# *Thinking parametric design: introducing parametric Gaudi*

*Carlos Roberto Barrios Hernandez*, Department of Architecture, Massachusetts Institute of Technology, 77 Massachusetts Avenue, Room 10-491M, Cambridge, MA 02139, USA

*This paper presents an innovative methodology for parametric design called* Design Procedures *(DP) and shows how it is applied to the columns of the Expiatory Temple of the Sagrada Familia. Design Procedures are actions that generate parametric models where geometrical components are consider as variables. A brief introduction on parametric design is followed by illustrated explanations of the traditional forms of parametric models. Design Procedures is presented as an alternative to overcome the topological and geometrical limitations of traditional parametric models. The DP is able to generate all original designs by Gaudi plus an infinite number of new designs.*


*Keywords: case study, computational model(s), design model(s), parametric modeling, parametric design*

With the increasing demand of flexible tools for Computer Aided Design (CAD), Parametric Modeling is becoming a mainstream of *Computer Aided Architectural Design* (CAAD) software, in order to make variations in the design process less difficult. This is traditionally called *Parametric Design*. Until recently, parametric design was understood as highly sophisticated and expensive software made exclusively for manufacturing in aerospace, shipping and automobile industries. However, designer's demands for flexibility to make changes without deleting or redrawing in a computer has pushed the incorporation of parametric modeling as standard tools in traditional CAD programs (Barrios, 2004).

Variations in design are a fundamental part of the design process in the search for solutions to design problems. Design variations support improvement of design which in turn improves the quality of designed artifacts. Designers constantly go back and forth between different alternatives in the universe of possible solutions, working in a particular

**Corresponding author:**
C. R. Barrios Hernandez
cabeto@mit.edu

www.elsevier.com/locate/destud
0142-694X $ - see front matter *Design Studies* **27** (2006) 309–324
doi:10.1016/j.destud.2005.11.006

part at a given time, or looking back at the whole from a broader perspective. This is a continuous and iterative search process of variations of a design idea, and it is very likely to revisit a previously abandoned solution to rework it. As a result, designers demand flexible tools that allow variations in the design process until a solution is established for further development.

In this context, this paper presents Design Procedures (DP) as a methodology that enhances the design capability of a parametric model to perform design variations, by using shapes as parameters and thinking of parametric design as a general procedure. Consequently, a parametric model becomes a flexible tool allowing changes at the topological and geometrical levels. The paper starts by presenting definitions of parametric design and parametric modeling, followed by a brief overview of traditional parametric models accompanied by examples. Next DP is defined as a systematic methodology to overcome the limitations of traditional parametric models, followed by a case study on its application to the columns of the Expiatory Temple of the Sagrada Familia. Examples of the original column designs accompanied by new generated designs are shown to prove the strength of the DP. The DP is able to generate all the original designs of the columns plus an infinite number of new designs all from a single parametric model.

## 1 Parametric design

*Parametric Design* is the process of designing in environment where design variations are effortless, thus replacing singularity with multiplicity in the design process. Parametric design is done with the aid of *Parametric Models*. A parametric model is a computer representation of a design constructed with geometrical entities that have attributes (properties) that are fixed and others that can vary. The variable attributes are also called *parameters* and the fixed attributes are said to be *constrained*. The designer changes the parameters in the parametric model to search for different alternative solutions to the problem at hand. The parametric model responds to the changes by adapting or reconfiguring to the new values of the parameters without erasing or redrawing.

In parametric design, designers use declared parameters to define a form. This requires rigorous thinking in order to build a sophisticated geometrical structure embedded in a complex model that is flexible enough for doing variations. Therefore, the designer must anticipate which kinds of variations he wants to explore in order to determine the kinds of transformations the parametric model should do. This is a very difficult task due of the unpredictable nature of the design process.

Parametric design has historically evolved from simple models generated from computer scripts that generate design variations (Monedero, 2000) every time the script is run with different parametric values, to highly developed structures based on parent–child relations and hierarchical dependencies. Currently, parametric CAD software offers sophisticated three-dimensional interactive interfaces that can perform variations in real time, allowing the designer to have more control and immediate feedback when a parameter is changed. Computer implementations of parametric models include structures that show the historical evolution of the model, allowing the designer to go back to a previous stage of the design and apply changes. These changes will be propagated through a chain of dependencies of the modified parameters, which means that a designer can go to any stage, change the value of the parameters, and reconstruct the model. A parametric model will either propagate the changes through the structure and reconfigure the model to the new values, or inform the designer if the modified parameters will create any problems in the solution. More sophisticated parametric modeling software has integrated knowledge-based systems, thus offering better inference to the designer about the consequences of the parametric changes the designer does. Knowledge-based systems in conjunction with parametric modeling are under development and depend on a powerful computational structure based on artificial intelligence, but perhaps are the next big step in the new generation of expert CAD systems.

