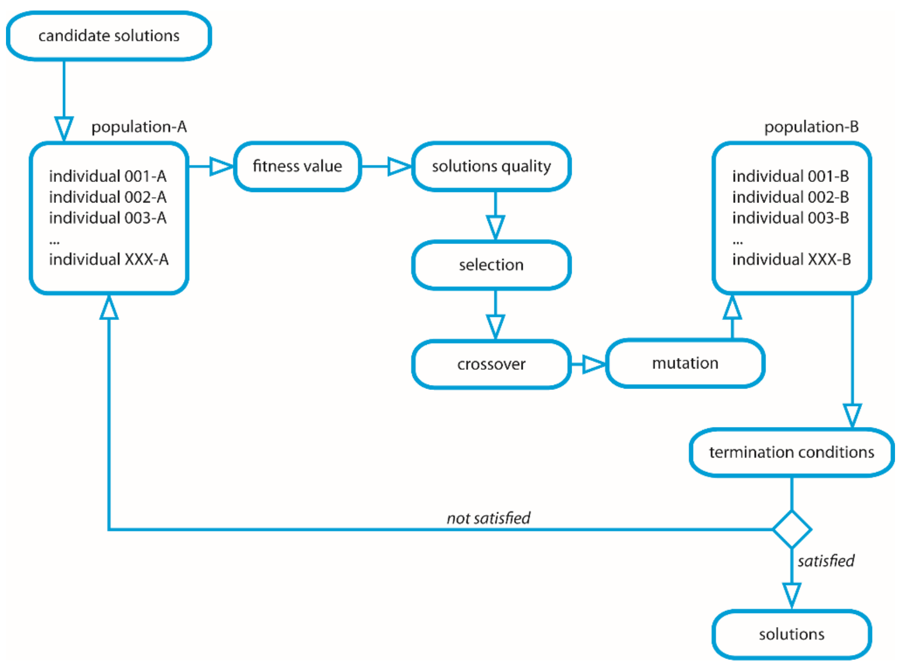


Fig.5.16B *Genetic Algorithms Flowchart Applied to the Generation of Humanoid Families*

5.4.3 Implementation of GA to Generate Humanoid Family Members

In this section, we describe the production of the generations of humanoids by adapting the principles associated with genetic algorithms to the design of the next set of harmonious children. The aim is to start with the first parents (“the Adam and Eve parents”) generated via the harmonised shape grammar and apply the concepts of genetic algorithms to create a set of harmonious yet diverse population. Each individual parent is represented in terms of chromosomes with 28 genes capturing three specific characteristics: anatomy (13 genes), colour (8 genes), and other attributes (7 genes). A fitness function is applied to each set of chromosomes. The search for a harmonious set of off-springs is based on the algorithm described in Fig.5.16A and Fig.5.16B to select individuals which are identified by a set of harmonious principles.

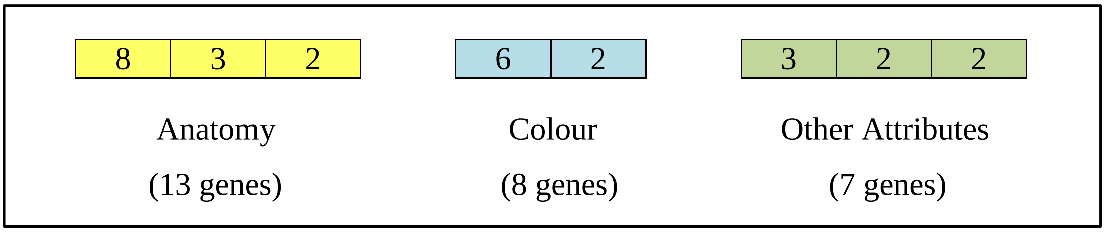


Fig.5.17 *Humanoid Chromosomes*

5.4.3.1 GA Encoding

The humanoid anatomy is encoded as a vector consisting of three sections: body shapes, head components, and body functions represented as thirteen genes. Eight genes are used to define the humanoid body shapes (e.g. head, neck, top torso (chest), middle torso (stomach), bottom torso (hip), shoulders, upper limbs (arms), lower limbs (legs)), three genes to define the head components (e.g. eyes, ears, teeth), and two further genes to define the appropriate body function components: one gene defining the habitat and one other defining the posture. Table 5.1 describes how each gene is represented. For example, the genes representing the body shape, head, and body function components may be either a cube shape, denoted by 1, or a cylinder shape, denoted by 2, or a sphere shape, denoted by 3. The habitat gene is denoted by 1, 2, and 3 describing the terrestrial, subaquatic, and celestial categories, respectively. Finally, the posture gene is denoted by 1 to represent biped and 2 to represent quadruped.

Table 5.1: *Anatomy Chromosome*

Anatomy Chromosome					
Section: 1st		Body Shape Components			
Gene	Code		1	2	3
A1	HD	Head	Cube	Cylinder	Sphere
A2	NK	Neck	Cube	Cylinder	Sphere
A3	TT	Top torso	Cube	Cylinder	Sphere
A4	MT	Middle torso	Cube	Cylinder	Sphere
A5	BT	Bottom torso	Cube	Cylinder	Sphere
A6	SH	Shoulders	Cube	Cylinder	Sphere
A7	UL	Upper limbs	Cube	Cylinder	Sphere
A8	LL	Lower limbs	Cube	Cylinder	Sphere
Section: 2nd		Head Components			
Gene	Code		1	2	3
A9	EY	Eye(s)	Cube	Cylinder	Sphere
A10	ER	Ear(s)	Cube	Cylinder	Sphere
A11	TH	Teeth	Cube	Cylinder	Sphere
Section: 3rd		Body Function Components			
Gene	Code		1	2	3
A12	HB	Habitat	Terrestrial	Subaquatic	Celestial
A13	PO	Posture	Bipedal	Quadrupedal	N/A

The colour of the human anatomy components is encoded as a vector consisting of two sections: neighbour colour and contrast colour represented by of eight genes. Table 5.2 describes how each gene is represented. For example, the genes representing neighbour colour and contrast colour are either red, denoted by 1, or red-orange, denoted by 2, or orange, denoted by 3, or yellow-orange, denoted by 4, or yellow, denoted by 5, or yellow-green, denoted by 6, or green, denoted by 7, or blue-green, denoted by 8, or blue, denoted by 9, or blue-violet, denoted by 10, or violet, denoted by 11, or red-violet, denoted by 12.

Table 5.2: *Colour Chromosome*

Colour Chromosome														
Section: 1st		Neighbour Colour												
Gene	Code		1	2	3	4	5	6	7	8	9	10	11	12
B1	CHD	Head colour	R	R-O	O	Y-O	Y	Y-G	G	B-G	B	B-V	V	R-V
B2	CNK	Neck colour	R	R-O	O	Y-O	Y	Y-G	G	B-G	B	B-V	V	R-V
B3	CTT	Top torso colour	R	R-O	O	Y-O	Y	Y-G	G	B-G	B	B-V	V	R-V
B4	CMT	Middle torso colour	R	R-O	O	Y-O	Y	Y-G	G	B-G	B	B-V	V	R-V
B5	CBT	Bottom torso colour	R	R-O	O	Y-O	Y	Y-G	G	B-G	B	B-V	V	R-V
B6	CSH	Shoulders colour	R	R-O	O	Y-O	Y	Y-G	G	B-G	B	B-V	V	R-V
Section: 2nd		Contract Colour												
Gene	Code		1	2	3	4	5	6	7	8	9	10	11	12
B7	CUL	Upper limbs colour	R	R-O	O	Y-O	Y	Y-G	G	B-G	B	B-V	V	R-V
B8	CLL	Lower limbs colour	R	R-O	O	Y-O	Y	Y-G	G	B-G	B	B-V	V	R-V

The other attributes are encoded as a vector consisting of three sections represented by seven genes. Three genes are used to define the character behaviour and characteristics (e.g. behaviour, character size, character weight), two genes to define the texture (e.g. primary, secondary), and two further genes to describe material (e.g. primary, secondary). Table 5.3 describes how the genes in the other attributes are represented. For example, the gene representing behaviour is either depressive, denoted by 1, or neutral, denoted by 2, or friendly, denoted by 3. The gene representing size is either short, denoted by 1, or medium, denoted by 2, or tall, denoted by 3. The gene representing weight is either thin, denoted by 1, or normal, denoted by 2, or overweight, denoted by 3. The primary texture and secondary texture genes are either rock texture, denoted by 1, or wood texture, denoted by 2, or plastic texture, denoted by 3, or metallic texture, denoted by 4, or leather texture, denoted by 5, or fur texture, denoted by 6. Finally the primary and secondary material genes are either Lambert material, denoted by 1, or Blinn material, denoted by, 2, or Phong material, denoted by 3, or Anisotropic material, denoted by 4.

Table 5.3: *Other Attributes Chromosome*

Other Attributes Chromosome								
Section: 1st		Behaviour and Characteristics						
Gene	Code		1	2	3	4	5	6
C1	BE	Behaviour	Depressive	Neutral	Friendly	N/A	N/A	N/A
C2	CHP	Size	Short	Medium	Tall	N/A	N/A	N/A
C3	CHS	Weight	Skinny	Normal	Overweight	N/A	N/A	N/A
Section: 2nd		Texture						
Gene	Code		1	2	3	4	5	6
C4	TXP	Primary texture	Rock	Wood	Plastic	Metallic	Leather	Fur
C5	TXS	Secondary texture	Rock	Wood	Plastic	Metallic	Leather	Fur
Section: 3rd		Material						
Gene	Code		1	2	3	4	5	6
C6	MAP	Primary material	Lambert	Blinn	Phong	Anisotropic	N/A	N/A
C7	MAS	Secondary material	Lambert	Blinn	Phong	Anisotropic	N/A	N/A

5.4.3.2 GA Operators

The application of crossover and mutation of the parents' chromosomes may change the structure of the chromosomes and may impact on the harmonious arrangement of the chromosomes associated with anatomy structure, colour scheme, and other attributes. Only individuals who score high fitness function values following crossover or mutation are selected to become the parents for the next generations.

