A practical generative design method

Key words: generative design, parametric design, evolutionary design, computer aided conceptual design

Abstract

A practical method of exploring design possibilities on top of CAD systems is proposed. It is suitable for complex unquantifiable multi-criteria design problems where designers need to explore design alternatives within vast design spaces. Designs are generated by creating a genotype model within the CAD system and varying its parameters randomly within pre-set limits. These designs are then filtered through various constraint envelopes representing geometric viability, manufacturability, cost and other performance related constraints to ensure their viability. The genotype is based on successful designs embodying knowledge about the design problem and solutions to it. A distinguishing feature of this method is its ability to work seamlessly and harmoniously with current design practices from conceptual to detailed design. It is an interactive, designer driven method that is based firmly on the tenants of emergence. It makes minimal impositions on the designer's work practice and maintains both the flexibility and fluidly required for creative design exploration, especially at the conceptual stages of design. The design philosophy behind this generative method and the key steps involved in implementing it are presented here, with examples.

1. Introduction

The design of complex artifacts such as buildings or products requires designers to explore many alternatives. Currently, most of this exploration happens initially with the aid of pencil and paper, driven by a process that we have little understanding of. CAD is rarely used at this stage of the design process. In its current form, CAD remains a tool of implementation and increasingly a tool of analysis most useful at the later stages of the design process. But at this stage, all the important design commitments have already been made and significant improvements cannot be made. Can CAD help designers in the early stages of design to develop and explore design possibilities?

Generative design is about this. Though little known amongst engineers, *generative design* is now seen to be at the cusp of going main stream in architecture. Leading global architectural practices have embraced it[1] and most architecture programs teach it at Masters level. Generative design enables architects to explore thousands of design possibilities at various stages of design. Even though a concise definition of generative design is yet to emerge and formal methods are to be developed, it's significance is now widely recognized by architects and design researchers [2]. Proposed here is a particular method of implementing generative design on top of contemporary CAD systems that are parametric and history based.

1.1 Types of design problems that are suitable for generative design

As computer aided design is spread across many professions, encompassing a broad range of activities, we need to clarify here the type of design problems the proposed method is most suited for. Design problems may be broadly categorized as routine and creative design problems. Advances in design automation is gradually replacing the first category of design problems with fully or semi-automated design procedures. The second category of *'creative design problems'* remains elusive, mainly due to the inherent complexity of many design problems which arise mainly out of a multiplicity of design objectives, the contradictory and unquantifiable nature of some of these objectives, lack of complete domain knowledge and the vastness of the design space. However, most design problems have a combination of quantifiable and unquantifiable evaluation criteria. In some cases the evaluation of even the unquantifiable criteria is highly subjective (will differ from designer to designer). Aesthetic criteria are a prime example of this. In such cases, the design outcome will naturally depend very much on subjective choices made by the designer. In other words, it will reflect the designer's intensions and taste. The proposed method is suitable for this category of design problems. Architecture, product design, game design and animation design clearly belong to this category.

The proposed method is not suitable for most engineering problems where most of the key performance is computable or for design problems where it is possible to map between problem space and solution space. Genetic Algorithms (discussed in Section 2.2) is more suitable for this class of design problems.

1.2 The stages of design

In discussing generative design, it is important to establish the various stages of the design process, of which there are many. We are concerned here only with the extremities of the design process: conceptual (early stage) and detail (late stage) design. Many types of automated design exploration methods are available for late stage design. In this stage, important aspects of the designs are established and exploration is carried out within narrow bounds to improve specific performances. This is referred to as design optimization. Generative design on the other hand operates at the other end - the early conceptual stages of design. The ability to explore design variations at the early stages of design can produce far more beneficial results, than optimizing it within narrow means at the final stages of design. Most CAD packages now support analytical and optimization tools that are used extensively for late stage design. While the proposed method can be used for design optimization, its primary value is in its use in early stage design where CAD is currently rarely used.

CAD is mostly used in detail design. There are many conceptual and implementational challenges that prevent its use in early stage design. Despite decades of research and proposals made by academic researchers for structured conceptual design processes, they have not met with much success [4] in terms of industry adaptation. The reasons are explained in detail in this section. These issues need to be addressed in developing design processes for early stage design.

1.3 CAD and conceptual design

The creative design process remains highly unstructured and is currently unsupported by CAD systems. There has been a noticeable tendency amongst most engineering design researchers to view creative design processes as a somewhat inefficient and a "*haphazard process*". Many methods have been developed to eliminate this haphazardness by imposing a '*rational*' structure to it. These efforts remain largely unsuccessful [3, 4]. Proposals to apply structure to the conceptual stage of design have not met with much success, primarily because formalized processes *"impede the thinking effort by an invasive framework"* [3]. Freedom to create, modify and discard seems to be of paramount importance. Guidon shows that top-down breakdown is problematic for conceptual design[5] and implies that structured approaches that are suitable for routine design may be fundamentally unsuitable for conceptual design. This haphazardness is seen by Guidon as *"the natural consequences of the ill-structuredness of problems in the early stages of design"* [5]. Perhaps, this haphazardness should be viewed as positive indications of a creative process, where new learning and understanding of the problem and solution space, emerging out of the exploration process, is altering the course of search. The lack of it, on the other hand may indicate that both the problem space, solution space and the relationship between the two is well understood, effectively disqualifying it as a creative design problem.

1.3.1 Difficulties in supporting conceptual design in CAD

In addition to the inability to procedurize conceptual design, there are other reasons why CAD remains unsuitable for use in conceptual design.

1. At the conceptual stage, vague concepts and forms have to be considered and represented. CAD in its current form is unsuitable for representing vague concepts.
2. Designs are developed based on reactions to previously generated concepts. CAD does not provide the creative stimulation that designers derive from the process of hand sketching [6].
3. Design is an iterative process of searching the design problem space as well as the solution space. [7]. Designs and solutions co-evolve [8], during the design process.
4. Many possibilities are considered and most of them are discarded at the early stages of design. In this context, designers need to represent a wide range of concepts efficiently. They are reluctant to invest the additional effort required to represent concepts that CAD systems require.

There are also other cognitive, epistemological, methodological and computational issues that are discussed in detail [4]. If CAD is to be used in early stage design, the phenomena of early stage conceptual design needs to be better understood.

1.3.2 The centrality of emergence

Recent research[9, 10] in design process have identified emergence as the key driver of early stage design exploration. In creative design processes, the direction of design exploration is dependent on and is directed by the result of previous exploration which is the key characteristic of emergence. Creative design is based on reflection, reaction, critique and inspiration being drawn from the process itself. More importantly, it is very much dependant on the designer's internal representation and understanding. Oxman defines '*conceptual emergence*' as a search for *'The fit between visual images stored in the designer's mental image memory and the way the designer maps these images into a formal–configurational Schema'*[10]. She has experimentally verified the existence of hi level cognitive structures such as visual schemas and prop types that help designers think visually. Conceptual design relies heavily on the ability of the designer to identify emergent values. The designer relies on experience and understanding to identify solutions to the design problem within vague and unresolved concepts. In other words, the designer's ability and understanding is used to identify promising prospects within the vast expanse of search space at stages when the design is yet to be fully formed. The designer's creative imagination is relied upon here to complete the missing aspects of incomplete propositions. This is currently accomplished by the process of sketching.

1.3.3 The role of sketching

Sketching is central to most creative design processes where the human designer plays a central role in directing the design process. It helps trigger creative thought processes in exploring emergent concepts[11]. The main use of sketching during conceptual design has been found to be the stimulation of the designer's imagination. "*the designer does not represent images held in the mind, as is often the case in lay sketching, but creates visual displays which help induce images of the entity that is being designed*." [6, 12]. "*Drawn shapes play a critical role not only in representing a design concept but also in allowing the designer to re-interpret them to develop new ideas. In the conceptual and creative aspects of design, this re-interpretation of what has been drawn appears to play an important role* "[13]. This is why designers keep sketching as it both stimulates the creative process with emergent concepts and helps them to refine the concepts. Production of design ideas depend heavily on this interaction with conceptual sketches. This interaction is central to emergence[12].

It is precisely the emergent quality of conceptual design and its reliance on visual reasoning facilitated by sketching that is not supported by CAD. "*However, present computer-aided drawing, computer-aided drafting, and computer-aided design systems prevent the discovery of visual shape semantics. Such systems have inadvertently enforced fixation so that it is not surprising that they are not used in early stages of architectural design.*"[13]

1.3.4 Alignment with work practices

Conceptual design development is a process where many threads of possibilities are developed, abandoned and re-combined till a satisfactory scheme emerges out of the exercise. Often, this exploration is directed by the outcomes of previous explorations[12] . This emergent process appears to be a chaotic work process. One of the key challenges in the generative design process is to facilitate the fluidity of this process. Sketching seems to be the preferred process by which the designer navigates the solution space[11]. Out of such a seemingly unstructured search process emerges solutions that appeal to the designer's internal judgment or intuition[13]. CAD in its current form is unable to support this process. Hence, any proposal that is made for supporting early stage creative design should be able to operate in such chaotic conditions.

1.4 Supporting conceptual design in CAD

Supporting conceptual design in CAD is fundamentally difficult due to the paradigms of conceptual design discussed previously. Out of them, the centrality of the human designer in driving the design process, the non-procedural and emergent nature of the process and the inherent vagueness, incompleteness and ambiguity of early stage design need to be considered in developing CAD based conceptual design processes.

The failure of previous attempts at developing a conceptual design process is attributed to "*disparagement or ignorance of the importance of the human role and factors in conceptual design*" [4] . Many previous proposals have been procedural and mechanistic in their structure and have shown little understanding of the emergent nature of the process. Many CAD systems now, (e.g.: Sketcup, Alias) recognize the need for vagueness, which they somewhat allude to by imitating hand drawn sketch lines. Though it is a superficial gesture, it is a recognition and attempt at introducing the associated visual qualities of incompleteness which is an essential part of conceptual design.

1.4.1 Key requirements for supporting conceptual design in CAD

We identify and outline here key requirements that need to be met in order for CAD to support conceptual design.

1. Make minimal demands on and minimal disruption to current designers' work processes.
2. Be flexible in allowing the designers to navigate the design space in the way they see fit.
3. Be structured as an assistive tool, giving the designer the choice to either use it or not use it.
4. Support and enable emergence in ways that stimulate the creativity of the designer.
5. Enable an efficient transition of design content in to a detailed design phase.

Most importantly, it should work harmoniously with the designer's preferred practice with minimal disruption and not hinder the creative freedom that is necessary for creative design. The workflow should be driven by the designers in the order they see fit, depending on the type of the design problem and should support their own highly developed design method. In short, it should not have a pre-structured workflow. It should be 'non-mechanistic' [4]. It should not impose on the designer an externally conceived framework for design. Instead, it should support emergence which provides a fertile source of inspiration towards further exploration, similar in quality to the creative stimulation that designers derive out of hand sketching.

The important and often overlooked aspect of conceptual design is the generation of new knowledge about the design problem throughout the design development process. New understanding will inevitably require the continuous modification of concepts, evaluation criteria and constraints throughout the design development process. Ideally : "*The environment should support easy editing and re-organization of the requirements, design issues, and design decisions as the incompleteness and ambiguity of the problem specification are reduced through the design process*"[5].

1.5 Aims and objectives of the proposed method

The main objective of the proposed method is to enable human designers to explore efficiently larger range of design possibilities than what is manually possible for the class of problems outlined in Section 1.1. It is structured to stimulate the designer's creativity in guiding the designer through viable design spaces constrained through performance criteria. The proposition is also practical rather than theoretical in that, it needs to be designed with practical considerations in mind. It is to be implemented with minimum overheads on existing processes.

In addition it needs to work seamlessly throughout the entire design development cycle, from conceptual to the final stages of design. The chosen concepts should be easily transferable to the detailed design phase. It should be able to function without complete information about the design problem, which is often the case in early stage design; where the design problem itself is under formulation. In supporting the creative work process of designers it should enable designers to work on limited aspects of the design. It should allow the designer to explore selective regions of the design space at various levels of detail.

The remainder of the paper is organized as follows: in the next section we discuss the research that is related to the proposed approach. In the following section we present the theoretical frame works for the proposed generative design method along with methods of implementation. Details of the steps are provided. We then compare the proposed method with the genetic algorithm based method in the generation of a coffee table design. We explore other applications and conclude with a discussion on further research.