Regardless of the implementation and sophistication, all parametric models can be categorized into two kinds: those that perform variations and those that generate new designs by combination of parameterized geometrical entities. A parametric model can also be a combination of both kinds, although it is very unusual due to the complexity of the model and the computer performance required.

## 1.1 Models for parametric variations

*Parametric Variations* (PV), also known as *variational geometry* or *constrained-based models*, is a kind of parametric model based on the declarative nature of the parameters to construct shapes. The designer creates a geometrical model of any kind, and its attributes are parameterized based on the desired behavior, thus creating a *parameterized modeling schema* (Figure 1A). A parametric modeling schema shows which attributes of a geometrical model are parameterized and how the designer can change the values of the parameters. The idea behind a PV model is that the geometrical components are controlled by means of changing the values of the parameters or

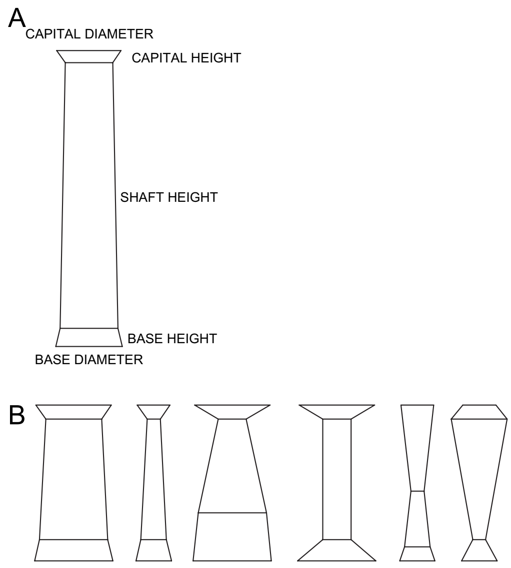


*Figure 1 (A) Parametric modeling schema of a column describing the parameterized attributes. (B) Family of designs showing six instances based on Parametric Variations (PV)*

constraints without changing the topology (number of components and their relations). The parametric modeling schema is the starting point for parametric variations of the designs. Every time the designer changes a parameter a *design instance* is created. The collection of design instances generates a *family* of designs as a result of the changes done to the parametric model (Figure 1B). The most important quality of a PV is that the model allows transformations of the geometry without erasing and redrawing, in a closed contained system.

In a PV model, the geometry is subject to more than one parameterization schema, thus creating more than one way to generate design instances. Figure 2 shows two parameterization schemata of a rectangular shape. In the first parametric schema the rectangle is parameterized by the length and height attributes. In the second parametric schema the

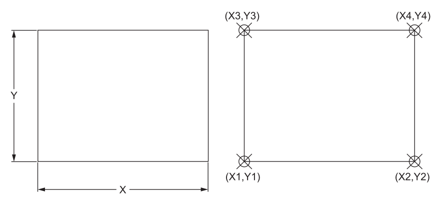


*Figure 2 Parameterization schemata of two rectangular shapes. The first schema shows a rectangular shape with* length *and* width *as parameterized attributes. The second schema shows the same rectangular shape with vertices as parameterized attributes*