In this project, the algorithm applies one-point crossover randomly. An individual gene is randomly chosen to undergo the mutation, which modifies its value by +/- 1. Both of these operators are applied to every section within the anatomy, colour, and other attributes and illustrated in Fig.5.18, Fig.5.19, Fig.5.20, Fig.5.21, Fig.5.22, and Fig.5.23.

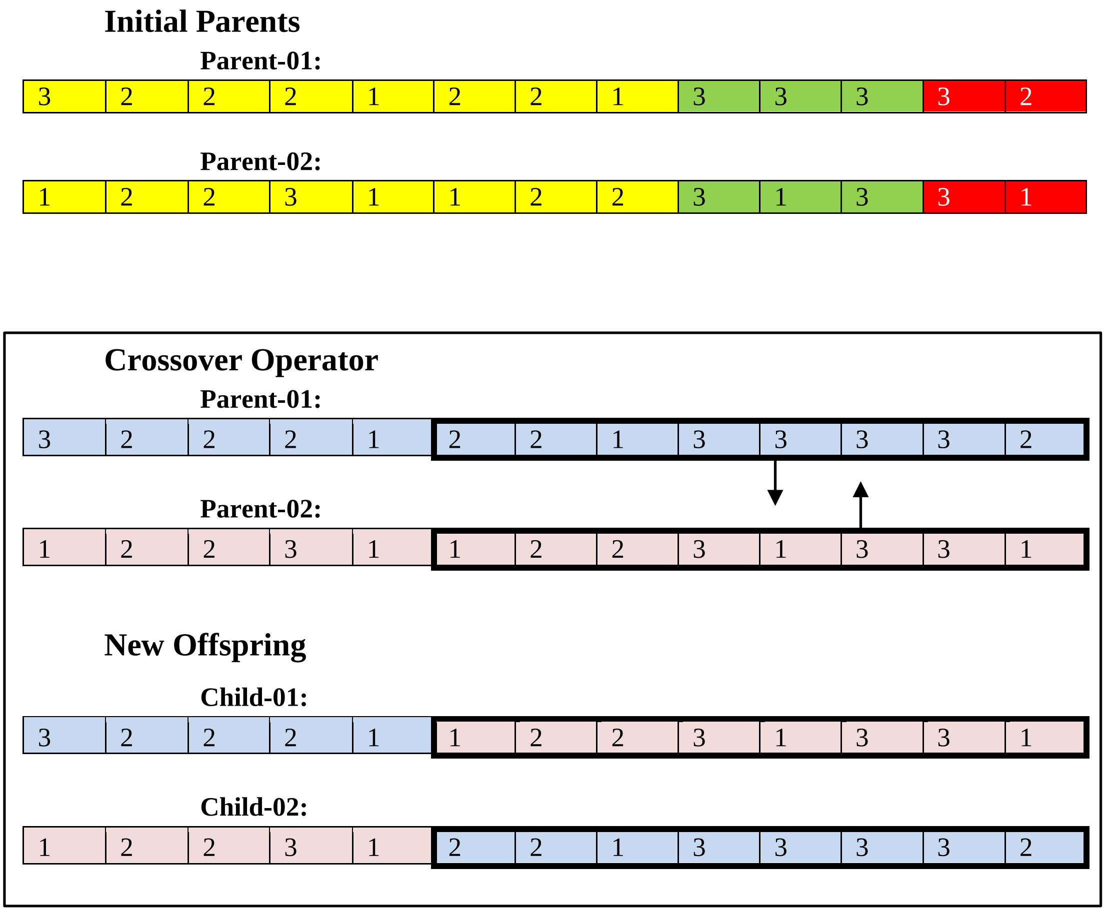


Fig.5.18 *Crossover Example applied to Anatomy Chromosome*

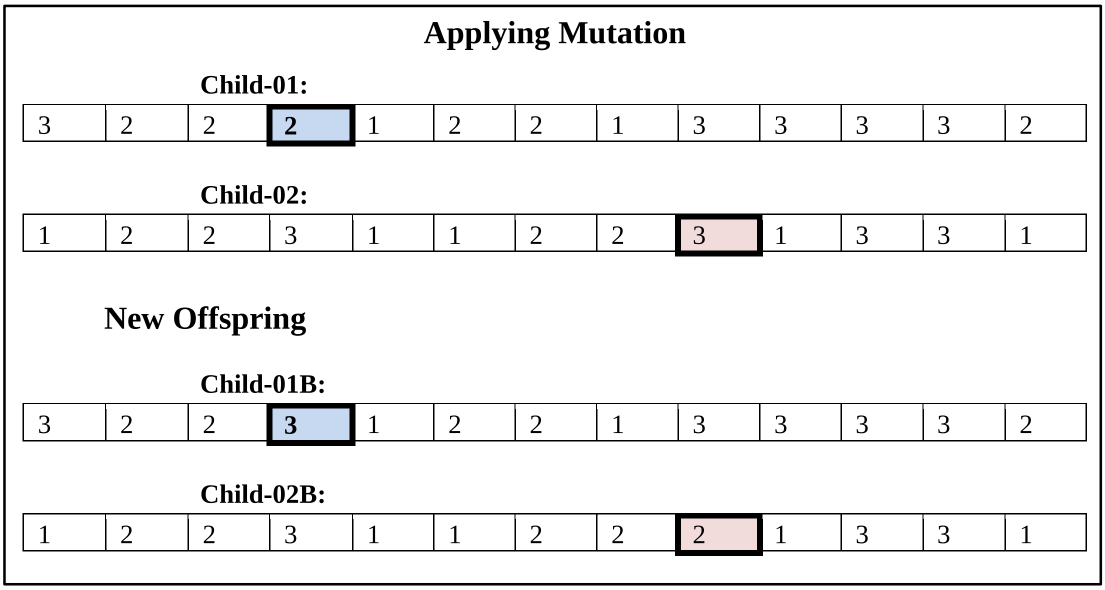


Fig.5.19 *Mutation Example applied to Anatomy Chromosome*

Initial Parents

Parent-01:

5	4	11	5	3	5	7	5

Parent-02:

10	10	10	4	2	11	8	4

Crossover Operator

Parent-01:

5	4	11	5	3	5	7	5

Parent-02:

10	10	10	4	2	11	8	4

New Offspring

Child-01:

5	4	11	5	2	11	8	4

Child-02:

10	10	10	4	3	5	7	5

Fig.5.20 *Crossover Example applied to Colour Chromosome*

Applying Mutation

Child-01:

5	4	11	5	2	**11**	8	4

Child-02:

10	**10**	10	4	3	5	7	5

New Offspring

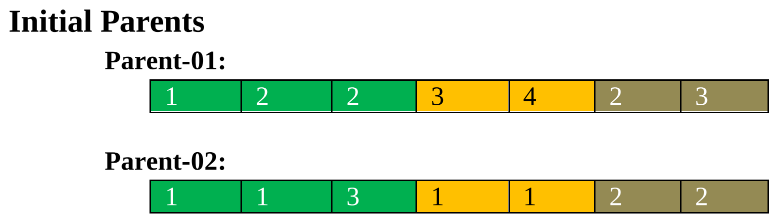
Child-01B:

5	4	11	5	2	**12**	8	4

Child-02B:

10	**9**	10	4	3	5	7	5

Fig.5.21 *Mutation Example applied to Colour Chromosome*



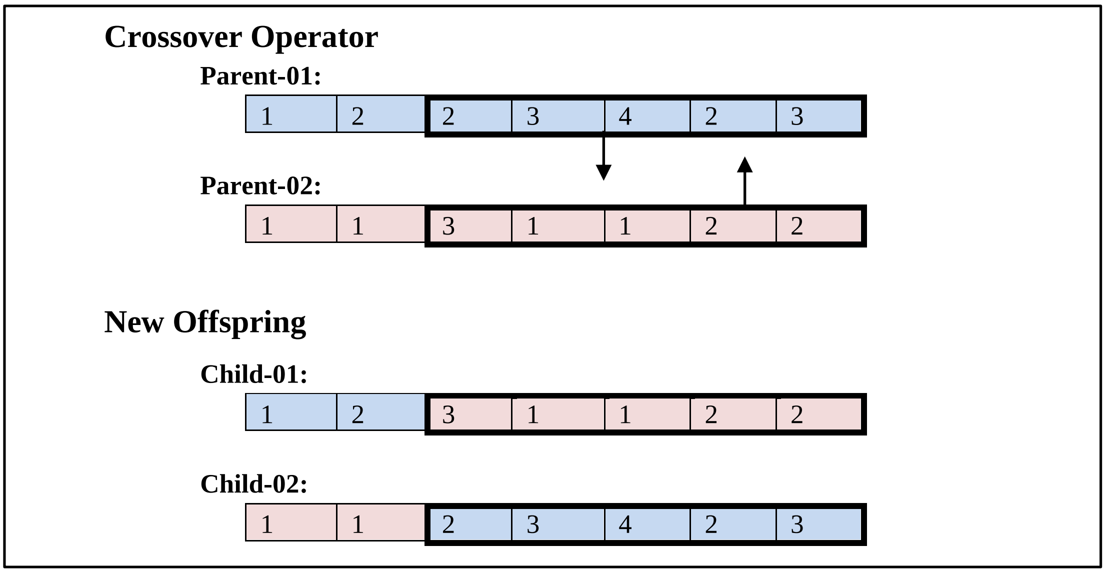


Fig.5.22 *Crossover Example applied to Set of Attributes Chromosome*

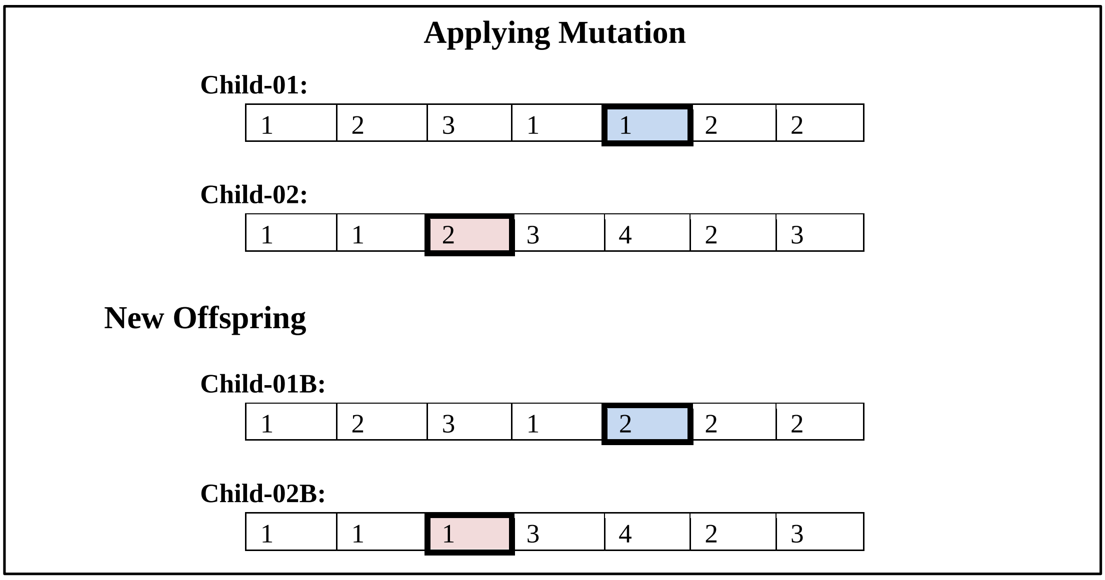


Fig.5.23 *Mutation Example applied to Set of Attributes Chromosome*

5.4.3.3 Fitness Functions

For each section of the chromosomes, a fitness function is applied based on the principles of the design. A fitness value is calculated and a score given to the genes. Those chromosomes that score high values are selected for reproduction.

5.4.3.3.1 Fitness Function for Anatomy Chromosome

The fitness function measures the harmony structure of the new anatomy chromosome by focusing on principles and elements of design which include unity, variety, emphasis (focal point), scale and proportion, balance and rhythm, aimed at creating an effective communication (Brainard, 1998). Unity may be achieved by creating a repetition which is based on grouping by similarity. Elements which are similar visually are perceived to be related (Arntson, 1998). In the humanoid population, chromosomes which exhibit similar shapes should score high fitness values whereas chromosomes with different shapes should score low values. However, selecting individuals with similar shapes could be dull and unappealing, so emphasis on some characteristics is applied to bring some contrast into a design. An element in contrast with another element is more easily seen and understood (Faimon and Weigand, 2004). However, when the contrast is too strong, it can provoke a chaotic feeling and disturb the unity principle. To maintain harmony, the percentage between unity and emphasis has to be precisely calculated. In the implementation the ratio of unity and emphasis is set at 7 to 1. A fitness function is applied to each section of the humanoid anatomy and the sum of these scores is used to rank the individuals' fitness with the following weights: 30% for body shapes, 20% for head components, and 50% for body function components (Fig.5.24) in the following way:

- ***The fitness function for the body shape section.*** The shape section uses unity and emphasis to score the fitness value. Shapes are divided into two types: rectilinear and curvilinear shapes. Rectilinear shapes are created from straight edges and angular corners where curvilinear shapes are created from curves and round forms (Lauer and Pentak, 1995). To apply the unity principle, an individual whose body shape chromosome consists of similar shape types (e.g. a combination of cube shapes and cuboid shapes) scores a higher fitness value than an individual with a mixture of shapes (e.g. a combination between cube shapes and sphere shapes). However to maintain emphasis, the fitness function allows minor differences of shape types (e.g. cuboid shapes which are rectilinear are allowed to be combined with a few cylinder shapes (a minor difference) but are not allowed to be combined with curvilinear shapes such as spheres).

- ***The fitness function for the head components section.*** Balance, which is the distribution of visual weight of design elements, is applied to the head components (Lauer and Pentak, 1995). When design elements are imbalanced, this creates a disturbing and uncomfortable feeling (Faimon and Weigand, 2004). The visual weight balance is calculated by the quantity (Stewart, 2002), position (Arntson, 1998), size (Arntson, 1998), orientation (Stewart, 2002), and shape (Arntson, 1998) of head components.

- ***The fitness function for the body function.*** The habitat and posture of an individual are defined by genes describing the specific habitat of the humanoid (e.g. a subaquatic humanoid requires fins and tail, a terrestrial requires arms and legs, a celestial requires wings), and one chromosome defining the posture. The unity is also applied in this section to maintain harmony, so an individual that has limbs and posture appropriate to its habitat scores a high fitness value.

	Body Shapes	**Head Components**	**Body Function Components**	Total
Genes	○○○○○○○○	○○○	○○	
Scoring	8	3	2	13
Weighting	30%	20%	50%	100%
Criteria	Unity, Emphasis	Balance	Unity	

Fig.5.24 *An Example of Anatomy Chromosome Scoring System*

5.4.3.3.2 Fitness Function for Colour Chromosome

The fitness function focuses on colour theories which follow the principles and elements of design. Colours are determined by the wavelength of light (Bevlin, 1994). There are two types of colours: additive colours and subtractive colours. Addictive colour is colour created from an emitted light source (Bevlin, 1994). The additive primary colours are red, green and blue. Subtractive colour is colour created from light reflecting off a pigmented surface (Stewart, 2002). The subtractive primary colours are red, yellow and blue. The implementation is based on subtractive colours. A twelve-slot colour wheel system is used in order to identify colour hues (Fig.5.7). The first, fifth, and ninth slots are for three primary colours: red, yellow and blue. A secondary colour is a combination of two primary colours together. Orange (a combination of red and yellow) is placed in the third slot. Green (a combination of yellow and blue) is placed in the seventh slot. Violet (a combination is blue and red) is placed in the seventh slot. A tertiary colour is a combination of primary and secondary colours. Red-orange is placed in the second slot. Yellow-orange is placed in the fourth slot. Yellow-green is placed in the sixth slot. Blue-green is placed in the eighth slot. Blue-violet is placed in the tenth slot. Red-violet is placed in the twelfth slot. These twelve colours from the twelve slots colour wheel are applied by the fitness function to each section of the colour chromosome to generate a score. The sum of these scores is used to rank the individual's fitness using the following weights: 60% for neighbour colours and 40% for contrast colours (Fig.5.25) in the following way:

- ***The neighbouring colour section*** uses unity to score the fitness value. To create unity, the colours of the genes in this section have to be the same as each neighbour's colour. The genes are given a maximum score at the start of the GA. A chromosome which has a full-neighbour combination (e.g. green, yellow-green, and blue-green) is given a zero penalty point. A penalty value is computed depending on the number of non-neighbouring colours identified in the chromosome.

- ***The contrasting colour section*** uses emphasis to score the fitness value. To achieve emphasis, the colours of the genes in this section have to be in a contrasting colour with respect to its neighbouring colour section.

Contrasting colours are colours that sit opposite to each other on the colour wheel. For example, red (first slot) sits opposite to green (seventh slot), so they are contrasting colours. The combination between neighbouring and contrasting colours should make the design more appealing and interesting to the eye. However, if the weighting of the contrasting colour is too high, it may give a chaotic and unattractive appearance. The fitness function sets the ratio between neighbouring and contrasting colours at 6 to 2 to maintain colour harmony.

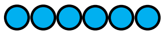
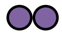
	Neighbouring Colours	**Contrasting Colours**	Total
Genes			
Scoring	6	2	8
Weighting	60%	40%	100%
Criteria	Unity	Emphasis	

Fig.5.25 *An Example of Colour Chromosome Scoring System*

5.4.3.3.3 Fitness Function for Other Attributes Chromosome

The following principles and elements of design below are also applied to genes for behaviour and characteristic, texture and material, resulting in the following weights: 60% for behaviour and characteristic, 20% for texture and 20% for material (Fig.5.26).

- The fitness value is also based on the principle of unity to select the best behaviour and characteristic genes of a chromosome. A score is given to each gene and the sum of these scores is used to rank the fitness of other attributes. ***Behaviour is divided into three types:*** neutral, friendly, and depressed**.** ***Characteristics are associated with size and weight.*** Size refers to short, medium or tall, and weight to thin, normal or overweight appearance. To create unity, an individual which exhibits behaviour of the same type as its parents scores a high fitness value, whereas dissimilar behaviour scores a low value and opposite behaviour scores zero.

- ***There are two types of textures:*** actual and visual texture. Texture in this implementation is limited to visual texture and represents the surface quality of an object (Lauer and Pentak, 1995). This may be created with many marks or shapes (Stewart, 2002) or by reproducing the value and colour patterns of actual textures (Lauer and Pentak, 1995). The fitness function for texture is based on the unity, balance and rhythm principles. Repetition of similar textures may create unity. An element with a more complex texture gives a heavier feeling than a simple texture (Arntson, 1998). To create balance, the similarity of texture complexity should be considered. Rhythm is created by one or more elements in a design. It is used repeatedly to create a feeling of a structured pattern. To achieve rhythm, the similarity of texture has to maintain uniformity. The fitness scoring system divides texture into three types: natural texture, unnatural texture and biological texture. Natural texture is texture from nature, including rock and wood. Unnatural texture is texture created by humans, including plastic and metal. Biological texture is texture from living organisms, including leather (skin) and fur. To achieve unity, balance and rhythm, both primary and secondary textures should be compatible.