2. Related works

The theoretical frame work of the proposed method draws heavily on previous research on constraint driven parametric search and genetic, evolutionary algorithms. It may be seen as a "*Creative Evolutionary System*" [14] or as "*Interactive Evolutionary Systems*" (IES) [14], [15], as it is designed to aid human creativity, but with important distinctions that we discuss in this section.

2.2 Genetic Algorithms

The most dominant method in computational design exploration is Genetic Algorithms. A compilation of Genetic algorithm based methods in CAD is given in [15]. Genetic Algorithms require the maintenance and breeding of a population of designs, which are evaluated by fitness function. The proposed method does neither and therefore is distinctively different from Genetic Algorithms. In the proposed method, the evolution of the design is driven entirely by the designer, who by constantly adding design details, constraints and evaluation criteria, modifies both the design representation and the search space until valid designs are found. In GDM, the genotypes are CAD models and the phenotypes are instances of it, thus the mapping between the two is direct. The other key differentiation is the intent. The proposed method is designed to search for a multiplicity of viable solutions that are different to each other; for the designer to select, instead of a selection of a single optimal solution. Despite these differences, GDM shares with Genetic Algorithms the concept of a genetic model, performance space, mutation and selection criteria. Genetic Algorithms that rely on numerical optimization are fundamentally unsuitable for design problems that require complex evaluation criteria. "*optimization methods can only respond to the objective parameters that are coded into the problem, and as a result, non coded parameters, such as aesthetics, historic context, or meaning are left out of the optimization problem, and as a result finally left out of the final design solution*"[16].

2.3 Generative Design

Research in what is now known as generative design, was pioneered by Frazer[17] in the early 1970's. Its development has gone through various phases, led by mainly academic researchers focused largely on design theory. Though the lack of methods of implementation was recognized [18], formal methods for generative design were not forthcoming. With the growing interest of practitioners and schools of architecture, this gap was filled by CAD companies[19-21], [22] offering various generative design solutions. The current aim of generative design is best summed up by Shea; "*generative design systems are aimed at creating new design processes that produce spatially novel yet efficient and buildable designs through exploitation of current computing and manufacturing capabilities*" [2].

The need [23] , [18] for a guiding theoretical frame work for generative design was now felt by leading researchers. *" Given the growing amplitude of issues and subjects in digital design as witnessed by both practice and publication, we are faced with the need to formulate a theoretical framework that is suitable to the conceptualization of the subject* "[24]. In this context, a particular form of generative design - the Generative Design Method (GDM) is defined here as - a designer driven, parametrically constrained design exploration process, operating on top of history based parametric CAD systems that are unstructured in terms of design development and supportive of emergence.

Generative design is applicable to both parametric and procedural design. We do not make this distinction in this paper as contemporary CAD systems use internal procedures to construct geometry, though the procedural aspects are hidden from the user.

Generative design requires the following components:

1. A design schema
2. A means of creating variations
3. A means of selecting desirable outcomes

Marsh [25] classifies the same with greater emphasis on performance measurement as '*Configuration Variation'* (1 & 2 combined), *'Performance Metric '* (aspects of 3)*, and Decision-Making Response'* (3). The overall scheme of a generative system (operating on procedures) is given by Bohnack et al.[26] and is implemented in a java based scripting language named '*processing*'[27] . A modification has been made to his diagram (Fig.1) to make it relevant to CAD based generative design by the inclusion of a generic model as an alternative to algorithms and the inclusion of constraints as an alternative to the modification of source code, which is not done on CAD based generative systems . It is important to note here, the central role of the designer in continuously modifying the generative scheme based on the resultant outcomes; by which the solutions space is navigated in search for viable design solutions.

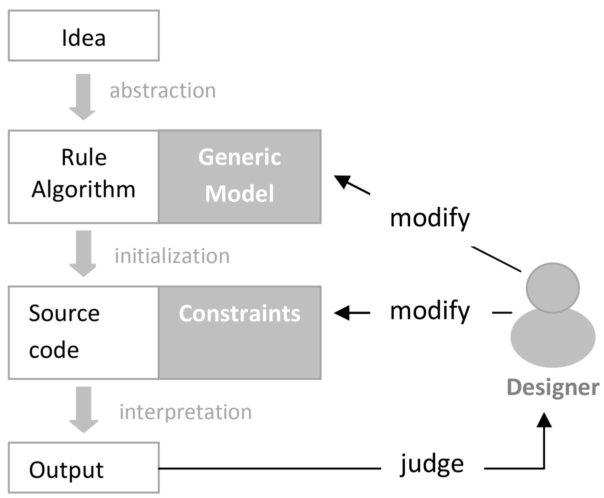


Fig.1. Generative Design Process

2.4 Related methods

A wide range of methods and modeling techniques are related to the proposed method. The closest amongst them are discussed here.

2.4.1 Constraint Based Evolutionary Decision Support System

A Constraint based cooperative, interactive design method using genetic algorithms has been proposed by Guoyan[28] for early stage conceptual design. It treats the design process as an interlaced combination of constraint optimization, modeling and optimizing the scheme for searching and evaluating designs. In short, it is an interactive design development process driven entirely by the designer. The designer interacts specifically with (1) variables, (2) constraints, (3) objective functions and (4) search strategy. It strategically separates the design tasks into quantitative and qualitative tasks. The quantitative tasks are left to the computer and the qualitative processes are handled by the human designer so that "*respective advantage and characteristics of human and computer in the design process*" [28] can be used to the best advantage. It relies on Genetic Algorithms for generating design solutions.

2.4.2 GENE_ARCH

Though GENE_ARCH has been specifically developed as an evolution based generative design system for sustainable architecture[29], the same principles are applicable to a wide variety of conceptual design problems. Caldas[29] demonstrates how building performances can be greatly increased by combining parametric generative schemes and building simulation software to evaluate thermal performance. He uses fitness functions and Parento Genetic Algorithms to optimize the chosen multi-criteria design problem. The generative scheme seems to interface directly with CAD systems to create the variations required for thermal and lighting analysis.

2.4.3 Intelligent Genetic Design Tool

The Intelligent Genetic Design Tool (IGDT) has been proposed by Buelow [16], to aid in early stage design exploration processes, intended to stimulate the designer's creativity and to avoid convergence and encourage divergence in design exploration . His intention is to thoroughly explore the design space and make it apparent to the designer. It is also a designer driven process and does not rely totally on quantitative criteria. It is suitable for form finding problems in structural design where the design space is searched through the use of two layers. An outer layer representing the topology (connection patterns) and the inner layer representing the geometry of the structure. Both layers are algorithmically searched. IGDT is designed to dynamically adapt to evolving design criteria and to assist designers in the early conceptual design phase.

2.4.4 Genetic Algorithm Designer (GADES)

A Genetic Algorithm Designer was proposed by Bentley[30] for the evolution of creative design. It evolves shapes from random blobs. A phenotype is first specified based on the design space and the genotype is specified based on the solution space. A suitable evolutionary algorithm is then chosen and the fitness function is defined. Multi-objective genetic algorithms are then used to evolve the solutions. This is a classic application of evolutionary algorithms for form design. The generated results are compared in section 4.

2.4.5 Shape Grammar

Shape Grammar is a method of representing geometry and embedding geometric logic in design representations. However, the sophisticated geometry and constraint modeling capability of modern CAD systems seem to have subsumed [31] t, he original intent of Shape Grammar. In other words, much of Shape Grammar can now be built into CAD models, using native CAD functions. Shape Grammar also suffers from the "*computational complexity of grammars and the difficulty in developing useful interfaces*"[23] . Despite being developed more than 30 years ago, its adaptation by industry is limited due to various reasons. Deak [32] points out the requirement of a specialist grammar modeling knowledge, the "*understanding of the grammar by anyone who was not involved with its creation difficult*." He proposes '*CAD grammars'* - a CAD based encoding of grammatical rules that combines the shape and graph grammar to create what he calls '*designerly grammars'.* The use of "*CAD grammars allow a homogenous model to be used for the design representation and the tools to generate it*", making it much more applicable in CAD environments.

2.4.6 Other related methods

Shea has proposed *efiForm,* a generative structural optimization method based on structural topology and shape annealing [2] . *efiForm* demonstrates the use of structural grammars, performance metrics, structural analysis, and stochastic in creating structural form. This is a highly structured but interactive process where design generation is carried out by grammatical design transformations. An L-system based generative grammatical encoding has been developed by Hornby [33] who is able to demonstrate that the generative encoding of the genetic model is able to create significantly fitter solutions than non-generative encoding. He is able to demonstrate this in the design of a table. A neural network based approach has been proposed [34], to enable the system to learn the preference of the designer in selecting designs. If this method can be implemented across a range of design problems, it will enable to reduce the cognitive load in the selection process. The search aspect of the proposed method is closer in many ways to what is known as a morphogenetic approach [35], which focuses on the dynamics of growth. More advanced evolutionary computational approaches called "Computational Embryology" [36] which is currently being developed, is likely to greatly improve the representational capability of computational genetic models.

3. Generative Design Method (GDM)

The Generative Design Method (GDM) is a comprehensive CAD based generative design exploration method designed to work at all stages of the design development process – spanning from conceptual to detailed design. The GDM is composed of six key components:

1. **Genotype** – is composed of a generic parametric CAD model, list of design parameters and their initial value and initial exploration envelope.
2. **Phenotype** – generated CAD files (that may include build history, built-in relationships and built-in equations).
3. **Exploration envelope** – a list of minimum and maximum values of the driving parameters specifying the limits of the design space to be explored.
4. **Design Table -** a data table that stores the driving design parameters, their initial values and the limits and other data that may be required and the generated design values preferable in an accessible spreadsheet format.
5. **Design Generation Macro** – a macro or a spreadsheet function that operates on the design table. It generates random variations of the driving parameters within limits set by the initial design envelope.
6. **CAD system** - is a parametric CAD engine with a transparent and editable build history, preferably with a 3D geometric kernel with capabilities to manage geometric relationships, engineering equations and connect to external design tables.
7. **Performance filters** – A pass/fail software filter, that is able to evaluate the performance of generated designs based on data from the design table, CAD system or associated analytical packages.

We now briefly describe how these components are connected (Fig.3) to create a generative design system.

3.1 A brief description of the process

The steps in implementing the Generative Design Method are:

1. Creating the genetic model
2. Setting the initial envelope
3. Generating designs
4. Filtering phenotypes
5. Selection & fine tuning

A generic CAD representation of the design is first created in a parametric CAD system. The driving dimensions are then set with an initial value (using the native dimensioning system of the particular CAD package) and stored in the design table (Table. 1). The maximum and minimum range of these values are then set (individually or as a percentage value) to limit the search within an exploration envelope.

This is then collectively referred to as *genotype*. The genotype here represents the design space and the limits of search in a format that is operable in CAD.

The Design Table Macro then generates random values within the exploration envelope. The CAD system then generates new instances of the designs based on these values. The generated designs are referred to as *phenotypes.* Performance filters are then used to judge the viability of these phenotypes. These pass/fail filters are constructed out of key requirements that define the viability of the design. The phenotypes that pass through these filters are then considered viable designs. The filters draw values that are directly related to the design parameters such as distances, which may be drawn from the table and values such as volume and weight may be drawn from the CAD package. If the CAD system posseses a geometric kernel that is able to detect build- failure then it may be used as a geometric filter. Proximity filter may be used to filter out designs that are similar to each other to ensure that the generated designs are somewhat dissimilar. These steps are described in greater detail (Section 3.3.1 – 3.3.5).

All designs generated may be saved and retrieved for comparison or design refinement depending on the work process preferred by the designer. GDM also allows the designer to explore design possibilities interactively around a generated design. This is accomplished by setting the generated parametric values of the *phenotype* as a new *genotype* and by re-setting the initial envelope to cover a smaller region of interest (Fig 3).

3.1 Representation of the design problem

The design problem here is represented as a constrained parametric search problem, where design solutions are explored based on a parametric representation of a design which in itself evolves throughout the design development process.