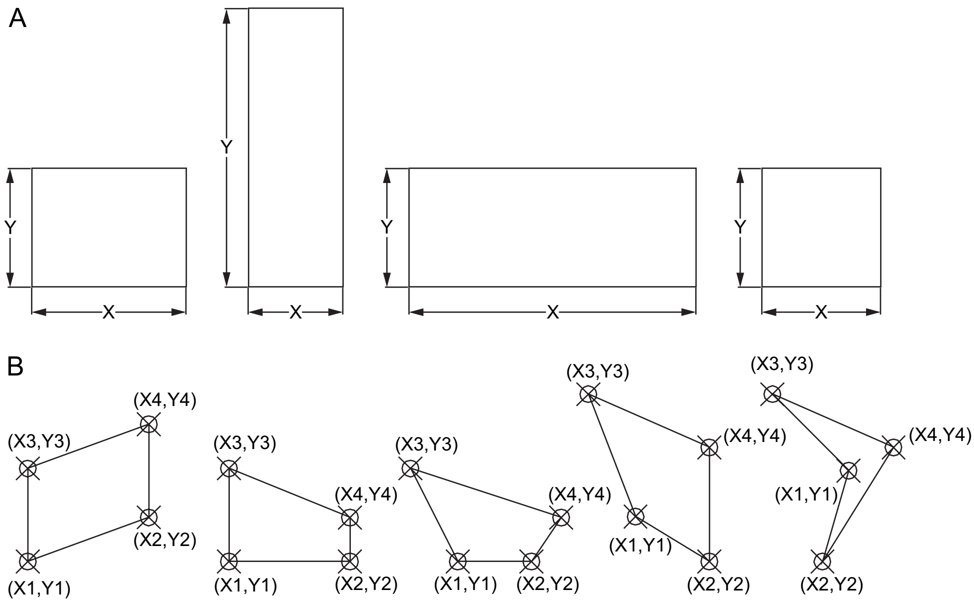


*Figure 3 (A) Family of designs produced by parametric variations the first PV model. The variations occur when the variables "X" and "Y" have different values, corresponding length and width of the rectangle. (B) Family of designs produced by parametric variations the second PV model. The variations occur when the values of the coordinates of the vertices are changed*

same rectangular shape is parameterized by the coordinate values of the vertices. Figure 3A and B shows the corresponding family of designs generated from the two parametric schemata indicating the different possibilities that each PV model can generate.

## 1.2 *Models for parametric combinations*

Parametric Combinations (PC) is the second class of parametric models that is most used. A PC model is composed of a series of geometrical shapes that are arranged according to rules that create more complex structures. Also known as *associative geometry models*, or *relational models*, PC offers another degree of complexity beyond the parameterization of the geometrical components, which is done by constructing combinations according to specific rules. In PC models, the important aspect is the spatial relations and rules of combination between the primitive components, which determines different design compositions. By combining components in different ways a variety of design solutions are achieved. In Figure 4A, a column design is divided into three components: base, shaft and capital; and different designs for each of the components are present. A column design is the result of the combination of the three elements according to the rules. Figure 4B shows a family of designs from the PC model.

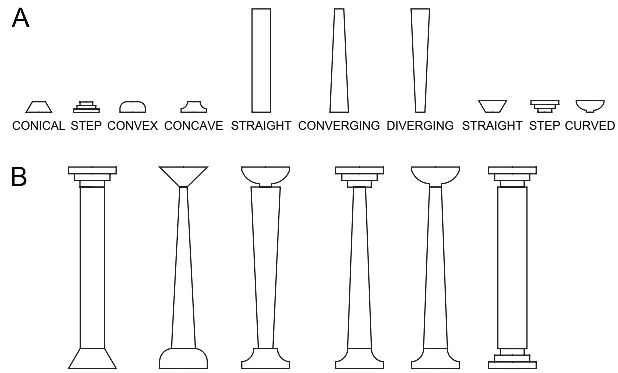


*Figure 4 (A) Components of the PC model: Base, Shaft and Capital. (B) Family of column designs made from PC model*

## 1.3 Parametric hybrid models

Although Parametric Hybrid (PH) models are less used than Parametric Variations or Parametric Combinations, they offer the best of both and can be very robust for design exploration. However, they are very difficult to construct and require a strong data structure for the design vocabulary. In most cases it is better to construct and design with two models in parallel, one for variations and another for combinations. Figure 5 shows a family of column designs from a hybrid model where the components of the PC have been parameterized like a PV and combined to generate new designs.

# 2 Design procedures: a new approach to parametric design

In computation, Procedures are defined as a set of finite instructions that performs a specific task. Also known as a subroutine or function, a procedure takes some parameters as inputs and computes them to produce an answer or answers as outputs. Procedures are small parts of larger computer program intended to achieve partial results.

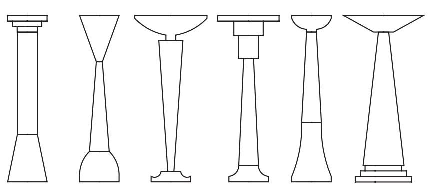

*Figure 5 Family of column designs made from a Hybrid Parametric Model. The components of the column are parameterized and combined according to the composition rules. Hybrid models offer parametric variations and combinations in a single structure*

A procedure is characterized for encapsulating pieces of knowledge in small manageable modules that in some cases can be used as primitives for other procedures. In computation this practice is known as encapsulation.