- ***The material section*** uses unity in scoring the fitness value of the chromosome. Similar to texture, repeating similar materials can create unity. A particular type of material has properties suited to a particular type of texture. Lambert material goes well with natural and biological texture types. Blinn is associated with texture from natural and unnatural types. Phong goes well with unnatural types, whereas Anisotropic is used with texture of the biological type. To achieve unity, material and texture should be combined appropriately. A positive score is given if the genes are combined according to the following associations of texture and material.

	Behaviour and Characteristic	**Texture**	**Material**	Total
Genes	●●●	●●	●●	
Scoring	3	2	2	7
Weighting	60%	20%	20%	100%
Criteria	Unity, Balance, Rhythm	Unity, Balance, Rhythm	Unity	

Fig.5.26 *An Example of Set of Attributes Chromosome Scoring System*

5.5 Generation of Harmonious Humanoid Characters

The population is replaced and reproduced until the algorithm termination conditions are met, implying that the number of harmonious individuals in a given generation is greater or equal to the required threshold. The data, captured as chromosomes, is exported back to the computer-aided design tool, Autodesk MAYA, and translated into the MEL-scripting language. Based on these GA chromosome data, the translation process then applies the same morphological, lexical, syntactic, semantic and pragmatic rules, which were used to generate the parents, to re-create the offspring (Fig.5.27). Table 5.4 shows an example of GA chromosome data of harmonious generations dividing into three groups according to their habitat. The three groups are terrestrial, subaquatic and celestial characters. The chromosomes in each group are represented in two colours: grey and white. The grey colour cells represent the parent chromosomes whereas the white colour cells represent offspring chromosomes. The harmonious characters are re-created in 3D based on their chromosome data and illustrated in Fig.5.28A, Fig.5.28B, and Fig.5.28C.

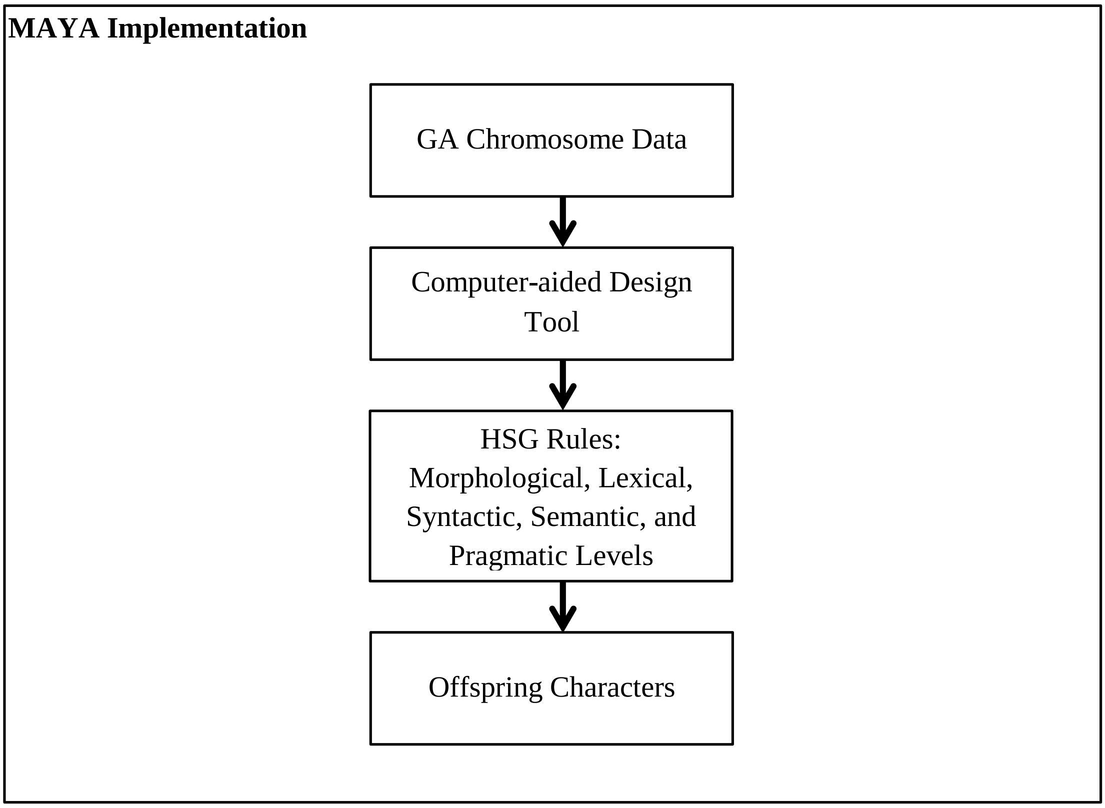


Fig.5.27 *MAYA Implementation: Component C Algorithm*

Table 5.4: *An Example of GA Chromosome Data*

Chromosome Chart																											
Anatomy Chromosome													Colour Chromosome								Other Attributes Chromosome						
HD	NK	TT	MT	BT	SH	UL	LL	EY	ER	TH	HB	PO	CHD	CNK	CTT	CMT	CBT	CSH	CUL	CLL	BE	CHP	CHS	TXP	TXS	MAP	MAS
Terrestrial																											
2	1	1	1	1	2	1	1	3	3	2	1	1	1	1	2	2	3	3	3	3	3	1	2	1	5	1	1
2	2	2	1	2	2	2	2	3	3	2	1	1	3	3	4	3	4	2	1	1	3	2	1	1	5	1	1
2	2	2	1	1	2	2	2	3	3	2	1	1	3	3	1	1	1	2	1	4	3	2	1	1	5	1	1
2	1	2	2	2	2	2	2	3	3	2	1	1	3	3	2	2	4	1	1	1	3	1	2	1	5	1	1
2	2	2	2	1	2	2	2	3	3	2	1	1	4	4	3	3	3	2	2	1	3	2	3	1	5	1	1
2	2	2	2	2	2	2	2	3	3	2	1	1	1	1	4	4	3	3	3	2	3	3	1	1	5	1	1
Subaquatic																											
2	1	1	1	2	1	1	1	3	1	3	2	2	6	6	7	7	6	12	5	5	1	2	2	1	5	1	1
2	2	2	1	2	1	2	2	3	1	3	2	2	7	7	6	12	7	7	6	6	1	2	1	1	5	1	1
2	2	2	2	2	2	2	2	3	1	3	2	2	5	5	6	6	5	7	11	11	2	1	3	1	5	1	1
2	1	1	2	2	1	2	2	3	1	3	2	2	6	6	5	7	6	12	7	7	2	2	3	1	5	1	1
2	1	2	1	2	2	1	1	3	1	3	2	2	5	5	7	6	5	5	12	12	1	1	1	1	5	1	1
2	2	2	1	2	1	2	2	3	1	3	2	2	7	7	5	6	7	12	5	5	2	3	3	1	5	1	1
Celestial																											
2	2	2	2	3	3	2	2	3	3	3	3	1	11	11	12	12	1	6	1	1	3	2	2	1	5	1	1
2	3	2	3	3	3	2	2	3	3	3	3	1	12	12	11	6	1	1	11	1	2	3	2	1	5	1	1
2	2	2	2	3	3	2	2	3	3	3	3	1	11	11	12	12	1	6	12	1	3	3	1	1	5	1	1
2	2	3	3	3	3	2	2	3	3	3	3	1	11	11	12	12	1	1	1	6	3	3	1	1	5	1	1
2	3	2	2	3	3	2	2	3	3	3	3	1	12	12	11	12	6	11	11	12	2	2	3	1	5	1	1
2	2	2	2	3	3	2	2	3	3	3	3	1	12	12	1	6	11	1	1	11	2	1	3	1	5	1	1

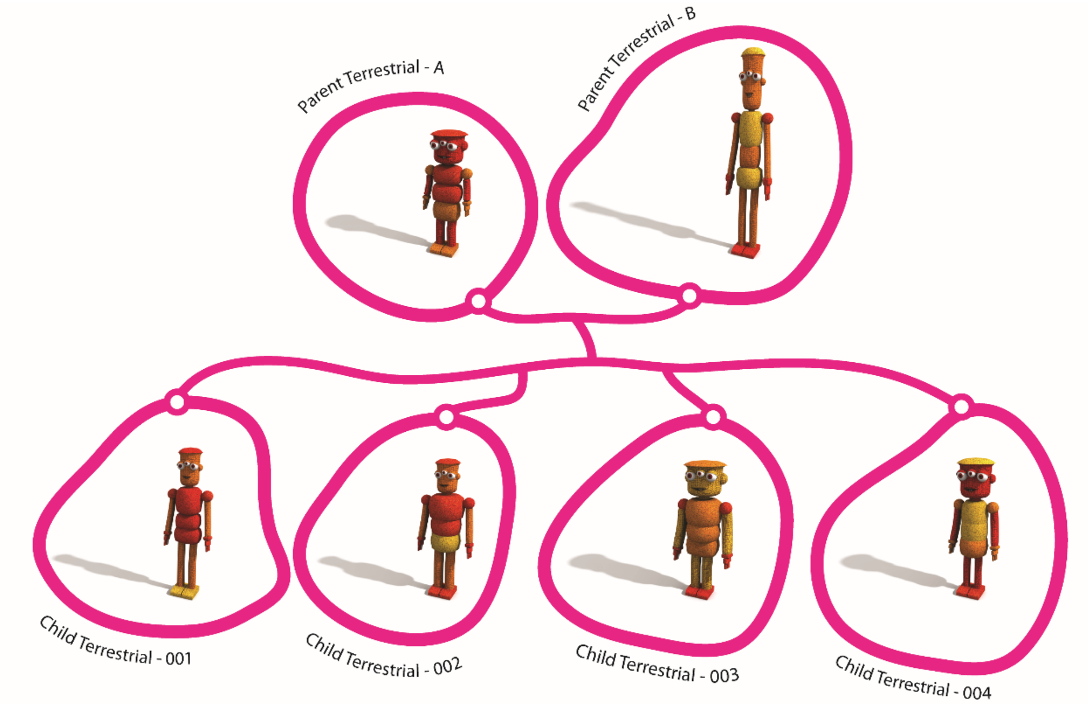


Fig.5.28A *An Example of Regenerated Terrestrial Humanoid Characters Imported Back to MAYA*