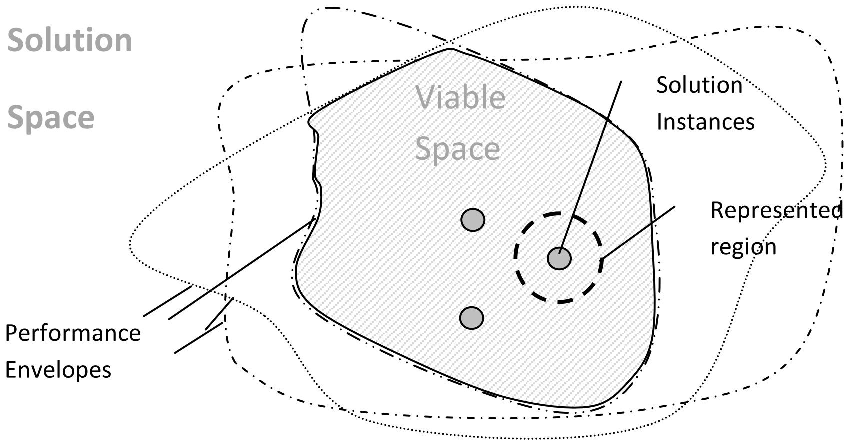


Fig.2. Representation of solution space.

The design exploration is defined here as the search for viable designs within a series of performance envelopes. The performance envelopes represent the parametric limits of the design that satisfy specific requirement. The viable design space is then the intersection of various performance envelopes.

In generative design, designers are faced with the problem of selecting amongst thousands of designs. This places a significant cognitive burden on the designer. The designer is able to assess only a limited number of design solutions without cognitive fatigue [16]. Hence, the designs presented to the designer for assessment have to be limited in number and widely dispersed with the viable design space. Each instance may be taken to represent a region of design possibilities as shown (Fig.2). Such an approach makes it possible for human designers to explore large regions of design space based on a limited number of design instances.

GDM allows for the continuous evolution of the solution space. Since the parametric representation of design makes the design space easily navigatable, accuracy of the envelopes representing the performance limits becomes less of an issue in the early stages of exploration, where the focus is mainly on the identification of viable regions. Once the final design is chosen, the region around it can be examined in finer detail and the limiting performance envelopes can be defined with greater certainty

3.2 Implementation

The GDM is a search process that allows the designer to explore designs within the Exploration Envelope. The proposed arrangement affords great flexibility. It allows the designer to change the CAD model, the genotype values and the filters at any stage of the design process. It also allows the designer to apply GDM methods to selected features of the design. It is this feature that enables it to support conceptual designs where the genotype and performance criteria are still under evolution.

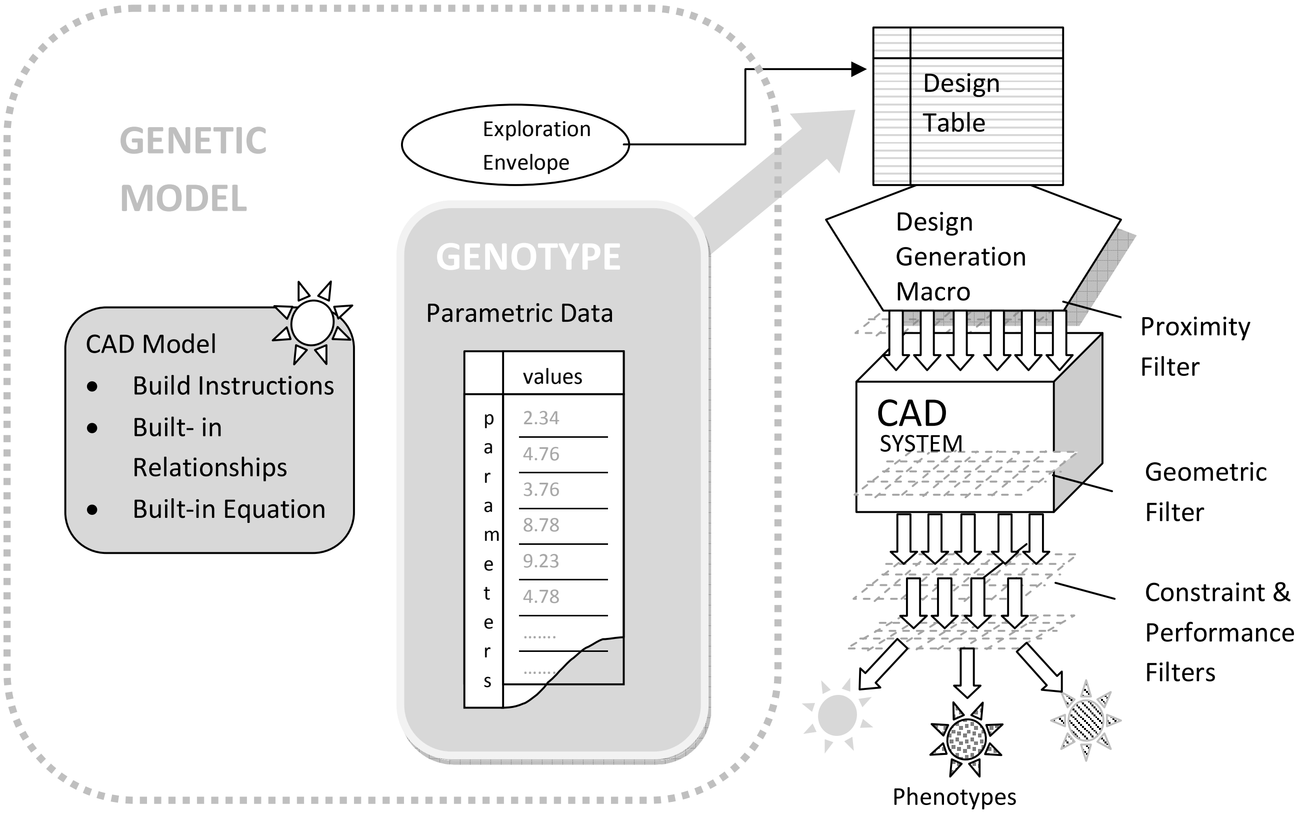


Fig.3. Generative Design Method – the overall scheme.

3.3.1 Creating the Genetic Model

The genetic model in GDM is essentially a CAD file that can be defined as *"the representation of a family of objects that share the same topological constraints but have different geometry"*[37]. The genetic model needs to capture the design intent defined as "*the functional requirements provided by the customers; that is a set of geometric and functional rules which the final product have to satisfy. The design intent is represented by parameters, constraints, features and design history*"[38].

The genetic model needs to engender a rich set of design possibilities as it represents the design space. The genetic model should not only capture the common (generic) geometry of the desired designs but also the common underlying patterns behind the geometry. Nature provides many great examples as to how geometric variations can be created while maintaining an underlying structure. While there are about 300,000 beetles that appear to be very different from each other, they all share a pattern of relationships that is common and constant.

A well structured genetic model will be able to represent a much wider range of design variations than a poorly structured model. The genetic model needs to be robust and hold its geometric logic while being subjected to significant and unpredictable random variations during the generative stages of the design. An example of a family of designs generated [39] out of a single genetic model is shown (Fig. 15). The build sequence of the CAD model also plays an important role in determining its robustness and thus the variability of the model. The underlyingbase geometry needs to be structured early on (or higher up) in the design tree and all the less important features at the latter stages or lower down in the design tree. Since most CAD systems construct the geometry sequentially, such an arrangement would prevent non-critical failures of the less important aspects of the design, invalidating the entire design.

Developing the genetic model is a design exercise in itself and it would require quite a few iterations. Janssen identifies some of its key attributes. “ *In order to evolve challenging designs, the variability of the designs must be carefully controlled so as to ensure that the designs are complex, intelligible, unpredictable and desirable*” [40]. The genetic model is also an embodiment of knowledge of the design problem and solutions to it. [41]. Structuring genotypes with performance objectives is a way of embedding design knowledge into the genotype itself.

The genetic model may contain some embedded requirements. E.g. if the intention is to design a bottle for containing one litre of liquid, one of the driving parameters could be determined by equations within the CAD environment to ensure its compliance. By embedding such requirements into the genetic model, we can be assured that these requirements are met in all generated solutions. But if there are too many of such built-in requirements, we face a real danger of creating an over constrained generative model and a reduced search space.

In setting up the initial parameter values, the commonest form of the design needs to be considered. Generative Design relies on the computer’s capacity to reach out into the outer edges of the design space and produce unexpected and creative results. To achieve this, the start state of a genetic model (genotype) should be somewhat in the center of the design space representing the most common design. This can be achieved in cases where the commonest form is known; e.g. wine glasses. In other words the designer should avoid extreme representations when creating the initial instance of the genotype.

3.3.2 Setting the Exploration Envelope

The Exploration Envelope is set at minimum and maximum values to limit the design search space. These limits need to be set with approximate values as their main purpose is to prevent the waste of computation energy in exploring unviable regions of the design space. Hence, the initial envelope is set based on the designer’s understanding of the geometric limits of feasible design outcomes. Detailed constraint envelopes will further limit the exploration space but in regions that it does not, the initial envelope can be set to act as the default constraint envelope. The initial envelope can also be set as a ***n***-dimensional envelope where ***n*** is the number of parameters. But for now, it is set as an independent, single dimensional, minimum and maximum value.

3.3.3 Generating designs

Design generation is a fairly simple process and it is accomplished through functions available in conventional spreadsheets such as Excel.. The same can be accomplished by embedded Macros, which we refer to as the Design Generation Macro. Generated designs that are too close to previously generated designs can be discarded using the proximity filter (Fig. 3) to ensure that the generated designs are dissimilar. While this may not be the optimal approach, it ensures to some extent, the diversity of generated designs. A better approach would be to assess the generated designs, to ensure that they are spaced with sufficient distance from each other in the performance space. A measure of geometric differences could, for example be used to ensure visual differentiation of generated designs. But such strategies will require the generation of a large number of designs before they can be assessed, making it computationally costly.

3.3.4 Filtering phenotypes

Developing the accurate constraints envelope will require significant investment of effort. Initially, it is sufficient to set the constraint envelope with reasonable accuracy, primarily as a bounding region to set the limits of design exploration. Once the candidate solutions are identified, the constraints in these regions can be reset with much greater accuracy. GDM allows for the adding, deleting and modification of constrains in the form of filters throughout the design development.

Most parametric CAD packages have built-in engineering functions and connect seamlessly to a slew of analytical packages that may now be used, to assess various performance aspects of the design. The comprehensive mapping of the constraint envelopes for the initial design space may be computationally costly, but it will speed up the evaluation process. Though it is possible to pre-compile these envelops, it is not recommended; mainly because the genetic model undergoes significant development during the design process. The order of assessment may be determined by computational cost and work process issues that are discussed (Section 3.5).

3.3.5 Selection and fine tuning

The optimal generation strategy would be to first generate design solutions that are placed significantly apart in the performance space. Once the high potential designs are identified, the regions around them can be explored in much finer detail. In exploring design space, the designer faces the same set of problems faced by a geographer in mapping new and uncharted territory, looking for minerals that are found in only regions with particular combination of geological characteristics. Many parts of the unmapped territory would represent continuous stretches of unremarkable designs, lacking in value or novelty. There may be regions where interesting things begin to happen. Hence, an experienced geographer will first set out to understand the “lay of the land”. The exploration of performance space requires a similar approach. The designer will develop a mental map of the design solutions space and be able to move from one design to another (Fig. 4).