## 2.1 Design procedures

A *Design Procedure* is as a set of instructions that performs actions that generate parameterized geometrical models. Unlike traditional parametric models, where geometrical components are varied, a design procedure constructs a parametric model which can then be used to generate instances of designs, therefore changes and transformations of both topology and geometry are possible. The design procedure carries instructions in a systematic order, where geometrical components are constructed and parameterized at the same time. For example, a line can be the result of a point moving in a certain direction (point-direction procedure). The location of the point in space, the direction of the line and its length are the parameters of the line constructed by the procedure, therefore the origin, direction and length of the line can be altered after the line is constructed. Other examples of a line procedure can also be the result of the intersection of two planes (intersection procedure): the shortest distance between two points in space (two point procedure), the edge of a polygon (edge procedure), or any other kind of operation that takes any input and generates a line as a result. Consequently, a line is not an explicit representation of itself, but a parameterized geometrical component that depends on the procedure that generated the line. The design procedure creates a parent–child type dependency relation just like in any parametric model, where the parent is the input, and the resulting geometry is the child. The two point procedure for creating a line will have the end points of the line as parameters; while each of the points is a parametric entity on itself, therefore a design procedure results in a parametric model where input shapes can be parameterized entities creating a special kind of encapsulation.

A cube can be modeled as the result of the following procedure: *a square shape which is translated along an axis* (extrusion procedure) by a distance equal to the length of the side of the square. This procedure will generate a cube. In a PV model of a cube a parameterization schema will have length, width and height as the parameterized attributes. Any parametric variations will result in cube-like shapes or parallelepipeds, but no parametric variation will transform the cube into a cylinder, or will create an oblique solid.

If we take a closer look to which attributes can be parameterized in the procedure we could list the following:

- The initial shape, in this case the square (the shape as a parameter)
- The direction of the axis
- The length of the axis (which determines the size of the extrusion)

The initial shape (square) can be parameterized in many ways allowing a variety of shapes to be extruded, such as rhomboids, trapezoids and any quadrilateral. Nevertheless, the initial shape as a parameter can also be substituted with any kind other than quadrilaterals, which results in different designs. In addition, the axis does not necessarily have to be a straight line or be normal to the plane containing the initial shape, although this is assumed in a normal extrusion. As a result, the parametric modeling procedure allows all sorts of new designs with oblique and curved shapes. The initial shape can be a pentagon and the axis oblique which creates a different object than the cube both in topology and geometry levels.

## 3 Design procedures for the Sagrada Familia columns

### 3.1 The Sagrada Familia

Located in Barcelona, Spain, The Expiatory Temple of the Sagrada Familia was designed by Antonio Gaudi between 1883 and 1926. Gaudi worked on the project for a total of 43 years at a very slow pace; by the time of his accidental death only 1 of the 18 towers was finished. Knowing that the Temple would not be finished in his lifetime, Gaudi dedicated himself exclusively to the Sagrada Familia for the last 12 years of his life, resigning any other commissions and living on the construction site. In this period, between 1910 and 1926, Gaudi developed a unique language for the forms of the temple, and devoted his efforts to elaborate strategic methods that would allow his apprentices to carry on the work long after his death. His design process is manifested in plaster models he used for design exploration.

### 3.2 Generation process of the column

Gaudi spent a total of two years to develop a strategic methodology for the generation of the columns. The formal language of the columns of the Sagrada Familia represents a synthesis of manipulation of simple geometrical rules to make complex forms resulting in a rich language with no precedents in architecture (Burry, 1993). Gaudi's novel solution consisted in the superimposition of two helicoidal shapes simulating the organic growth existing in plants. He used two opposite rotations, one clockwise and one counterclockwise, thus avoiding the weak look of a single rotated column (Burry, 2002). Both opposite rotations cancel each other and a new shape emerges.

This process of double rotation of the columns is better explained graphically. Figure 6A shows a square shape extruded along a vertical axis with a 22.5 rotation angle. This is a single twisted column. Figure 6B shows the same rotation procedure but on the opposite direction. Again, this is a single twisted column, but with a −22.5 rotation. These are the procedures that generate the *rotation* and *counter-rotation* shapes. When the two shapes are superimposed (Figure 6C) and a Boolean intersection is performed, the resulting shape is the actual column as developed by Gaudi, as shown in Figure 6D. Even though Gaudi did not use Boolean intersections as we know them in modern computers, the resulting shape from the Boolean intersection is analogous to the actual column originally designed by Gaudi (Gomez et al., 1996) .