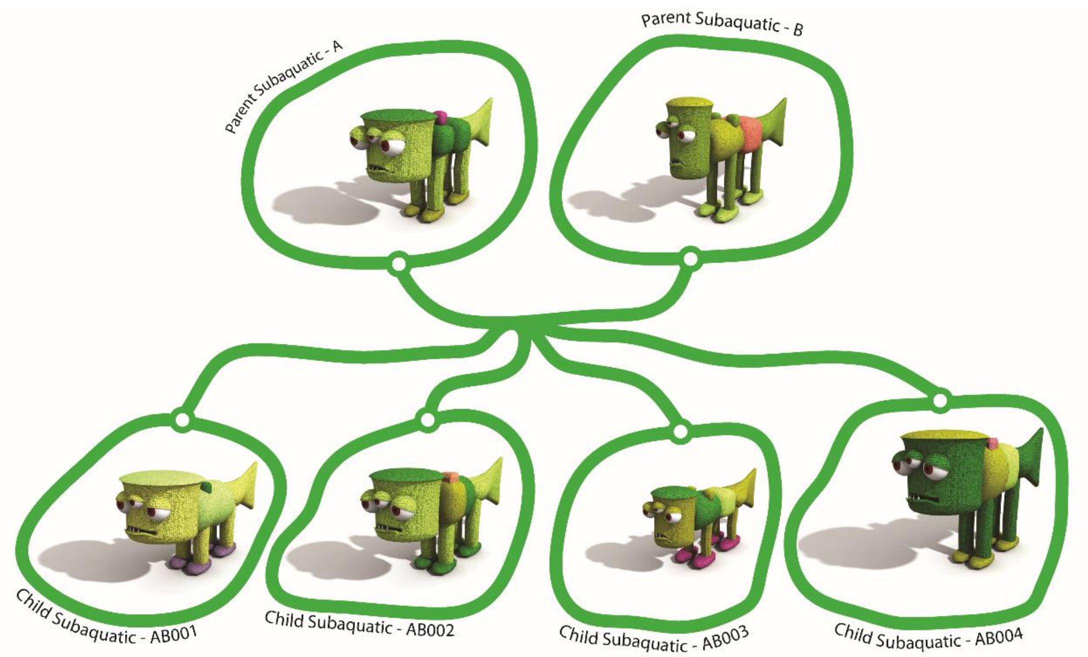


Fig.5.28B *An Example of Regenerated Subaquatic Humanoid Characters Imported Back to MAYA*

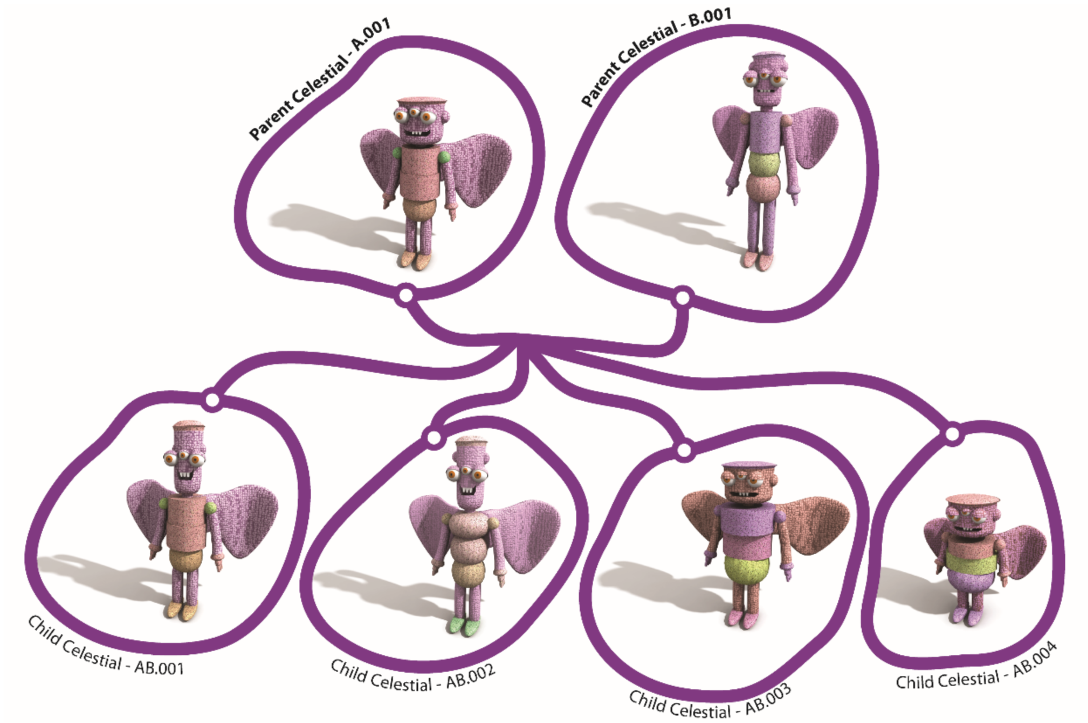


Fig.5.28C *An Example of Regenerated Celestial Humanoid Characters Imported Back to MAYA*

5.6 Evaluation of the Framework

The main aim of this thesis is to extend the existing shape grammar to generate a set of harmonised characters through the application of the five levels of analysis: morphology, lexicon, syntax, semantics and pragmatics. The generation of new characters should be harmonious in their design and appropriate to their environment. These generated humanoid characters should also show harmony in their anatomy, with relevant body components and shapes appropriate to their habitat and attributes. Finally, their characteristics and behaviour should also be in harmony with the other members of their family.

This section attempts to analyse the generated harmonious characters with respect to human perception in design. Norman (2004) explains that advances in our understanding of emotions and affectation have implications for the science of design. In assessing any product design humans apply three levels of design: visceral, behavioural and reflective. The first level is visceral and focuses on the overall appearance of a design. This is the first thing that comes to audiences' mind when they first see a product design. It is what that design looks like to them. This level is fast: we make rapid judgments of what is good or bad, safe or dangerous, and send appropriate signals to the muscles (the motor system) and alert the rest of our brain. It is concerned with character appearance. The second level is behavioural and is concerned with design function. This is the next step of brain processing. After seeing an appearance of a design, we start to think about its function. A good looking design character with bad functional components cannot convince audiences to believe and appreciate its design. The emphasis on accomplished behaviour is to create a design which encompasses both appearance and function. The reflective level is about self-image and personal satisfaction from our experiences. After seeing a shape and understanding the function of a given design, we determine our own overall impression of the product. The behavioural and reflective levels, however, may be very sensitive to experience, training, education and culture. Norman claims that the best designs come from following a cohesive theme throughout, with a clear vision and focus (Norman, 2004).

We shall apply the above levels to our generated characters. At *visceral level*, the appearance of our humanoid characters is related to their humanoid anatomy and corresponding habitat which is related to the morphological, syntactic and semantic levels. The humanoid anatomy is the first important configuration of basic shapes, shape rules and context which deliver a sense of what the character is to an audience. Character functions are divided into three sets: terrestrial, subaquatic and celestial. Terrestrial characters are designed with core components that are required for them to live on the ground. Legs and feet are necessary for activities such as walking, running, jumping, etc. Subaquatic characters, instead of having walking body functional components, have been designed with fins and tails to function appropriately in their environment. Similarly, celestial characters have wings attached to their bodies to assist in their flying ability. Characters have fulfilled the *behavioural level* by having their specific functional components to suit their design purposes. Character habitat defines where a character lives and this affects the configuration of the humanoid anatomy. Any missing important body component (e.g. a humanoid without a head) or unbalanced body proportions (e.g. head larger than torso) or inappropriate set of limbs for a given habitat (e.g. fins for a terrestrial humanoid) will result into a failed design. Humanoid anatomy and its relevant habitat contribute heavily to the cohesion and coherence in character design. The *reflective level* requires a consideration of design colours, texture, material, characteristics and behaviour. This level is difficult to evaluate objectively because it is based on the audience’s culture, personal experiences and self-satisfaction which influence their expectations from a given design object. To bring some objectivity to the design of humanoids the implementation has applied design principles regarding colour, texture and material harmony. Grammar shape rules provided unity, balance and rhythm for the personality and behaviour of the humanoids. This was achieved at pragmatic level. Applying genetic algorithms helped generate a realistic feeling for the audience and suggest that the humanoid members of a given family are compatible and exhibit an agreeable set of features and personalities, and thereby they belong to the same world or same story.

This thesis has extended shape grammar from the two initial levels, namely vocabulary and transformational rules to five levels, namely morphological, lexical, syntactic, semantic, and pragmatic levels. This extension has allowed us to design harmonious and contextual humanoid characters. Harmony has been achieved by including rules related to the principles of design, colour theories, shape proportion, and human perception in design, in particular the visceral, behavioural, and reflective levels. Visceral and shape proportion were applied at syntactic level to ensure that all shapes were created following the harmony criteria. Behavioural and principles of design were applied at semantic level to ensure that body components reflected unity and assembled in a meaningful way and appropriate to their habitat and environment. Principles of design, shape proportion, colour theories, and reflective were applied at pragmatic level to create characters with attributes, characteristics, and behaviour in harmony with other members. The new harmonised shape grammar has generated a set of harmonious parents for the terrestrial, subaquatic, and celestial humanoid species. To generate offspring, this thesis has applied genetic algorithm with a fitness function which selected members whose chromosomes complied with the principles of design. Only those harmonious new offspring were selected to produce the next generation. Finally, these chromosomes have been converted into MAYA parameter and have successfully generated a population of harmonious members for the three species.