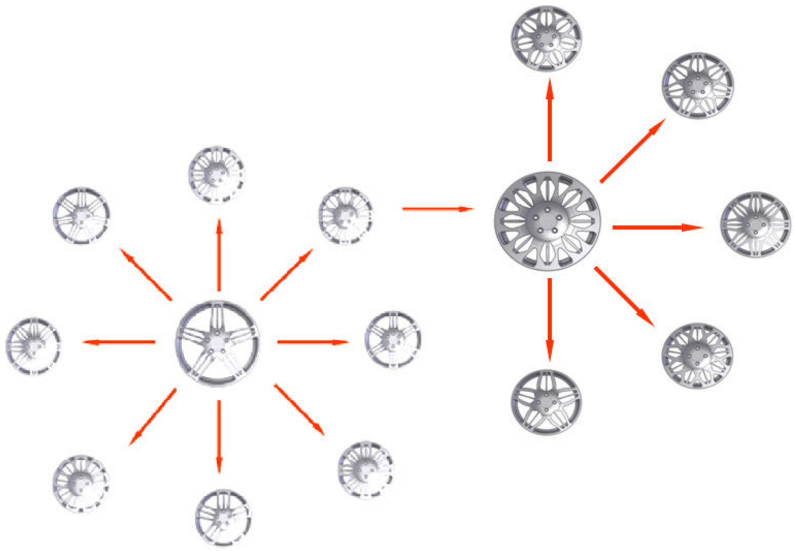


Fig. 4. Exploring the design space

Performance may become erratic in certain regions. These could well be high potential regions as performances rise or drop dramatically. Then, beyond a point the designs would hit the edges of the constraint envelopes. These regions are also interesting, as they could be regions where performance could not be further increased due to the limitations imposed by constraints. By negotiating these constraints, the designer may seek to achieve novelty or increase its performance. If the designs are to be evaluated visually, they may be rendered realistically or be rendered real time in 3D. Once the designs are chosen, they can be fine tuned easily due to their parametric nature. Variations in texture and color may also be explored at this stage.

3.4 Software implementation

GDM is a design method that is reliant on the capabilities of contemporary CAD systems. Current CAD systems are mostly history based. The modeling and file structure is dependent on build history as in STEP standard (ISO 10303) [11], hence the term CAD here is used to refer to procedural parametric CAD or history based CAD. Most modern CAD packages with geometric kernels have abilities to set up geometric relationships within the part file. They also have equation editors that allow designers to create equation driven relationship between dimensions. These two features can be used with great advantage in GDM.

Though many CAD packages have means of storing data internally, an external data storage approach was preferred for reasons of transparency and for easy connectivity to other analytical packages. The design table is essentially a spreadsheet with basic spread sheet capabilities which include functionalities to generate random numbers and abilities to work with internal or external macros. Alternatively, a data file can be used with a program that can achieve the same results. Three ways of implementing GDM is discussed here.

3.4.1 Software free implementation

GDM can be implemented without any additional software with a CAD package that is able to communicate directly with a spreadsheet such as Microsoft XL. CAD packages such as SolidWorks™[42] are able to create, save and read external XL based design tables.

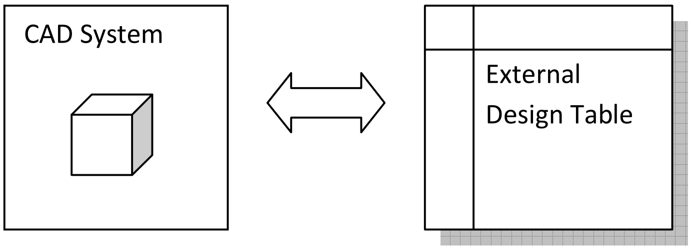


Fig.5. Software free implementation of GDM

The driving parameters are stored in the Design Table using built in CAD functions. The designs are then generated using a simple function that creates a random value between two numbers. In the case of Microsoft Excel the RANDBETWEEN function can be used.

B8 = RANDBETWEEN(B5,B6) (1)

A random value between two values stored is cells B5 & B6 which is then stored in cell B8. The table is may be structured as shown on (Table.1). The maximum and minimum values B5 & B6 can be based on a percentage of the initial value or it can be set by the designer. The generated values in this design table are then read by the CAD program to create the generated instances of the design. Additional filters can be implemented within the XL table using simple XL functions to filter out (delete) the designs that fail the set criteria.

3.4.2 Spreadsheet macro based implementation

Most spreadsheets have enabled the authoring of Macros that can be sued for generating the new designs. The advantage of this approach is that the same macro can be used for various generative design projects. They can also write the data in formats required by other analytical packages.

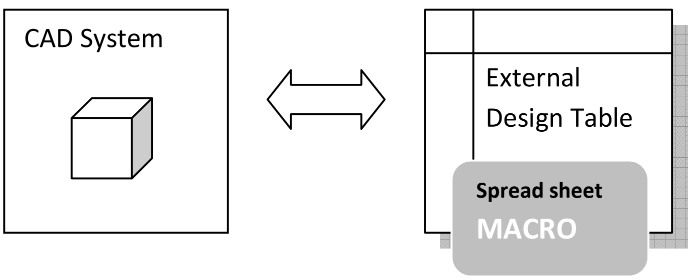


Fig.6. Spreadsheet Macro based implementation of GDM

Examples of this is available in www.opengenerativedesign.com [43]. Filters too can be structured as macros. The read/write facility of the CAD package is then used to create the generated instances of the design. One advantage of this approach is the separation of the Generative scheme from the CAD package. Open Generative Design [43] provides examples where the same design table is used by both SolidWorks™[42] which posses a geometric kernel and Rhino/Grasshopper™[20] which does not have a geometric kernel. The stated aim of the Open Generative Design Initiative is to develop CAD independent methods of implementing generative design. This will enable the building of common data structures that can be shared across CAD platforms.

3.4.1 Software free implementation

Another way to use a CAD based macro is to implement GDM. This approach has significant advantages in terms of ease of use. It greatly reduces the steps involved in setting up the generative scheme and in navigating the design space directly from a CAD environment that the designer is familiar with. Genoform™[21] a plug-in for SolidWorks™ is an example of such implementation. It is able to operate directly within the CAD environment (Fig.18). The external data table is used here purely for storage of chosen designs. An internal data storage is used to save generated data.

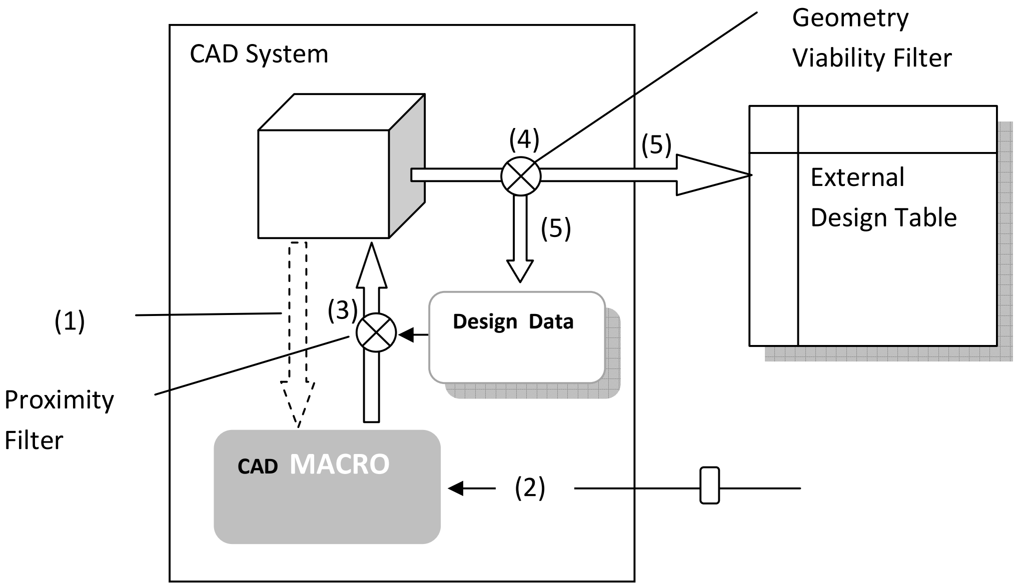


Fig. 7. CAD Macro based Implementation of GDM

CAD macros may be implemented with the following steps.

1) Extract the parameters that are dimensioned (driving parameters) and their corresponding values (initial values) from the CAD file and store these values in an external design table (Fig.8).

2) Set the limits of the exploration envelope initially as a percentage of the initial values using a creativity slider bar (Fig.18) and store these values in the external design table (Table.1). These maximum and minimum values may be modified later if necessary.

3) Generate a design instance within the exploration envelope, store its value in the design table (Fig.8) and instruct the CAD engine to create an instance of the design using the generated values. A software proximity filter may be applied here to asses if the generated parameters are beyond a threshold distance of values generated previously to avoid similar designs being generated. If it is within a certain Euclidean distance the design is re-generated.

4) If the CAD kernel is able to generate the design without major geometric errors (triggering build failure) display the design and store it as a design instance within the CAD system. This is considered a geometric viability filter that is only possible is CAD systems with geometric kernel capable of detecting unviable geometries. If it fails to re-generate the design, it generates a fresh instance of the design (Go to step 3).

5) This process is repeated between 10 or 20 times and the designs may be stored giving the designer an opportunity to store the desired designs in an album. This album may then be recalled to narrow down the selection by comparing the generated designs against each other.

This process is executed entirely from within the CAD environment. At anytime during this process, the designer may alter the design, add or delete new design parameters, modify the exploration envelope or the threshold values that control the diversity between the generated designs. Such an arrangement not only provides complete flexibility but it is almost identical to the normal CAD based design environment used by designers. Some of the unique implemental aspects of this implementation is covered by US Patent 7,552,032 [44].

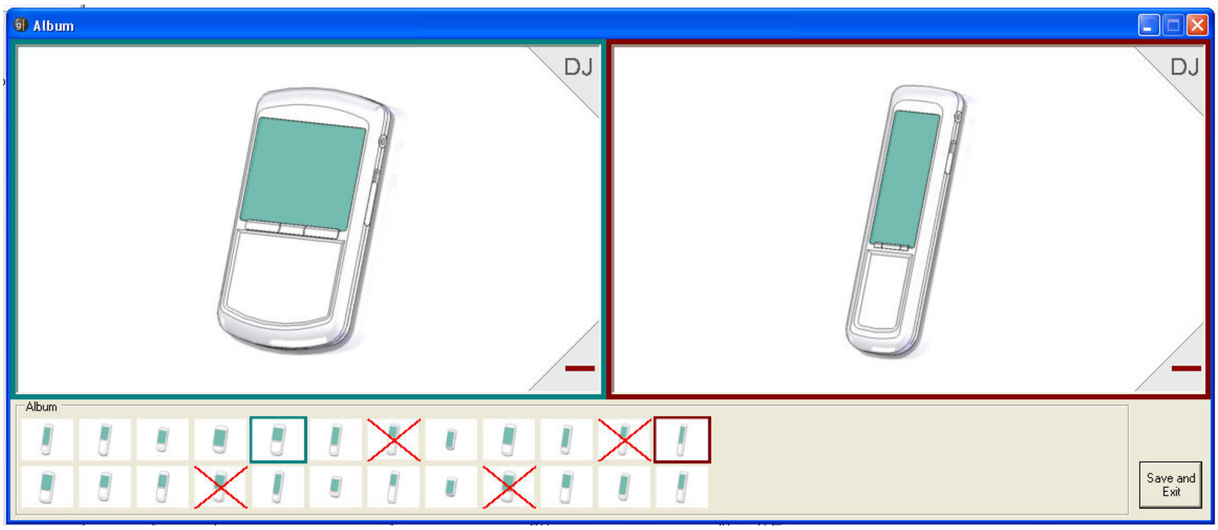


Fig. 8. Selection by comparison

3.5 Computational cost

The design search space in many design problems can be excessively large. In such cases the computational costs becomes an important consideration. The computational time involved for generating designs and filtering through various filters can be estimated as thus. If ***m*** is the average model generation time and ***fn*** is the percentage of designs passing through filter ***n*** and ***ftn*** is the time it takes for the filter to evaluate the solutions. ***ft0*** is the internal filter. If ***g*** is the number of genotypes generated then, the computational time ***Tn*** at the ***n***th filter can be evaluated as:

$$Tn = m \,.\, g \times \left(1 + \sum_{n=0}^{n} ftn \right) \quad (1)$$

This computational time does not take into account other computer processing times such as saving files retrieving files etc. The model rebuilding time depends very much in the use of the geometric kernel. If the model is "*well defined*" ,then the models re-build quickly and if the kernel is employed to solve many equations in order to construct the geometry, then it will be much slower.

In GDM the designer has the choice of setting requirements as built in equations that can be embedded into the generative scheme, or as filters or as part of the evaluation criteria. From a computation point of view, the equation would be the more efficient as it would not waste computational resources required in creating and testing a range of unviable solutions. In essence, equation filters and evaluation criteria are all used for the same purpose – to prune the solution space; but their computational costs differ. Thus, the efficiency of the generative scheme depends on the strategies used in pruning the search space.