Gaudi used this method to design all the columns of the temple, varying in sizes and shapes, according to a hierarchical order and their location in the temple. The bigger columns are located on the central nave and the crossing, while the smaller columns are on the lateral nave and the upper parts supporting the vaulted ceiling. The bigger columns have a larger diameter and the initial shapes are larger polygons, while the smaller columns are made with smaller shapes. In addition, the rotation angle is in direct proportion to the height and diameter of the column, a larger column has a smaller rotation than a small one.

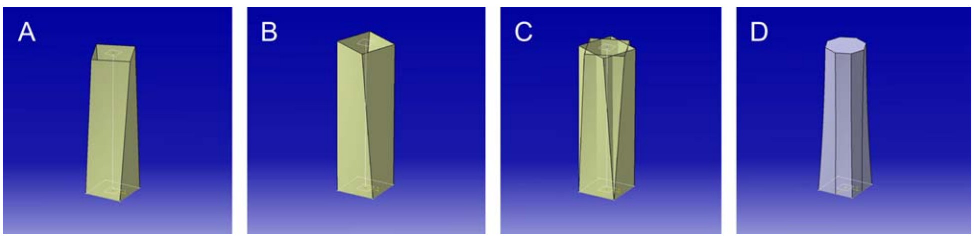


*Figure 6 Generation of the Sagrada Familia columns. The first image shows the rotation of 22.5° of the rectangular shape (rotation). The second image shows the same shape with the rotation angle done in the opposite direction (counter-rotation). The following image shows the superimposition of the two rotated shapes, which is only possible to visualize in a computer model. The last image shows the Boolean intersection of the two rotated shapes, which generates the form of the column. This column is known as the Column of Four, because is generated with a square shape*

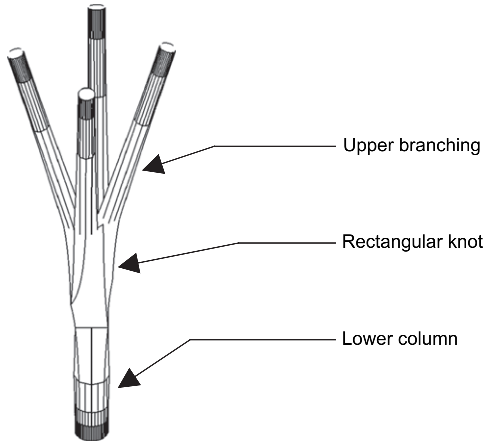


*Figure 7 Lateral nave column showing the lower column, the rectangular knot and branching. The rectangular knot serves as a transition component between the lower column of six and the upper branching*

## 3.3 Generation of the rectangular knot

The rectangular knot is located on the lateral nave and serves as a transitional piece between the lower part of the column and the branching elements above. The lateral nave is supported by a composition of a six-sided column which branches into four small four-sided columns (Figure 7). The transition from the column to the branches is done through a special shape called a knot. The rectangular knot serves both as a capital for the column of six and as a base for the upper branching structure.

Following the same procedure of double rotation present in all the columns of the temple, the rectangular knot takes it's name from the rectangular shapes that are use to generate it. The rectangular knot is created by two rectangular shapes oriented at 90° to each other that twist 45°. The rotation and counter-rotation produce two opposite twisted shapes (Figure 8A and B) that are superimposed (Figure 8C) and intersected to obtain the rectangular knot in its final form (Figure 8D).

## 3.4 Design procedure for the rectangular knot

The parametric model of the rectangular knot was made using a design procedure that is able to generate all the columns in the Sagrada Familia. Unlike Gaudi's method of double rotation of one shape, the design procedure takes four figures as the initial shapes of the column. The four initial shapes are grouped in two pairs: the rotation pair and the counter-rotation pair. None of the four figures that form the initial shapes have parameterized attributes, thus they are simply explicit geometrical