5.7 Limitations and Conclusion

The implementation of our harmonised shape grammar was based on the ontological representation of humanoid characters expressed in terms of the five levels of analysis: morphological, lexical, syntactic, semantic and pragmatic levels. The aim is to generate a set of contextual and harmonious characters. The implementation involved three stages. The first stage generated terrestrial, subaquatic and celestial parents and expressed their character design in terms of chromosome data. The second stage subjected the parents' chromosomes to a genetic algorithm to generate the offspring. The three GA operators, crossover, mutation, and fitness functions were applied using harmony principles of design described in section 5.4.3. The third stage used the chromosomes, resulting from the GA, to reconvert them into 3D modelling data following HSG rules to generate the offspring guided by the design principles and harmony criteria. The implementation is discussed below:

- **Design detail and quality**

The term design detail is easily confused with design quality. Design detail refers to the level of detail of the design, whereas design quality refers to the level of effectiveness of the design function in describing a product's operational and design requirements. In this research, the quality of design has been achieved through semantic and pragmatic levels. However, the design detail is not implemented in this research. A humanoid's facial detail components (e.g. hair, eyebrows), and hand and foot components (e.g. finger nails, toe nails) are not developed. The implementation of HSG was aimed at achieving the concept art[1] or the concept design level of detail which is suitable for low-polygons 3D character modelling for commercial game applications. In order to improve the design detail of humanoids at the level of 3D characters, further refinement of the ontological representation is required in stage one and stage three.

- **GA chromosome representation**

Humanoid characters are encoded as 28-gene-chromosome data where 13 genes describe their anatomy, eight genes their colour and seven genes their other attributes. In order to convert their chromosome data back to a 3D geometrical

[1] Concept art is a form of illustration used to convey an idea for use in, but not limited to, films, video games, animation, or comic books before it is put into the final product.

format, this process requires two necessary data which are the chromosome data (the 28-gene-chromosome data) and the information from the rules that created their parents. By employing these two data together, a 3D geometrical character may be created. The information from the rules that created their parents is required to reduce the number of chromosome genes. The larger the number of chromosome genes, the more time the recombination process takes. However, as mentioned in the previous limitation, to improve the design detail additional genes would be required. Each cell in a human being has approximately 20,000 genes. Thus, in order for an implementation to capture all details of a character, the number of genes would have to be increased.

- **Animation**

The harmonised shape grammar has generated a set of harmonious humanoids. However, these characters are fixed in a single pose. By applying loop movements and further rules to their body functions, behaviour, and characteristics, animation could be implemented. The semantic and pragmatic levels could be further improved by adding character movements to capture emotion and gesture. The same principles of design (e.g. unity, balance, rhythm, emphasis) would have to be included in the loop movements, animation rules and functional features, to ensure harmony among the members of each humanoid family.

Chapter 6: Discussion and Conclusions

6.1 Introduction

The main motivation for using shape grammars is to support designers and provide computational aid to automate aspects of the design process. This support is designed to achieve efficiency, cost reduction, consistency and accuracy (Gu, 2012). The challenge in developing a shape grammar is to produce a design that meets the contextual constraints and does not require post editing to control the terminal shapes. This research project is an attempt at addressing the issues related to context and harmony in shape grammars. Its aim is to investigate whether the semantic and pragmatic levels of analysis in natural language processing can be integrated into an existing shape grammar to bring context and harmony into design, and, in particular, to character design.

Consequently, an alternative generative design grammar to provide context and harmony in design has been developed. Inspired by the natural language processing discipline, this thesis has extended the current notion of a shape grammar by adding two new stages: semantic and pragmatic levels. Contextual information brings meaning to the design whereas harmony brings cohesion and coherence to the design. Context in humanoid character design is achieved at the semantic level by generating humanoids whose anatomy is appropriate to their habitat and environment. Harmony in design has been focused on humanoids' attributes, characteristics and behaviour. This is realised at the pragmatic level.

This newly extended shape grammar is applied to generate the first set of parents of harmonised humanoid characters. The generation of their offspring is achieved using a genetic algorithm. A fitness function is used to select those members whose characteristics are in harmony with their parents and meet the principles of design. These principles include criteria related to unity, emphasis, balance, and rhythm, and others derived from colour theory, and also shape proportion rules. The selected members are evaluated using aspects of human perception in design (i.e. visceral, behavioural and reflective) (Norman, 2004).

6.2 Research Contributions

The contributions of this thesis are outlined below:

i. Integration of natural language processing and shape grammars. A shape grammar is a set of shape rules that can be applied to generate a set of designs. The aim of natural language processing is to develop computational algorithms able to achieve human-like language processing. This research builds a bridge between these two disciplines.

ii. Development of a harmonised shape grammar. A shape grammar includes two main levels: vocabulary of shape and shape rules. Natural language processing consists of six levels: prosody, morphology, lexicon, syntax, semantics, and pragmatics. A harmonised shape grammar integrates these two different disciplines and involves morphology, lexical, syntactic, semantic, and pragmatic levels. This brings context and harmony into a shape grammar.

iii. A novel computational approach of applying a harmonised shape grammar to designing objects based on an ontological representation of their concepts and relationships (e.g. humanoid characters).

iv. The generation of offspring from a pair of humanoid parents using a genetic algorithm whose fitness function is based on the principles of design and harmony.

6.3 Issues and Challenges in Implementing the Harmonised Shape Grammar

The new harmonised shape grammar consists of five levels of analysis. The morphological, lexical, and syntactic levels are derived from the existing shape grammar whereas the semantic and pragmatic levels are inspired by the natural language processing discipline. The aim of the first three levels is to generate shapes grammatically whereas the semantic and pragmatic levels are used to achieve context and harmony in design. The implementation of this humanoid shape grammar to generate a set of humanoid characters has met a number of challenges which are discussed below.

i. Challenge in Defining the Concept of Harmony in Design

Harmony in design is a subjective concept and influenced by culture and personal experience. It is often defined as a pleasing arrangement and combination of elements to form a consistent and orderly whole (Pettersson, 2013). In order to tackle this challenge, this research has defined harmony based on theories of design and colour principles. The development and evaluation of humanoid design have applied unity, balance and emphasis criteria to design shapes; unity, emphasis and colour theories to colour scheme creation; and unity, balance and rhythm to characteristics, texture and material issues. The final humanoid designs were evaluated using the visceral, behaviour and reflective criteria related to human perception in design, as advocated by the design community. Whilst visceral and behavioural design criteria focus on appearance and functionality of object design and can be measured in an 'objective' manner, reflective design relies on the designer's thought, skill and audience's culture. It is difficult to remove the subjective element as people of different cultures perceive design according to their traditions, needs and expectations.

ii. Challenges in Generating Offsprings

As the aim is to generate a family of humanoid characters in harmony with their members and environment, this research applied a genetic algorithm to generate a set of offspring which are diverse yet harmonious. To the inexperienced researcher the most challenging aspect relates to the chromosomal encoding scheme as this may have an impact on the quality of the selection process and convergence time. In this project, the humanoid characteristics are represented by 28 genes which are subjected to selection, recombination and mutation to produce new 'harmonious' members by applying fitness function rules based on the principles of design and harmony. The encoded genes of these harmonious members were then converted into MAYA Embedded Language (MEL-scripts) to produce the final design.

iii. Challenge in Defining an Ontological Representation of Humanoids

This thesis has taken a simple ontological approach to encode humanoid features. The ontological representation captures the main humanoid concepts with respect to natural language processing levels rather than the traditional anatomical ontologies which express entities in terms of their multiple relationships and inheritance between child and parent entities. Although the structure of this ontology could be easily adapted to produce different design objects, it has not exploited the full potential of what an ontology could offer. Each entity within the humanoid ontological representation could be developed further to include anatomical details such as additional facial components (e.g. hair, eyebrows, lips), additional body components (e.g. finger nails, toe nails) and motion capture.

6.4 Conclusions and Future Work

Shape grammars are used especially in computer-aided design applications. They are useful for generating a large variety of designs. However, they have serious limitations, in particular, they usually operate in limited experimental domains only and fall short in support for real designs (Chau *et al*., 2004), and lack support for high quality design and cannot guarantee aesthetic results (Huang *et al*., 2009). To address some of these issues, this thesis has developed an alternative generative design to provide context and harmony by adding semantic and pragmatic levels, to reduce time-consuming issues by capturing the characteristics of the desired object design through an ontological representation to control the design process, and to generate not only diversity but also quality in design. This thesis applies the harmonised shape grammar to the design of humanoid characters to test and validate this novel approach.

This research has addressed some gaps in shape grammars but sparked a number of important directions for further research, outlined below:

- **Animation**

Character animation could be created by describing the transformations of individual bones over a particular period of time which would involve the transformations of the bones' positions, rotations and scaling. These transformations which are referred to as animation tracking, could be stored as a dataset to reproduce, for example, a walk-cycle animation. It is proposed that the current research could be extended to include an animation track which could be specified according to a character's anatomy, body function, and habitat in the ontological representation and implemented at both semantic and pragmatic levels. This further extension would support the development and generation of humanoid and robots characters for the computer game industry which is facing new challenges in game design and content generation and could reduce the time and cost of the design and production process.

- **Social Relationship**

A social relationship could be used as a mechanism for context preservation (Cohen *et al.*, 1999) which allows an entity to behave differently towards different social actors (Tomlinson, 2005). A social relationship may be created as an emotion model. In the harmonised shape grammar, the social relationship could be represented at the pragmatic level as behaviour which includes neutral, friendly and depressive behaviour. There are various methods to define an emotion space. For example, Ekman's emotion model (1992) defines emotions by cardinal emotions, Mehrabian and Russell's model (1974) defines emotions by pleasure, arousal and dominance, Penksepp's model (1998) defines emotion space by drawing a distinction between basic emotions (e.g. anger and fear) and social emotions (e.g. love and grief). As future work, the different set of emotion models could be applied to change the behaviour of a character as and when required.

- **Genetic Algorithm**

The experimentation uses the inspiration of a GA to reproduce characters' offspring. However, more advanced GA techniques could be applied to improve the quality of solutions and convergence time with an improved chromosome encoding scheme.