Individual filters are used to remove unviable designs based on singular criteria. They are best used to represent evaluation conditions that are not easily analytically derivable. Evaluation is best used to represent multi criteria selection processes. Filters too can be computationally costly; hence it is best to eliminate the unviable designs as early as possible. This also means that less permissive filters should be placed first to reduce the number of designs passed on for further filtering. But this needs to be weighed against the computational time involved in making the pass/fail decision by the filter.

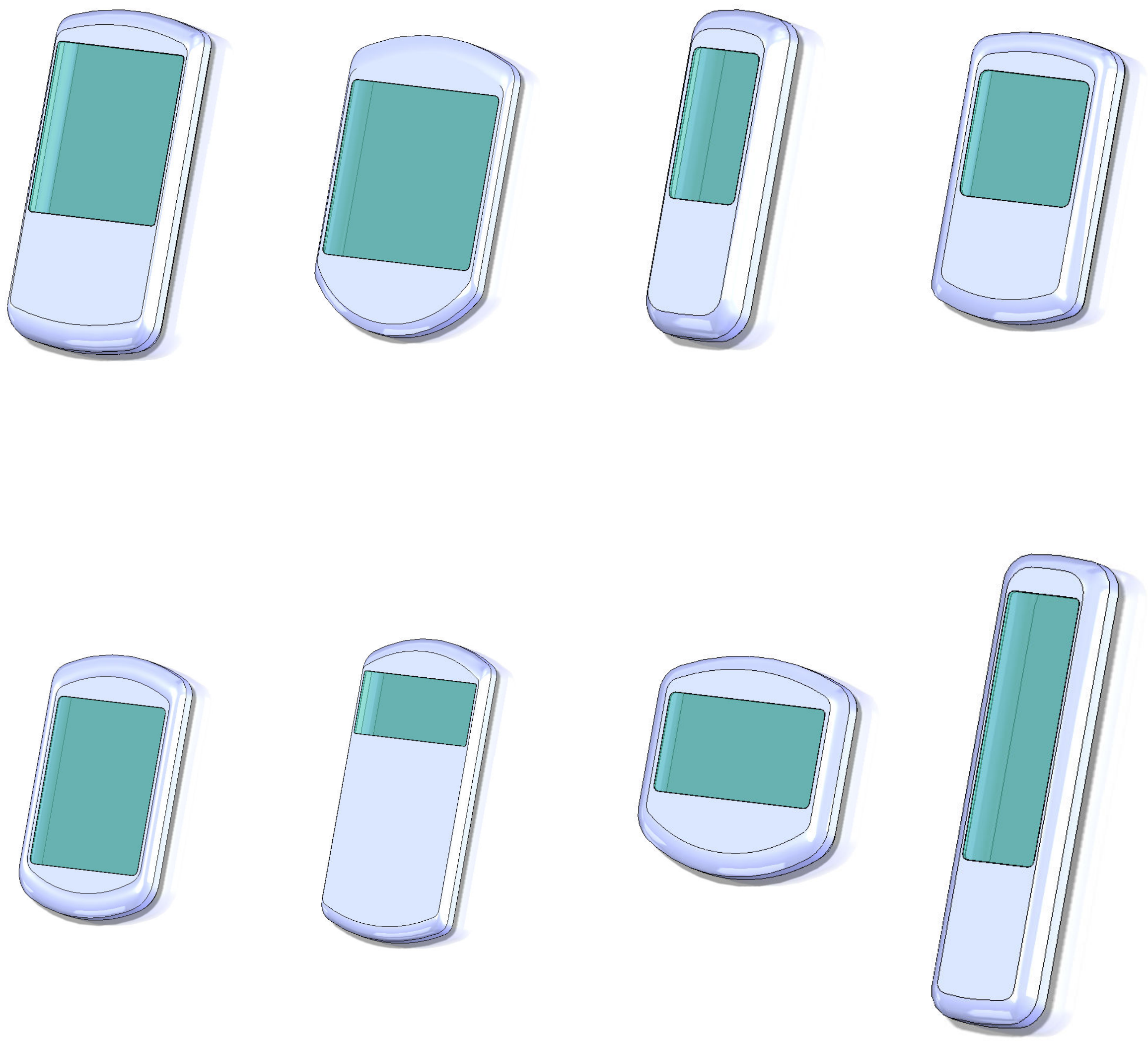
4.0 Design Examples

An MP3 player is designed here using the GDM method primarily to demonstrate the design process, quality of design variations that can be achieved and to illustrate how dimensioning can be used to embed geometric logic behind the design. More examples of GDM generated designs are also shown (Fig. 12). A further example (Fig.19) is provided in the following section for purposes of comparison with genetic design algorithms. It also illustrates the filtering process

4.1 Design development process

Product designers initially work with low levels of complexity mainly to get the form and its feel right. Industrial designers usually make foam models they initially with low levels of detail. The same process can be followed in GDM. The initial model can be purely the form outline which can be generated using

GDM (Fig. 9). Genoform™ was used to generate these designs. The action of the internal proximity filter may be observed from the geometric diversity of the generated designs.



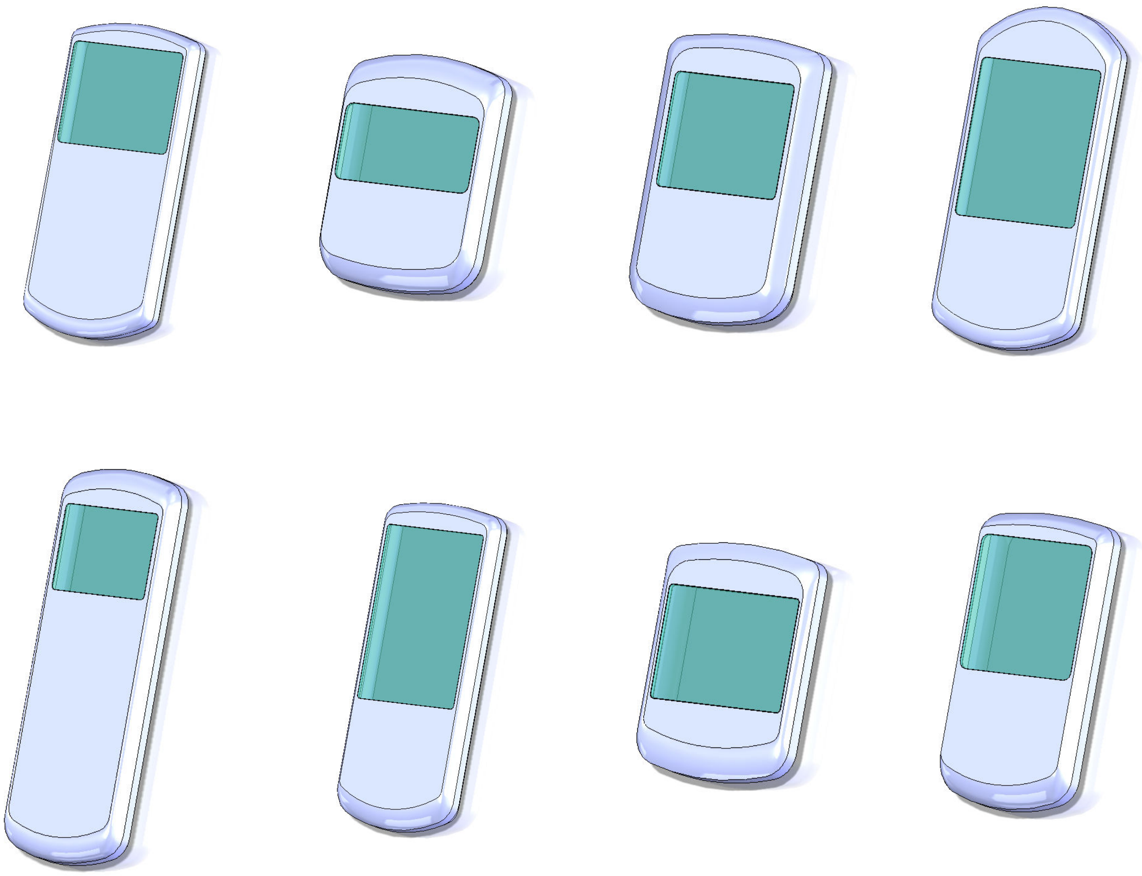
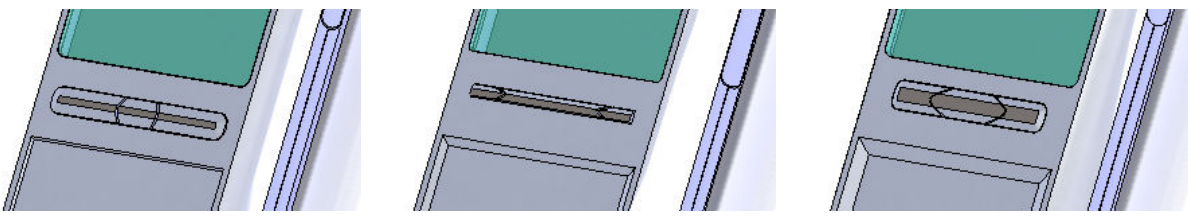


Fig.9. Generated base forms

In most CAD systems, the hide function and history function can be used to control the geometric build. This allows the designers to switch from high detail to low detail models or to any point in the earlier build history and also to switch off selected details in order to focus their attention on a limited aspects of the design.



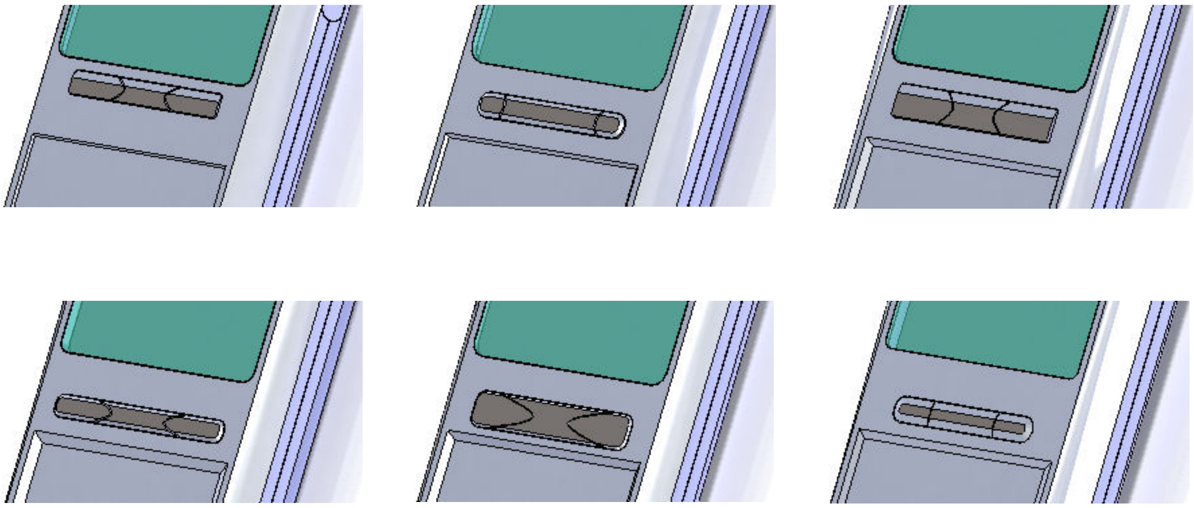


Fig.10. Generation of Design Details

Once a collection of initial base forms are selected, the designer may progressively add details (Fig.10) . The base form may be kept constant if needed. The designer may at any point modify the base form and the switches will modify according to the relationship that have been embedded in the model.

If a certain geometric form is to be maintained its exploration envelope value can be set to a singular value to preserve its current geometry in subsequent rounds of generations.GDM allows designers to explore a wide range of designs as shown (Fig.12). These designs were generated by a trained industrial designer[39] with some experience in generative design. The designer has the freedom to override any of the generated parameters by directly modifying the CAD model. This was done in the illustrated example at points where the designer has a high level of certainty as to the desired nature of the outcome. This approach is normally taken at the fine tuning stage of the design process. It is this capability that makes GDM a '*designerly* ' method; as it allows designers to directly manipulate the CAD model or use its generative capabilities at any point in the design process according to their wish.