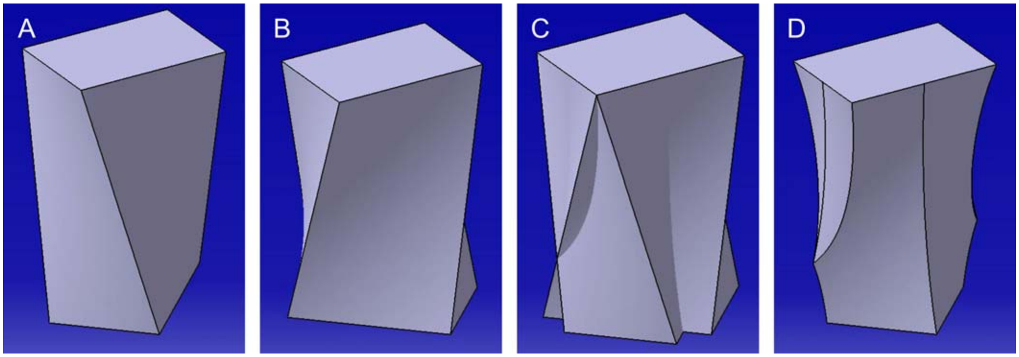


*Figure 8 The generation procedure for the rectangular knot is shown here. The procedure follows the same notion of double rotation of a geometrical shape to generate a superimposition and finally a Boolean intersection*

forms. The two pairs form a wire-frame skeleton (Figure 9) from which two twisted shapes are created by surface fitting functions. The rotation and counter-rotation shapes are generated from each respective rotation and counter-rotation pair. The superimposition and the Boolean intersection occur simultaneously in one operation (Figure 10).

The design procedure only differs from Gaudi's method by using 4 initial shapes instead of 1 while the superimposition and the Boolean operation remain unchanged. This small difference accounts for a larger number of variations due to the fact that the design procedure is not constraint by one initial shape. When the procedure is

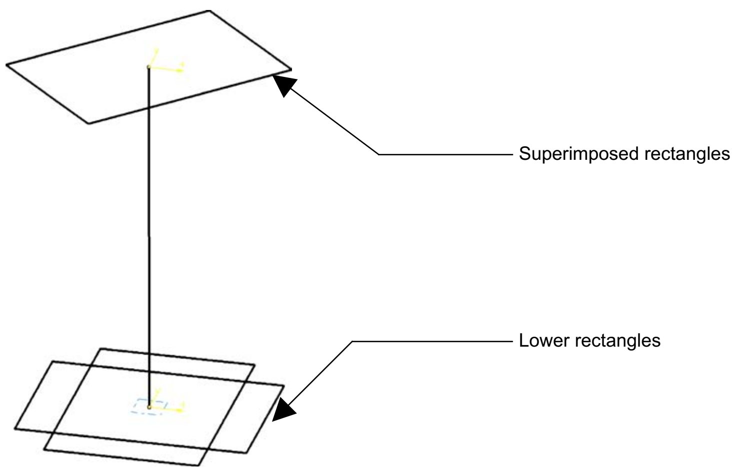


*Figure 9 Initial shapes of the design procedure that generates the rectangular knot. Each pair of rectangles generates the rotation and counter-rotation. The two lower rectangles are oriented at 90° while the 45° rotation of the twisting causes the upper rectangles to be superimposed*

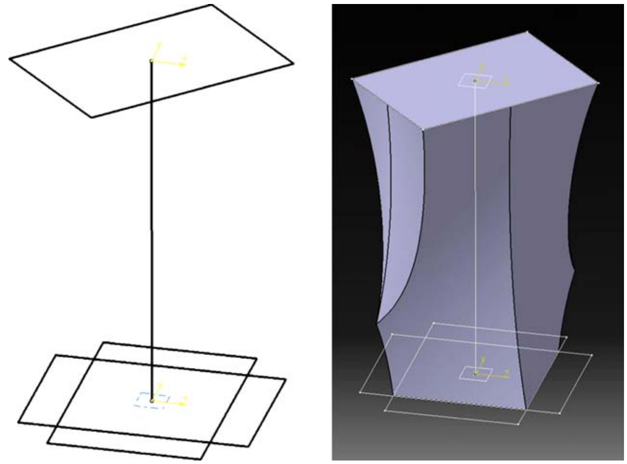

*Figure 10 Initial shapes and solid representation of the rectangular knot generated from the design procedure*

used with four rectangles as initial shapes, the column knot is generated. The model obtained by the design procedure was compared with the original model by Gaudi and no significant differences were found, which lead to the conclusion that the design procedure is formally accurate.

## 3.5 New designs

This design procedure allows the generation of new designs of the column knot by treating the initial shapes as parameters. Multiple design instances were obtained immediately without making new parametric models. The designs generated from the substitution of the initial shapes were of special interest, since the design procedure permitted topological changes to the parametric model. The initial shapes included not only regular polygons, but also irregular shapes, curved shapes and a combination of straight and curved lines (Figure 11).