- **Ontological Representation**

In this thesis, the ontological representation aims at creating a set of contextual and harmonious characters. The semantic level is used to understand character anatomy, body function and habitat. This information is stored as an ontological representation of a humanoid character. The integration of this representation with other ontological representations or development of a new one, would allow a change in design anatomy and design functions applicable to designs and production of complex characters for other application areas.

References

Agapitos, A. and Lucas, S. M. (2007) Evolving a Statistics Class using Object Oriented Evolutionary Programming, *Genetic Programming: EuroGP'07*, pp. 291-300.

Agarwal, M. and Cagan, J. (1998) A Blend of Different Tastes: The language of coffee makers, *Environment and Planning B: Planning and Design*, vol. 25, no. 2, pp. 205-226.

Aksamija, A., Yue, K., Kim, H., Grobler, F. and Krishnamurti, R. (2010) Integration of knowledge-based and generative systems for building characterization and prediction, *Artificial Intelligence for Engineering Design, Analysis and Manufacturing*, vol. 24, pp.3-16.

Allen, J. (1995) *Natural Language Understanding*, University of Rochester, The Benjamin Cummings Publishing Company, Inc., Redwood City.

Bentley, P. J. and Wakefield, J. P. (1997) Conceptual evolutionary design by genetic algorithms, *Engineering Design and Automation Journal*, vol. 3, pp. 119-131.

Bharati, A., Chaitanya, V. and Sangal, R. (2000) *Natural Language Processing: A Paninian Perspective*, PhD Thesis Project, Department of Computer Science and Engineering, Indian Institute of Technology Kanpur.

Brown, K. (1997) Grammatical Design, *AI in design*, University of Aberdeen, March-April, pp. 27-33.

Brychcin, T. and Miloslave, K. (2014) Semantic Spaces for Improving Language Modelling, *Computer Speech and Language*, pp. 192-209.

Burchett, K. E. (2002) Colour Harmony, *Colour Research and Application*, vol. 27, issue 1, pp. 28-31.

Cagan, J. (2001) Engineering Shape Grammars: Where we have been and where we are going, In Antonsson, E. K. and Cagan, J., *Formal Engineering Design Synthesis,* Cambridge University Press 2001, chapter 3, pp. 65-92.

Chakrabarti, A., Shea, K., Stone, R., Cagan, J., Campbell, M., Hernandez, N. V. and Wood, K. L. (2011) Computer-Based Design Synthesis Research: An Overview, *Journal of Computing and Information Science in Engineering*, ASME, June 2011, vol. 11, 021003(1-10).

Chase, S. C. (1999) Shape and Shape Grammar: From mathematical model to computer implementation, *Environment and Planning B: Planning and Design*, vol. 16, pp. 215-242.

Chau, H. H., Chen, X., McKay, A. and Pennington, A. (2004) Evaluation of a 3D Shape Grammar Implementation, JS Gero, Ed., *Design Computing and Cognition'04*, pp. 357-376.

Chevreul, M. E. (1861) *The Laws of Contrast of Colour: And Their Application to the Arts of Painting, Decoration of Buildings, Mosaic Work, Tapestry and Carpet Weaving, Calico Printing, Dress, Paper Staining, Printing, Military Clothing*, Translated by Spanton, J., New Edition, London: Routledge, Warne, and Routledge.

Duarte, J. P. (2005) A Discursive Grammar for Customizing Mass Housing: the case of Siza's houses at Malagueira, *Automation in Construction*, vol. 14, issue 2, March 2005, pp. 265-275, Elsevier Science.

Duarte, J. P. and Correia R. (2006) Implementing a Description Grammar for Generating Housing Programs Online, *Construction Innovation Journal on Information and Knowledge Management in Construction*, vol. 6, December 2006, pp. 203-216, Emerald Group Publishing.

Duarte, J., Rocha, M. and Soares, G. D. (2007) Unveiling the Structure of the Marrakech Medina: A shape grammar and an interpreter for generating urban form, *Artificial Intelligence for Engineering Design, Analysis and Manufacturing Archive*, vol. 21, no. 4, pp. 317-349.

Dunlap, R. A. (1997) *The Golden Ratio and Fibonacci Number*, Hackensack, NJ: World Scientific.

Fiedler, S. and Ilčík, M. (2009) *Procedural Human Posing Using CGA Grammars*, Institute of Computer Graphics and Algorithms, Vienna University of Technology, November 2, 2009.

Fiorentino, M., de Amicis, R., Monno, G. and Stork, A. (2002) Spacedesign: A Mixed Reality Workspace for Aesthetic Industrial Design, *The International Symposium on Mixed and Augmented Reality*, ISMAR '02, pp. 86-95.

Flemming, U. (1987) More than the sum of its parts: the grammar of Queen Anne houses, *Environment and Planning B: Planning and Design*, vol. 14, pp. 323-350.

Frowd, C.D., Hancock, P.J.B. and Carson, D. (2004) EvoFIT: A Holistic, Evolutionary Facial Imaging Technique for Creating Composites, *ACM Transactions on Applied Psychology*, (TAP), vol. 1, pp. 1-21.

George, A., Rajakumar, B. R. and Binu, D. (2012) Genetic Algorithm Based Airlines Booking Terminal Open and Close Decision System, *Advances in Computing, Communication and Informatics*, ICACCI'12, pp. 174-179.

Gips, J. (1999) Computer implementation of shape grammars, *NSF/MIT Workshop on Shape Computation*, MIT, Cambridge MA.

Girgis, M. R., Mahmoud, T. M., Abd El-Hameed, H. F. and El-Saghier, Z. M. (2013) Routing and Capacity Assignment Problem in Computer Networks Using Genetic Algorithm, *Natural Sciences Publishing*, NSP'13, no. 1, pp. 13-25.

Gobbel, G. T., Garvin, J., Reeves, R., Cronin, R. M., Heavirland, J., Williams, J., Weaver, A. and Matheny, M. E. (2014) Assisted Annotation of Medical Free Text using RapTAT, *Journal of Biomedical Informatics*, pp. 54-65.

Goldberg, D. E. and Rzevski, G. (1991) Genetic algorithms as a computational theory of conceptual design, *Applications of Artificial Intelligence in Engineering VI*, pp. 3-16.

Gruber, T. R. (1993) A Translation Approach to Portable Ontologies, *Knowledge Acquisition*, vol. 5, no. 2, pp. 199-220.

Gu, N. (2012) Towards an Integrated Generative Design Framework, *Design Studies*, vol. 33, issue 2, March 2012, pp. 185-207.

Halatsch, J., Kunze, A. and Schmitt, G. (2008) Using Shape Grammars for Master Planning, *Design Computing and Cognition '08*, part 8, pp. 655-673.

Harris, M. D. (1985) *Introduction to Natural Language Processing*, Reston Publiching Company; Inc., A Prentice-Hall Company, Reston, Virginia 22090.

Higman, B. (1969) A *Comparative Study of Programming Language*, Macdonald, The Press Syndicate of the University of Cambridge, Cambridge University Press.

Hoisl, F. and Shea, K. (2009) Exploring the Integration of Spatial Grammars and Open-source CAD Systems, *International Conference on Engineering Design*, ICED'09, Design Methods and Tools, vol. 6, pp. 427-438.

Huang, J., Pytel, A., Zhang, C., Mann, S., Fourquet, E., Hahn, M., Kinnear, K., Lam, M. and Cowan, W. (2009) *An Evaluation of Shape/Split Grammars for Architecture*, School of Computer Science, University of Waterloo, Ontario.

Ilčík, M., Fiedler, S., Purgathofer, W. and Wimmer, M. (2010) Procedural Skeletons: Kinematic Extensions to CGA-Shape Grammars, *SCCG*, Budmerice, May 13, 2010.

Jowers, I. and Earl, C. (2010) The Construction of Curved Shapes, *Environment and Planning B: Planning and Design*, vol. 37, no.1, pp. 42-58.

Kirshenbaum, E. (2000) Genetic Programming with Statically Scoped Local Variables, *Technical Report HPL-2000-106*, Hewlett Packard Laboratories, Palo Alto, 11 Aug, 2000.

Knight, T. (2000) Shape *Grammars in Education and Practice: History and Prospects*, The Department of Architecture School of Architecture and Planning, Massachusetts Institute of Technology, Cambridge, MA.

Koza, J. R., Andre, D., Bennett III, F. H. and Keane, M. (1999) Genetic Programming 3: Darwinian Invention and Problem Solving, *Genetic Programming: EuroGP'99*, pp. 93-108.

Krishnamurti, R. (1982) SGI: A Shape Grammar Interpreter, *Research report, Centre for Configurational Studies*, The Open University, Milton Keynes.

Krishnamurti, R. and Earl, C. F. (1992) Shape Recognition in Three Dimensions, *Environment and Planning B: Planning and Design*, vol. 19, pp. 585-603.

Krishnamurti, R. and Giraud, (1986) Towards a Shape Editor: The implementation of a shape generation system, *Environment and Planning B*: *Planning and Design*, vol. 13, pp. 391-403.

Kumar, M., Husian, M., Upreti, N. and Gupta, D. (2010) Genetic Algorithm: Review and Application, *Information Technology and Knowledge Management*, vol. 2, no. 2, pp. 451-454.

La Course, D. (1995) *Handbook of Solid Modelling*, McGraw Hill, ISBN 0-07-035788-9.

Langdon, W. B. (1998) *Genetic Programming and Data Structures: Genetic Programming + Data Structures = Automatic Programming!*, Springer Science and Business Media, Kluwer Academic Publishers.

Lee, H. (2007) *The Development of Parametric Shape grammars Integrated with an Interactive Evolutionary System for Supporting Product Design Exploration*, PhD Thesis Project, Industrial Design, Hong Kong Polytechnic University.

Lee, H. C. and Tang, M. X. (2004) *Evolutionary Shape Grammars for Product Design*, Hong Kong, Design Technology Research Centre, School of Design, Hong Kong Polytechnic University.