4.2 Modeling

Often the dimension of hand held electronic devices are driven by the dimension and form of its internal components. In the design of the MP3 player, we assume that it is driven here by the PCB dimensions, which can rarely be considered the key driving dimensions. The shell is proportioned using clearance dimensions of shell width, Shell H1 & H2. By dimensioning it as such, the geometry of the shell will vary according to the PCB geometry and a certain minimum clearance between the PCB and shell can be ensured. This is a simple example of using the dimensioning capability of CAD systems to maintain the geometric logic of the design during random generation. This is only possible if the CAD system posseses a geometric kernel that can manage the geometric logic of the design. The designer can exploit the geometric intelligence of the kernel to maintain proper relationships between the components purely by dimensioning.

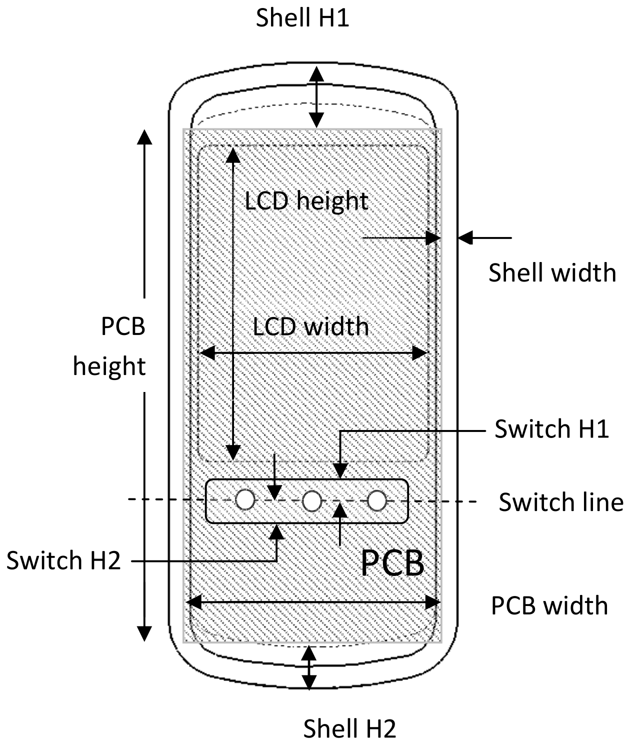


Fig.11. Genetic Model of MP3 Player

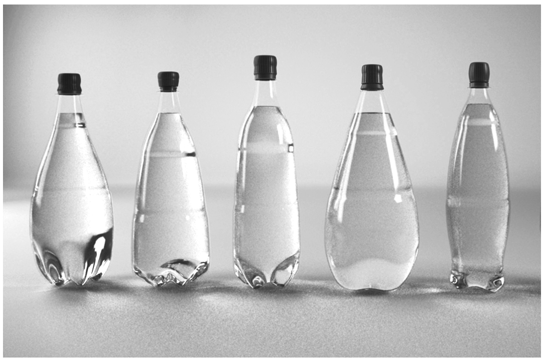
A similar arrangement can be used to ensure that the switch covers are located along the Switch line by dimensioning the Switch H1 & H2 from the Switch line. If the designer wishes to maintain a constant LCD area, the equation facility within the CAD system can be used to drive the other parameters example to ensure the given LCD area is maintained (this is not shown in this example). Similar equations can be used for example in the generation of bottle designs to keep the volume of the bottle constant. Such equations will effectively reduce the degrees of freedom without comprising the 'randomness' as the driven parameters will appear to have a random quality as well.

Fig.12. Generated & Fine tuned MP3 Player designs

5 Comparisons

The lack of clearly formulated generative design methods[23] makes it difficult to compare GDM with other CAD based generative design methods. Though many architectural magazines display parametric design variations using various generative schemes. Details of execution are rarely provided. Though there are some experimental applications in product design, currently there are no known mature examples in product design that we can use for comparison. As GDM is claimed to be a practical designer driven generative design method, it's evaluation needs to be done by designers. It 's efficacy needs to be evaluated and compared to other designer driven methods in terms of the diversity and uniqueness of solutions it can generate and the time and skill levels required for its execution. This is yet to be done. But the author's few years experience with 2nd year Industrial Design and Architecture students is that those proficient with SolidWoks™ within an hour, are able to use generative software to create design variations. Modeling skills however take longer to master. Some examples of student projects are available [45]. Examples shown in (Fig. 11,12) have been created by a trained industrial designer[39].

GDM’s application area (Section 1.1) defined as complex design problems with subjective and non-computable performances criteria makes benchmarking difficult. Examples however are provided (Fig 15.) to demonstrate the diversity of design variations that can be generated with GDM.



Generated bottles

Generated thumb drives

Generated spoons

Generated watches

Fig. 13. Product Designs generated using GDM

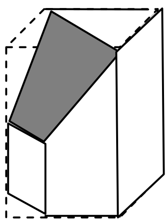
The closest generative method with product design examples was found to be GADES, which is based on Genetic Algorithms. As GA based methods are known for their ability to generate novel designs, we have chosen a particular implementation of it capable of producing 3D designs for the purposes of comparison.

5.1 Coffee Tables generated using GADES

GADES is a genetic algorithm designer developed by Bentley [30] . Using this method he demonstrates the form design of 16 different design examples; tables, set of steps, heat sinks, optical prisms, train fronts, boat bows, boat hulls, saloon cars, sports cars and floor planning for a hospital building. They were evolved over 500 generations with population sizes ranging from 160~200. Out of this, the coffee table was chosen as a range of generated examples is available for comparison.

The designs generated by Bentley require:

1) **Phenotype creation** - using a special partitioning system made of clipped stretched cuboids as shown in (Fig. 14). Every chromosome is arranged in hierarchy containing multiple blocks of nine genes with each block of gene being defined by 16 bits (Fig.15). A new type of cross over method called hierarchical cross over is used to create mutations using points of similarity to ensure that there is no loss of meaning.



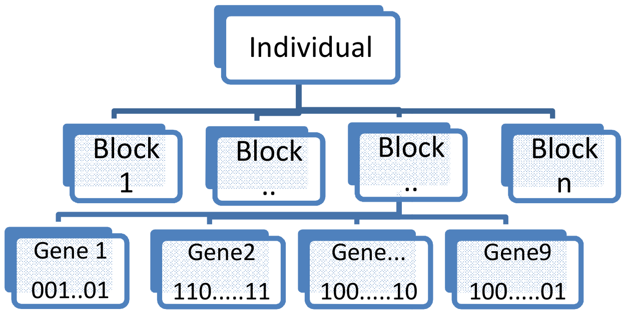


Fig. 14. Sample phenotype

Fig. 15. Hierarchical genotype

2) **Internal & external population** – All new solutions are held in the internal population, which replace the weakest ones in the external population. The internal population is randomly initiated or may be seeded pre-defined components.

3) **Genotype mapping** – external emroyogeny is used to map the genotypes to the phenotypes. Designs constraints such a symmetry are applied at this stage.

4) **Evaluation software** – is used to evaluate fitness values for multiple criteria such as mass, size, flat upper surface, supportiveness and unfragmentedness (to ensue table to floor connectivity).

5) **Multi Objective criteria** – weightage for different criteria's are set manually. Bentley introduced a new method to then scale the fitness values according to the ranges of fitness functions.

6) **Breeding** – the phenotypes with higher fitness values are bred in the internal population using hierarchical cross over and with some mutation.

7) **Iteration** – the process is iterated till acceptable solutions emerge.

Bentley views the evaluation software as the equivalent of designs specifications. He generates 20 coffee table designs in 35 minutes (Slightly over a minute per design). He notes that the designs "exhibit some of the properties of creativity". All the generated designs appear to be viable. He has been able to build one of the generated designs [30] [14].

5.2 Coffee Tables generated using GDM

Designs similar to the coffee tables generated by Bentley[30] were generated using GDM (Fig. 19) . This was done according the process described in section 3. The designs were generated in SolidWorks™ using the Genoform™ plug-in operating on a typical single processor PC (clock speed of 2 Gz). A selection of the generated designs (Fig. 19), the Genetic Model (Fig.16) and the genotype design (Fig.17) are shown.

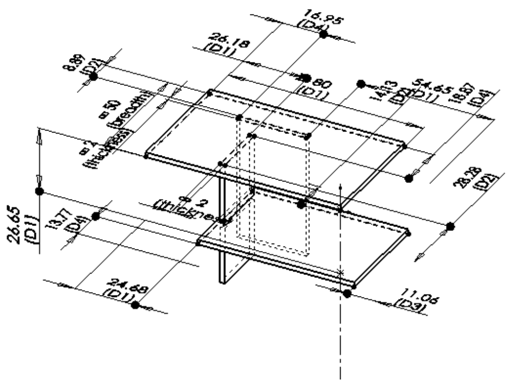


Fig. 16. The genotype (Quarter of the table)