When doing the initial shape substitution to the design procedure, there are some important restrictions to be considered: (1) the initial shapes must be closed shapes, since and open shapes cannot generate closed solids (inconsistent topology) and (2) the initial shapes must not be self intersecting entities, since self intersecting shapes will generate cusps in 3D. If the two previous conditions are fulfilled, a valid design instance can be obtained from the parametric model. This will automatically generate an exponential growth of the number of instances that the design procedure can produce.

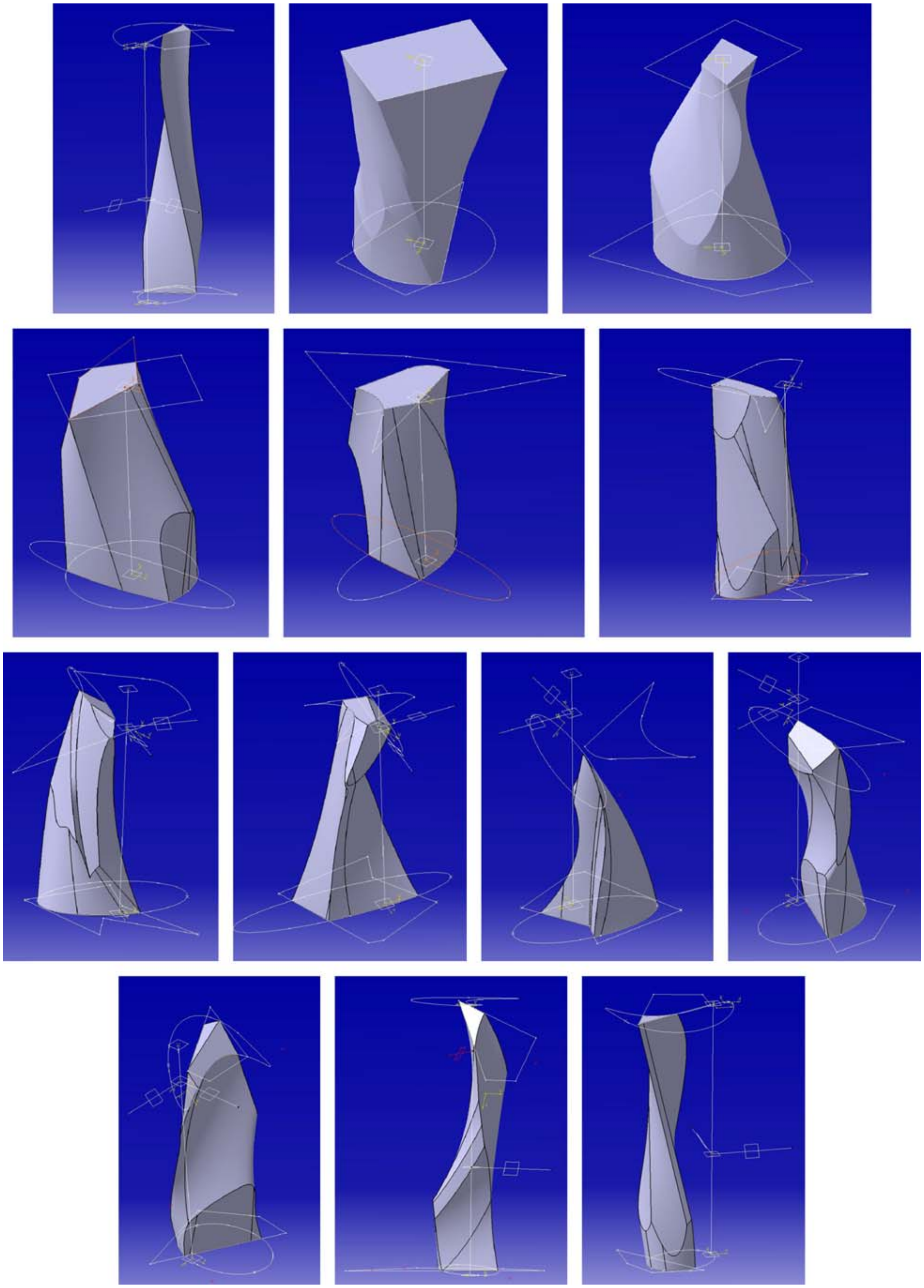

*Figure 11 Family of Designs generated from the Design Procedure. While the implicit parameters of the model remained unchanged, the shapes that form the initial and final pairs, shown in wire-frame, are subject to geometrical and topological changes*

# 4 Discussion

From a computational point of view, Design Procedures can be understood as a search-problem in a very large space of possible solutions. This task can be very expensive even with the most advanced search algorithms. On the other hand, a design procedure offers designers a powerful way to quickly generate parametric models that they can use for design exploration. Search for solutions in a large space of possibilities can be very provocative for a designer; another approach is to implement intermediate solutions where design procedures are constrained to produce certain designs only. These kinds are defined as *deterministic design procedures*.