Lee, H. C. and Tang, M. X. (2009) Evolving Product Form Designs Using Parametric Shape Grammars Integrated with Genetic Programming, *Artificial Intelligence for Engineering Design, Analysis and Manufacturing*, Cambridge University Press, vol. 23, pp. 131-158.

Liddy, E. D. (2001) Natural Language Processing, *In Encyclopaedia of Library and Information Science*, 2nd Ed., NY., Marcel Decker, Inc.

Livio, M. (2002) *The Golden Ratio: The Story of Phi, the Extraordinary Number of Nature, Art and Beauty*, Headline Review, First Edition, UK.

Lorensen, W. E. and Cline, H. E. (1987) Marching Cubes: A High Resolution 3D Surface Construction Algorithm, *Computer Graphics and Interactive Techniques*, SIGGRAPH '87, pp. 163-169.

Madureira, A., Ramos, C. and do Carmo Silva, S. (2002) A Coordination Mechanism for Real World Scheduling Problems using Genetic Algorithms, *Evolutionary Computation*, CEC'02, vol. 1, pp. 175-180.

Mantyla, M., Nau, D. and Shah, J. (1996) Challenges in Feature Based Manufacturing Research, *Communications of the ACM*, vol. 39, issue 2, pp. 77-85.

Martinovic, A. and Van Gool, L. (2013) Bayesian Grammar Learning for Inverse Procedural Modelling, In *Computer Vision and Pattern Recognition (CVPR)*, IEEE Conference, pp. 201-208.

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*, vol. 16, pp. 273-290.

McGraw-Hill Companies (2002) *McGraw-Hill Concise Encyclopaedia of Engineering*, The McGraw-Hill Companies, Inc., RR Donnelley, The Lakeside Press.

Merrick, K. E., Isaacs, A., Barlow, M. and Gu, N. (2013) A Shape Grammar Approach to Computational Creativity and Procedural Content Generation in Massively Multiplayer Online Role Playing Games, *Entertainment Computing*, pp. 115-130.

Minglun, G. and Yee-Hong, Y. (2001) Multi-resolution Stereo Matching using Genetic Algorithm, *Stereo and Multi-Baseline Vision*, IEEE, pp. 21-29.

Nath, L. and Joshi, A. (2014) Parsing Algorithm for English to Indian Language Machine Translation System using Tree Adjoining Grammar Formalism, *Journal of Advanced Research in Applied Artificial Intelligence and Neural Network*, vol. 1, pp. 14-30.

Nitzberg, K. (2008) *Trave;ing Through Time and the Consideration of Design*, Smithsonian Cooper-Hewitt, National Design Museum.

Norman, D. A. (2004) *Emotional Design: Why We Love (or Hate) Everyday Things*, New York: Basic Books.

O'Neill, M., Swaford, J. M. and McDermott, J. (2009) Shape grammars and grammatical evolution for evolutionary design, *GECCO'09*, July 8-12, pp. 1035-1042.

Orsborn, S. and Cagan, J. (2009) Multiagent Shape Grammar Implementation: Automatically Generating Form Concepts According to a Preference Function, *Journal of Mechanical Design*, December 2009, vol. 131, pp. (121007) 1-10.

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*, Cambridge University Press, vol. 20, pp. 217-246.

Petterson, R. (2013) *Graphic Design*, IIID Public Library, Wien, Austria.

Poli, R. and Langdon, W. B. (1997) Genetic Programming with One-point Crossover and Point Mutation, *Technical Report CSRP-97-13*, School of Computer Science, University of Birmingham, B15 2TT, UK, 15 Apr, 1997.

Preda, M., Salomie, I. A., Preteux, F. and Lafruit, G. (2005) Virtual Character Definition and Animation within the MPEG-4 Standard, *3D Modelling and Animation: Synthesis andAnalysis Techniques for the Human Body*, IRM, Sarris N., Strintzis, M.G., Ed.

Pullen, K. and Bregler, C. (2002) Motion Capture Assisted Animation: Texturing and Synthesis, *ACM Transactions on Graphics*, TOG, ACM SIGGRAPH 2002, vol. 21, issue 2, pp. 501-508.

Renner, G. and Ekrárt, A. (2003) Genetic algorithms in computer aided design, *Computer-Aided Design,* vol. 35, pp. 709-726.

Riemmenschneider, H., Krispel, U., Thaller, W., Donoser, M., Havemann, S., Fellner, D. and Bischof, H. (2012) Irregular Iattices for Complex Shape Grammar Façade Parsing, In *Computer Vision and Pattern Recognition (CVPR),* IEEE Conference, pp. 1640-1647.

.

Rood, O. (1881) *Students' Text-book of Colour, or Modern Chromatics, with Applications to Arts and Industry,* Appleton and Company, New York.

Sadowski, P. (1996) *The Knight on His Quest: Symbolic Patterns of Transition in Sir Gawain and the Green Knight*, University of Delaware Press.

Sandstrom, K. and Norstrom, C. (2002) MANAGING Complex Temporal Requirements in Real-time Control Systems, *Engineering of Computer Based Systems*, IEEE, pp. 103-109.

Sapidis, N. S. (1994) *Designing Fair Curves and Surface: Shape Quality in Geometric Medelling and Computer-aided Design*, Society for Industrial and Applied Mathematics, SIAM Activity Group on Geometric Design, USA.

Saunders, M., Lewis, P. and Thornhill, A. (2012) *Research Methods for Business Students*, sixth edition, Pearson Education Limited, Essex, England.

Schnier, T. and Gero, J. S. (1998) From Frank Lloyd Wright to Mondrian: Transforming evolving representations, in I. Parmee, Ed., *Adaptive Computing in Design and Manufacture*, Springer, London, pp. 207-219.

Sederberg, T. W. and Parry, S. R. (1986) Free-form Deformation of Solid Geometric Models, *Computer Graphics and Interactive Techniques*, SIGGRAPH ’86, pp. 151-160.

Sharma, C., Sabharwal, S. and Sibal, R. (2013) A Survey on Software Testing Techniques Using Genetic Algorithm, *International Journal of Computer Science*, IJCSI’13, vol. 10, issue 1, no. 1, pp. 381-393.

Shea, K. and Cagan, J. (1999) Languages and Semantics of Grammatical Discrete Structures, *Artificial Intelligence in Engineering Design, Analysis, and Manufacturing*, vol. 13, pp. 241-251.

Stiny, G. (2006) *Shape: Talking about Seeing and Doing*, London: MIT Press, Cambridge, MA.

Stiny, G. and Gips, J. (1972) Shape Grammars and the Generative Specification of Painting and Sculpture, *Proceedings of IFIP Congress*, vol. 71, pp. 1460-1465.

Stiny, G. and Mitchell, W. J. (1978) The Palladian grammar, *Environment and Planning B*, vol. 5, pp. 5-18.

Stiny, G. (1994) Shape Rules: Closure, Continuity, and Emergence, *Environment and Planning B: Planning and Design 21*, s49-s78.

Talbott, K. (2007) Context-sensitivity and Visual Engagement in Generative Systems, *Automation in Construction 16*, vol. 16, pp. 54-60.

Tapia, M. (1999) A Visual Implementation of a Shape Grammar System, *Environment and Planning B: Planning and Design*, vol. 26, no. 1, pp. 59-73.

Teboul, O., Kokkinos, I., Simon, L., Koutsourakis, P. and Paragios, N. (2011) Shape Grammar Parsing via Reinforcement Learning, *Computer Vision and Pattern Recognition (CVPR)*, 2011 IEEE Conference, pp. 2273-2280.

Teller, A. (1994) The Evolution of Mental Models, K. E. Kinnear, Jr., editor, *Advances in Genetic Programming*, MIT Press, Chapter 9, pp. 199-219.

Toch, E. (1977) *The Shaping Forces in Music: An Inquiry into the Nature of Harmony – Melody – Counterpoint – Form*, Dover Publication, Inc., New York.

Tomlinson, B. (2005) Social Characters for Computer Games, *International Journal of Interactive Technology*, Special Issue on Social Learning Through Gaming, vol. 2, no. 2, pp. 101-115.

Trescak, T., Esteva, M. and Rodriguez, I. (2009) General Shape Grammar Interpreter for Intelligent Designs Generations, *Sixth International Conference on Computer Graphics, Imaging and Visualisation*, cgiv, pp. 235-240.

Westland, S., Laycock, K., Cheung, V., Henry, P. and Mahyar, F. (2007) Colour Harmony, *Colour: Design and Creativity*, vol. 1, no. 1, pp. 1-15.

Wilkins, M. R. and Kazmier, C (2005) *Mel Scripting for MAYA Animators*, Second Edition, Elsevier Inc., San Francisco.

Wong, K. C., Leung, K. S. and Wong, M. H. (2010) Protein Structure Prediction on a Lattice Model via Multimodal Optimisation Techniques, *GECCO'10*, pp. 155-162.

Woodbury, R. (2010) *Elements of Parametric Design*, Routledge, New York.

Yasuda, G. and Takai, H. (2001) Sensor-based Path Planning and Intelligent Steering Control of Nonholonomic Mobile Robot, *Industrial Electronics Society*, IECON'01, IEEE, vol. 1, pp. 317-322.

Zhang, W. and Lin, S. (2008) Research on Tibet Tangka Based on Shape Grammar, *Computer-Aided Industrial Design and Conceptual Design*, CAID/CD, pp. 373-376.

Appendix A: Publications

- Kunkhet, A., Sharp, B. and Noriega, L. (2012) Natural Language Processing Based Shape Grammar, *Natural Language Processing and Cognitive Science Workshop*, NLPCS.

- Sharp, B. and Kunkhet, A. (2010) Harmonised and Contextual Based Driven Shape Grammar, *Software, Knowledge, Information Management and Application*, SKIMA conference.

Appendix B: Mind-Map