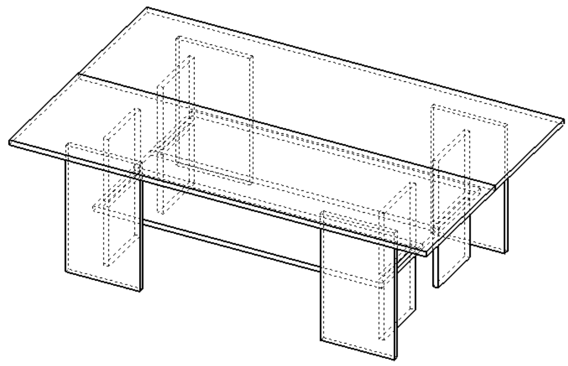


Fig. 17. The genotype CAD model built in SolidWorks™

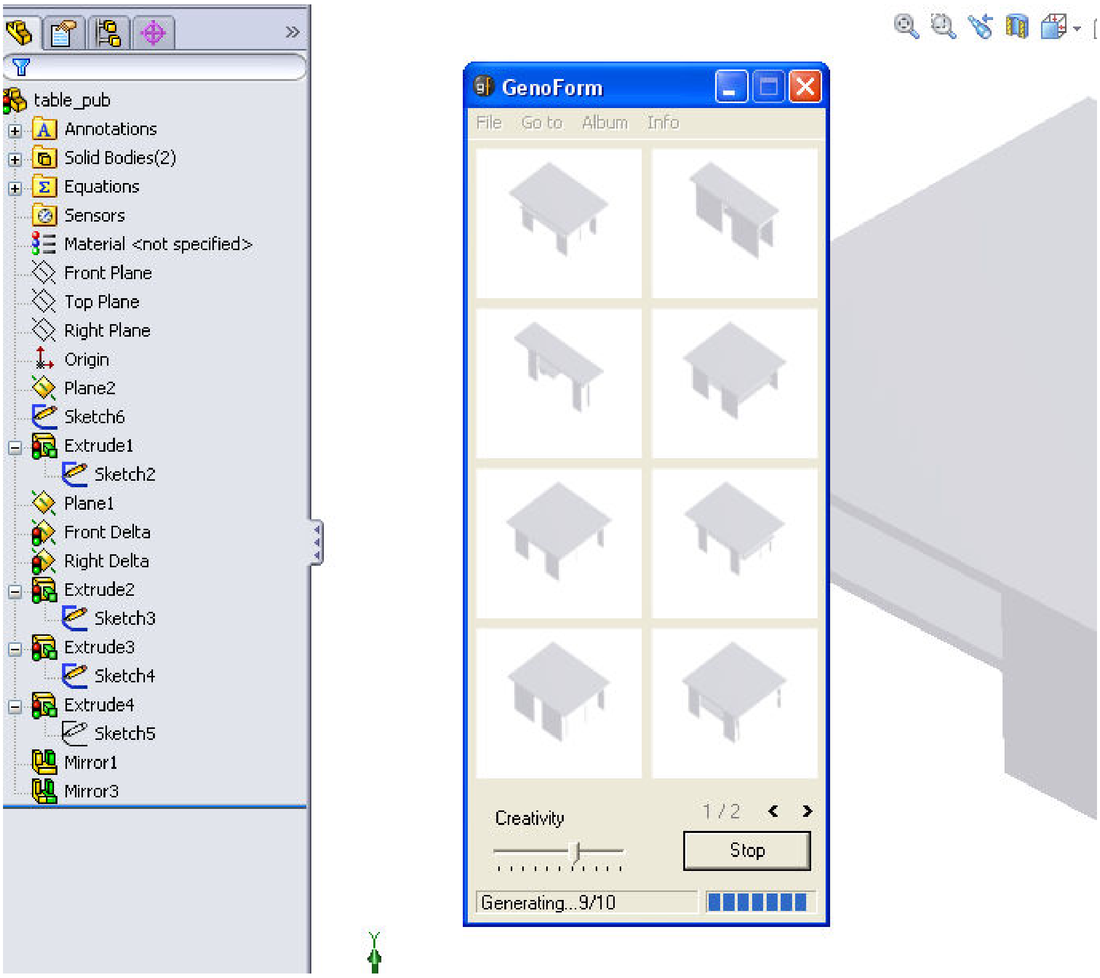


Fig.18. The Genoform™ plug-in generating table designs

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S
1	Design Table for: table																		
2		D1@Sketch6	breadth@Sketch2	D1@Front Delta	D1@Right Delta	D1@Sketch3	D2@Sketch3	D3@Sketch3	D4@Sketch3	D4@Sketch4	D1@Sketch4	D2@Sketch4	D2@Sketch5	D4@Sketch5	D1@Sketch5		Weight Kg	Center of g cm	
3	**Default**	**26.7**	**50.0**	**20.0**	**20.0**	**24.7**	**14.1**	**11.1**	**13.8**	**18.9**	**26.2**	**28.3**	**8.9**	**17.0**	**54.7**				
4																			
5	Max.Val	32.0	60.0	24.0	24.0	29.6	17.0	13.3	16.5	22.6	31.4	33.9	10.7	20.3	65.6		50	30	
6	Min.Val	21.3	40.0	16.0	16.0	19.7	11.3	8.9	11.0	15.1	20.9	22.6	7.1	13.6	43.7		20	10	
7																			
8	Design 1	28	47	21	21	23	16	11	15	16	29	27	10	20	64		45	25	
9	Design 2	24	51	24	22	27	16	11	16	16	28	29	8	19	46		42	32	
10	Design 3	27	58	23	21	28	12	12	12	18	31	33	10	15	56		56	23	
11	Design 4	22	60	18	23	27	13	10	13	16	21	25	10	19	45		39	27	
12	Design 5	31	48	21	21	28	13	11	14	19	31	24	8	18	52		47	22	
13																			
14																			

Table.1. XL Design Table showing generated values

In this example the weight of the table and the center of gravity are used to filter out designs that are too heavy (above 50kg) or too unstable (cg above 30 cm). These values are derived directly from SolidWorks™ and are shown in the Design Table (Table.1). Design 2 and 3 fail to pass through the filter in this example. Genform™'s built in proximity filter was used and the generated designs are displayed (Fig. 19) without further manual selection. Any number of designs can be generated through this process. It took on average less than 6 seconds to generate a single design.

Fig. 19. GDM generated Coffee table designs

5.3 Direct comparison between GADES & GDM

The quality and variability of the generated designs by GDM appear to be similar to those generated through GADES, despite the two approaches being significantly different. GADES requires the setting up of phenotypes, translating them into genotypes, developing and quantifiable evaluation criteria before the generative process can start. In comparison, in the GDM only the genotype needs to be created to commence the generative process. The critical skills for GADES would include the phenotype building, applying appropriate constrains, genotype mapping, and the setting up of evaluation criteria. All this would require different software tools and experiences. In comparison, the GDM scheme is built on top of CAD, requiring only a simple macro to create random numbers between set limits and instigate the CAD to generate instances of designs based on generated values. Further, the application of constrains in GADES is not directly geometrical and requires a deep conceptual understanding of genetic algorithms. The development of the generative schemes is often an interactive cyclic activity necessitating the building and testing various schemes. In GADES, it would require many types of activities where as in GDM it would require mostly the redesign of the genotype. The genotype building time is often a multiple of a single instance model building time, whereas it is likely to be significantly higher in GADES. Setting up of the generative scheme in GADES requires experience in building GA's and some programming. Setting up generative schemes in GDM requires only rudimentary CAD & modeling skills. The author has introduced GDM to over five batches of second year industrial design and

Architecture students (with SolidWorks experience) and has found that they are able to create and generate designs within an hour of introduction.

In comparing the practicality of these processes, it may be concluded that GDM is significantly easier to set up and requires significantly less skills than what is required to setup the equivalent scheme in GADES to create similar results.

5.4 Advantages of GDM

The GDM has many advantages over other code based generative schemes. They are mainly:

1. **Integration with work practices** – GDM works directly on CAD tools and uses CAD methods and commands to execute most of the generation. It imposes very little restrictions on the designer. GDM operates on an independent operational layer that does not interfere with the normal design process. This is an un-encumbered method that designers are more likely to adopt as they need not be totally dependent on it.
2. **Single Platform** – the generation scheme is executed using a single CAD design environment, without the need to translate between different frame-works as in schemes where the phenotypes and genotypes are different.
3. **Transparency** – the generative scheme is based on graphical representations that are relatively easily understood by designers.
4. **Ease of setting up** – the generative scheme is easily setup by building of generic CAD model.
5. **Effective in early stage design** – GDM can work with early stage design models and incomplete evaluation criteria, which is often the case in early stage design.
6. **Embedded Design intent** – the design intent can be embedded directly into the generative model.
7. **Embedding of Design intelligence** – it is possible to embed geometric logic and relationships using CAD's native functions and dimensioning facilities. Formulas and macros can also be included using CAD's native capabilities.
8. **Ease of transition** – it is relatively easy to take the design from early to late stage using the same model.
9. **CAD advantage** – advanced analytical packages now work seamlessly with CAD systems enabling the employment of sophisticated engineering criteria.

In summary, the key advantage of GDM is its ease of use. Designers spend a life time developing their own particular design process and are very resistant to process changes. It is unlikely, except for a small minority, that they will adopt processes that require a completely different approach to design. The skill sets and the programming knowledge required to implement most of the evolutionary algorithms based generative methods are well outside the range of most designers. Most design problems are multi-criteria problems where the evaluation criteria are hard to define. Thougha few multi-criteria optimization methods are available, multi-criteria problems are problematic for evolutionary algorithms that attempt to optimize performance on amalgamated criteria.

The defining of fitness criteria presents an even bigger challenge as complex criteria such as aesthetics remain hard to define. Even though there are some attempts at developing multi criteria evolutionary algorithms and attempts to quantify aesthetic fitness, they are unlikely to lead to methods that designers would accept. Evolutionarily algorithm based methods have a rigid structure that makes it difficult to incorporate into existing work practices.

5.5 Disadvantages of GDM

There are a number of disadvantages in the GDM. They are mainly:

1. **Limitation of search space** – the design space defined by parametric genetic model is limited in size. The use of mutation and cross-overs is likely to increase the size of the search space which could lead to more creative solutions.
2. **In-exhaustive search** – the generation scheme explores only limited regions of the solution space as directed by the designer.
3. **Evaluation fatigue** – GDM places, despite best efforts to limit the number of designs presented for evaluation, places significant evaluation burden on the designer.
4. **Genetic Modeling** – creating a high quality genetic model is not a straight forward process. It requires an iterative design and test process. It requires significant experience to develop expressive genetic models. Best practice for creating genetic models is not known.
5. **Enumeration of search envelope** – the search envelope has to be explicitly defined. This will require additional effort.
6. **Designer's prejudice** – will limit the process from exploring design possibilities that lay outside the designer's subjective hunch. Designs could be under explored and over guided.
7. **Optimization** – Genetic algorithm based methods are likely to outperform GDM in evaluatable single criteria design problems dueto its ability to automatically carry out fine grained exploration of the solution space.
8. **Accumulation of positive attributes** – Cross-overs tend to accumulate positive attributes. exponentially. The lack of this effect is a serious disadvantage.
9. **CAD Kernel dependence** – the genetic models cannot be transferred across CAD platforms, as the generative scheme is dependent on how the CAD's particular kernel interprets the design.

In summary, the GDM has many disadvantages. Genetic modeling also forces designers to parameterize early stage design which designers resist as they are used to designing fluidly without any encumbrances in the early stages of design development. The explicit definition of the constraint envelopes require significant time commitment that designers may not be willing to commit, especially in the early stages of the design process. GDM forces the designer to enumerate the design search space with the promise of more exciting design possibilities; it is too early to judge if this promise will be realized across a range of design problems.

However, it may be safe to state that in its current state of development, evolutionally algorithm based methods are unsuitable for design problems involving multiple complex criteria and difficult to integrate with current design practices. They are also unviable for early stage design development where the evaluation criteria and the design are still under development.

6 Conclusions

In this paper, we introduce a method of using CAD systems to help designers explore and develop design possibilities from early to detailed stages of design. We have demonstrated that it can implemented on top of CAD systems with minimal overheads fully exploiting the native capabilities of modern CAD systems; making it, a practical CAD based Generative Design Method. In the context of numerous failed attempts to impose structured and process for conceptual design, we have identified (Section 1.4.1) the key requirements that need to be met, for CAD based design methods to be useful in the conceptual stages of design. It has been demonstrated through examples that GDM meets most of these requirements.

The GDM method is shown to be entirely designer driven; free of restrictive and invasive frame works allowing designers to navigate the search space in the way they see fit. It's abilities to support workflows that are essential for creative design exploration have been demonstrated through examples. It's ability to work satisfactorily, with limited information in the early stages of the design is illustrated (Fig.6). As GDM is entirely CAD based the ease of transition from conceptual designs into the detailed design is seamless, (from Fig. 6 to Fig. 8) has been illustrated. It has been shown that the selected designs can be further improved or modified manually by the designer using the same phenotype model (Fig.7). Ways of subjecting chosen geometric entities to variations at any stage of the design development has been demonstrated. In short, GDM has been shown to be a 'designerly' method that is able to work harmoniously with designers' work processes in the design of products of medium complexity.

It has been shown that GDM can be implemented with history based parametric CAD systems with and without geometric kernels; with the use of simple design tables and macros. In CAD systems where geometric kernels are present, it has been shown that complex geometric relationships can be developed purely by using its native dimensioning system. In comparison with the shape grammar approach, it does not require the setting up of external rules (which require special expertise) and additional grammar interpreters to interact with CAD systems. It has been demonstrated that these geometric kernels are able to maintain the geometric logic (or the grammar) of the genotype through the generative process where the geometry is subject to significant change; demonstrating that the original intent of shape grammar can now be accomplished by most CAD systems with geometric kernels that facilitate dimensioning and the setting up of relationships.

It has been shown that it is possible to preselect designs that are significantly different to each other, by ensuring that the phenotype values are significantly different to each other. It has been also shown that filters can be used to eliminate designs with unacceptable performance, thus helping to reduce the selection load on the designer.

The claim of a 'practical' CAD based generative design system is made based on the simplicity and elegance in implementation and ease of use. The need for simplicity and transparency is considered to be of paramount importance if this method is to be adopted by industry. The GDM method makes the best of use of the ever increasing abilities of CAD systems to achieve this end.

In comparison with more established methods based on genetic algorithms namely GADES, it has been shown that comparable results can be achieved with significantly less effort; without the need for programming and the use of abstract frameworks, using transparent methods that are based on CAD. In conclusion, it has been shown that it is now possible to transform CAD tools into design exploration tools using methods that are transparent and relatively simple, in a way that is useful to designers.

6.1 Other applications

Central to GDM is the creation of genetic models and enumeration of the constraints that limit its creative freedom within realizable , feasible and desirable limits. Such genetic models combined with the constraint envelopes could have wide ranging applications outside generative design, particularly in applicators where design freedom is to be maintained within desirable limits. Mass customization, computer games and co-creation present such opportunities. The creative freedom offered to non-designers in these applications need to be curtailed within certain limits. In such applications, random generation can be replaced by user driven values which can be bound within acceptable limits. The ability to embed design intelligences and constraints within genetic models will enable non-designers (or consumers) to participate in the design of complex artifacts. Genetic models can also be used for parametric mapping of consumer preferences, as it presents formal ways of mapping geometric quality of product variants. This could provide companies with valuables information on consumer preferences.

Image based object recognition may also benefit from generative models, as they rely on internal representation of objects for the purposes of matching them with images of real objects. By replacing the parametric models that are stored internally with more expressive genetic models it would be possible to make more accurate matches. Such an approach can perhaps be used for creating 3D buildings from aerial photographs. Genetic models may also be seen as procedural models. Procedural representations are known to drastically reduce transmission bandwidths in online applications as geometries are reconstructed by client machines using inelegantly structured, highly compressed data. By using GDM, unlimited variations of great diversity can be created and deployed efficiently for online games, digital films and virtual environments. In addition, it will enable the participants to create equally rich and diverse content.

6.2 Wider implications of genetic modeling

Genetic modeling enforces a certain structure to design that transcends geometry. It provides a structure that embeds constructional history and captures the essential attributes of the design in an interpretable and classifiable format. Before the advent of genetics, biologist dealt with plants and animals based mainly on their external construction. The understanding of genetics and evolution helped understand the structure of life and its designs in a way that revolutionized biology. Genetics made it possible to interpret, classify and modify the design of life forms.

Language too went through a similar transformation. The field of design unfortunately lacks interpretable, classifiable and modifiable structure. The science of linguistics made it possible to extract meaning of out sentences and made written language computationally interpretable. In stark comparison, the structure behind design is currently poorly understood and yet to be developed.

Genetic modeling may one day, present an interpretable, modifiable and classifiable structure for design. The powerful search technologies that we enjoy today are the result of the understanding of the structure of languages. If such understanding of structure is developed for design it would facilitate search. In the context of genetically structured design, search is design.

Currently, design information is mostly captured in CAD ias final geometric form, which allows for geometry based searches. Such searches are of limited value. In architecture, Building Information Management systems (BIM) now make it possible to capture significant about of information about buildings. However, design currently lacks formal CAD independent data structures that are capable of capturing its complexity. If design information is captured in an interpretable, classifiable and modifiable form, then it would be possible as biologist do; to understand the evolutional heritage, constructional logic and the external conditions that shaped the form of all designed artifacts. Its implications will be far reaching; as it would connect design, manufacture and consumption in a way that was not possible before.

6.3 Future research

The proposed GDM method is at best, rudimentary in its current state. Its components need further development. Ways of structuring genetic models need to be further developed. Ways of developing genetic models that can create wider topological variations need to be explored. Ways of identifying geometric and visual differences between generated models will help in identifying designs that are distinctively different. This will help reduce the number of designs presented for the designer's selection without compromising the quality of search . Some of the CAD based shape clustering methods [46] may be used towards this end. Ways of defining kernel independent genetic models will enable the sharing of such models across CAD platforms and will help in the rapid growth of modeling knowledge. Ways of embedding design knowledge into models and its effect on the search space need to be better understood.

Methods of creating genetic models need to be developed further over a wider range of design problems. Ways of growing the complexity of the genetic models need to be explored perhaps based on *Embryogenesis* [14]. So far, the designs were generated as part files. Combinatorial methods are yet to be explored at assembly level. With its adaptation, GDM is bound to alter the way designers explore design possibilities. Its use by designers needs to be observed, as it will help us in developing it abilities further.

Though GDM is developed here as a linear design exploration process, it would be possible to implement it as a parallel design process as in evolutionarily algorithms, where populations of designs are maintained for breeding and selection. Ways of combing the best attributes of GDM and Genetic Algorithms need to be explored further. One clear possibility is to allow the designer to choose the candidate designs for breeding and selecting the more desirable outcomes for further breeding. The designer's involvement could open up a new and potent possibility – the breeding of specific geometric features. Very often, in generated designs only some attributes of the design will be remarkable and desirable. Current GA based breeding methods shuffle the entire genotype during cross-over. The designers' involvement will make it possible to pick particular desirable features of the design for further breeding, making it possible to rapidly build populations with desirable features.

In its current form GDM makes it possible to manage the creative scope of design exploration by the enlargement and contraction of the generative space in terms of its parametric attributes. The management of creative scope in design exploration has interesting implications. Some designs generated by GDM [47] seem to have creativity potential. In some cases, unanticipated solutions have been observed to emerge. The ability to structure design information in a genetic format may provide new ways of differentiating designs, resulting in better understanding of creativity. Perhaps, creativity can be defined in terms of distances between known designs and new designs in performance space.

Acknowledgements

I would like to thank Prof. Mark Gross for his critical comments; Mr. Marco Vanucci and Prof. Marc Aurel Schnabel for helping me improve the paper.

Role of the funding source

This research was supported by a research grant - Parametric Form Design RP-295-000-044-112; from the Department of Architecture of the National University of Singapore (NUS).

The Genoform software was developed by Genometri Pte Ltd.

References

1. Norton, S. *Construction & Generative Design, CAD details*. 2010; Jan 2010:[Available from: http://www.caddetails.com/2004/main.asp?article=January2010.htm.
2. Shea, K., R. Aish, and M. Gourtovaia, *Towards integrated perforamce-driven genertive design tools.* Automation in Construction, 2005. **14**.
3. Horva´th, I. *Conceptual design—inside and outside, in:*. in *EDIProd, 2nd Seminar and Workshop*. 2000. Zielone Gora, Poland.
4. Horva´th, I., *On some Crucial Issues of Computer Support of Conceptual Design*, in *Prodcut Eningeering: eco-desing, technologies and green energy*, D. Talaba and T. Roche, Editors. 2005, Springer Netherlands.
5. Guidon, R., *Designing the Design Process: Exploiting Opportunistic Thoughts.* Human Computer Interaction, 1990. **5**: p. 305-344.
6. Goldschmidt, G., *The Dialectics of Sketching.* Creative Research Journal, 1991. **1**(2): p. 123-143.
7. Maher, M.L. and J. Poon, *Modeling Design Exploration as Co-Evolution.* Microcomputers in Civil Engineering 1996. **11**(3): p. 195-209.
8. Dorest, K. and N. Cross, *Creativity in the design process: co-evolution of problem-solution.* Design Studies, 2001. **22**(5): p. 425-437.
9. Prats, M.e., *Transforming shape in design: observations from studies of sketching.* Design Studies, 2009. **30**(5): p. 503-520.
10. Oxman, R., *The thinking eye: visual re-cognition in design emergence.* Design Studies, 2002. **23**(2): p. 13-164.
11. Pratt, M.J., B.D. Anderson, and T. Tanger, *Towards the standerdized exchange of parameterized feature-based CAD models.* Compute-Aided Design, 2005. **37**: p. 1251.
12. Menezes, A. and B. Lawson, *How designers perceive sketches.* Design Studies 2006. **27**(5): p. 571-585.
13. Jun, H. and J.S. Gero, *Emergence of shape semantics of architectural shapes.* Environment and Planning B: Planning and Design, 1998. **25**: p. 577-600.
14. Bentley, P.J., *An introduction to evolutionary design by computers*, in *Evolutionary Design by Computers*, P.J. Bentley, Editor. 1999, Morgan Kaufmann San Francisco, CA. p. 1-73.
15. Renner, G. and A. Ekart, *Genetic algorithms in computer aided design.* International Journal of Pattern Recognition and Artificial Intelligence (IJPRAI) 2008. **35**(8).
16. Buelow, P.V., *Using evolutionary algorithms to aid designers of architectural structures*, in *Creative Evolutionary Systems*, P.J. Bentley, Editor. 2002, Morgan Kaufmann San Francisco, CA. p. 315-336.
17. Frazer, J.H., *Creative design and the generative evolutionary paradigm*, in *Creative evolutionary systems*, D.a.B. Corne, Peter, (eds.) Editor. 2002, Morgan Kaufmann Publishers Inc. (Elsevier): California, San Francisco. p. 253-274.
18. Caldas, L. and J. Duarte. *Implimnetatinal Issues in Generative Design Systems*. in *First International Conference on Design Computing and Cognition*. 2004. Boston.
19. Systems, B. *Generative Compoents*. 2010; Available from: http://www.bentley.com/en-US/Promo/Generative+Design/.
20. Grasshopper. *McNeel Corp.* 2010; Available from: http://www.grasshopper3d.com/.
21. Genometri, *Genoform*. 2002, Genometri Pte Ltd.: Singapore.
22. Cloud, P. *Para Cloud GEM*. 2010; Available from: http://www.paracloud.com/.
23. Chase S, C., *Generative design tools for novice designers: Issues for selection* Automation in Construction, 2005. **14**(6): p. 689.

24. Oxman, E.R., *Theory and design in the first digital age.* Design Studies, 2006. **27**(3): p. 229.
25. Marsh, A., *Generative and Performative Design: A Challenging New Role for Modern Architects*, in *The Oxford Conference 2008,*. 2008, WIT Press: Oxford.
26. Bohnacker, H.e.a., *Generative Gestaltung*. 2009: Verlag Hermann Schmidt Mainz
27. Reas, C. and B. Fry. *Processing*. 2010 June 2010]; Available from: http://processing.org/.
28. Guoyan, Y., W. Xianozhen, and L. Peng, *A constraint based evolutionary decision support system for product design*, in *Chinese Control and Decision Conference*. 2009, IEEE Industrial Electronics (IE) Chapter, Singapore Guilin , China. p. 2585-2590.
29. Caladas, L., *GENE_ARCH: An evolution-based generative design system for sustainable architecture.* Lecture Notes in Computer Science, 2006. **4200**: p. 109.
30. Bentley, P.J., *From coffee tables to hospitals: generic evolutionary design*, in *Evolutionary Design by Computers*, P.J. Bentley, Editor. 1999, Morgan Kaufmann San Francisco, CA. p. 405-423.
31. Jingyuan Huang, A.P., Cherry Zhang, Stephen Mann, and M.H. Elodie Fourquet, Kate Kinnear, Michael Lam, and William Cowan, *An Evaluation of Shape/Split Grammars for Architecture*, in *Technical Report CS-2009-23*. 2009, David R. Cheriton School of Computer Science, University of Waterloo.
32. Deak, P., C. Reed, and G. Rowe, *CAD Grammars: Extending shape and graph grammars for spatial design modeling.* Computer Aided Methods in Optimal Desing and Operations 2006: p. 119-128.
33. Hornby G, S. and B. Pollack J. *The Advantages of Generative Grammatical Encodings for Physical Design.* . in *In: Congress on Evolutionary Computation. 2001.* . 2001. Seoul: IEEE.
34. Gu, Z., M.X. Tang, and J.H. Frazer, *Capturing aesthetic intention during interactive evolution* Computer-Aided Design, 2005. **38**(3): p. 224.
35. Weinstock, M., *Morphogenesis and the Mathematics of Emergence* in *Emergence: morphogenetic design strategies*, M.e.a. Hensel, Editor. 2004, Architectural Design: Wiley-Academy: London.
36. Bentley, P.J.e.a. *New trends in evolutionary computation*. in *IEEE Conference on Evolutionary Computation, ICEC*. 2001. Soul.
37. Solano, L. and P. Brunet, *Constructive constraint-based model for parametric CAD systems* Computer-Aided Design, 2003. **26**(8).
38. Mun, D.e.a., *A set of standard modeling commands for the history-based parametric approach.* Computer Aided Design, 2003. **35**: p. 9.
39. Thian, L., *Generated MP3 Players*. 2005, Genometri Pte. Ltd.: Singapore.
40. Janssen, P., *A generative evolutionary design method.* Digital Creativity, 2006. **17**(1): p. 49-63.
41. Rosenman, M.A. and J.S. Gero, *Creativity in design using a design prototype approach*, in *Modeling creativity and knowledge-based creative design* J.S. Gero and M.L. Maher, Editors. 1993, Lawrence Erlbaum Associates: Hillsdale, NJ. p. 139-176.
42. SolidWorks. *Dassault Systèmes SolidWorks Corp.* 2010; Available from: http://www.solidworks.com/.
43. Open. *Open Generative Design*. 2010; Available from: www.opengenerativedesign.com.
44. Krishnapillai, A., *Method and system for automated design*, in *United States Patent and Trademark Office*, U.S.P.a.T. Office, Editor. 2009, National University of Singapore: USA.
45. NUS, S.P. 2005; Available from: http://www.flickr.com/photos/genoform/sets/72157622643020121/.
46. Jayanti, S., Y. Kalyanaraman, and K. Ramani, *Shape-based clustering for 3D CAD objects: A comparative study of effectiveness.* Computer-Aided Design, 2009. **41**(12): p. 999.
47. Krishnapillai, A. *Genometry: a genetically inspired parametric form generation method*. in *First International Conference on Design Computing and Cognition*. 2004. Boston.