Parametric models have the general purpose of providing a framework for high-level manipulation of geometrical components that perform transformations during the design process. Among the advantages of using those in design are:

1. The facility to perform changes in geometrical components without erasing a redrawing, allowing flexibility for design exploration and refinement.
2. Increased reusability of design solutions by encapsulation. Complex geometrical models can be placed into basic units that are treated as primitive entities.
3. Added rigor to design development, since a properly constrained parametric model allows some types of transformations, while restricting others.
4. Real time feedback when changes in the parametric model affect geometrical components or other parts of the design.

Design Procedures brings to the surface an important question concerning the validity of designs with respect to the design language. As previously mentioned, variations of a parametric model create instances which are grouped in a category named a family of designs. By simple analogy, a design procedure creates families of parametric models, in other words, families of families with a greater number of design instances. This matter calls for the evaluation of the parametric models as well as the instances.

Another important aspect to consider is the evaluation of the design instances. Evaluations can be one of three types: (1) performance based; (2) aesthetic (Stiny and Gips, 1978); and (3) compliance. In performance based, a design instance is evaluated with respect to an ideal result, and the model is modified to optimize a solution with respect to the ideal one. Aesthetic evaluation will determine if an instance satisfies a set of values determined by the designer. Compliance asserts if a design instance fulfills a predetermined set of requirements. Any of the aforementioned criteria can be implemented in a design procedure for evaluation of the design instances. The evaluation can be interactive in real time or afterwards.

## 5 Conclusions

Design Procedures are inherently non-deterministic and boundless; therefore it is impossible to foresee all the potential results. This is the major asset that a generative system can offer a designer, in particular during the initial stages of design where multiple solutions are explored almost simultaneously. The most difficult task that remains to be solved is how to overcome the initial setup, which can be a time consuming but worthwhile enterprise. Perhaps a careful and accurate analysis of the pre-conditions of setup would provide some solutions in this regard.

The design procedure was used to recreate the original column designs by Gaudi. Rapid prototypes of these designs were selected to be compared with the original models. No visual discrepancies where found when the rapid prototypes where compared with the Gaudi designs. As a result we deem the design procedure as truthful and accurate.

Design Procedures offers a novel solution to expand the universe for exploration of design instances, in particular as a model for generating parametric designs. Design procedures, which are based on a general course of action followed by a designer, is independent of the geometrical shapes and their representation. As a parametric models generation system, the possibilities for application of the design procedures are absolutely boundless.

## Acknowledgments

The author thanks Larry Sass at the Massachusetts Institute of Technology for providing support with the fabrication of the rapid prototype models.

Thanks are also due to Mark Burry, at the Royal Melbourne Institute of Technology in Australia, for providing advice regarding the process of generation of the columns.

Jordi Fauli and Jordi Cuso from the Junta Constructora de la Sagrada Familia are acknowledged for allowing me access to the Gaudi's

collection of plaster models and providing valuable feedback when I was comparing the original plaster models with the prototypes.

## References


**Barrios, C** Parametric Gaudi, in: *Proceedings of the VIII International Congress of the Iberoamerican Society of Digital Graphics SIGraDi*. Sao Leopoldo, Brazil, November 2004

**Burry, M** (1993) *Expiatory Church of the Sagrada Familia* Phaidon Press Limited, London 98 p

**Burry, M** (2002) Rapid prototyping, CAD/CAM and human factors *Automation in Construction* Vol 11 No 3 pp 313–333

**Gomez, J et al** (1996) *La Sagrada Familia: de Gaudi al CAD* Edicions UPC, Universitat Politecnica de Catalunya, Barcelona pp 166

**Monedero, J** (2000) Parametric design: a review and some experiences *Automation in Construction* Vol 9 No 4 pp 369–377

**Stiny, G and Gips, J** (1978) Algorithmic aesthetics: computer models for criticism and design in the arts 220 p


## Further reading


**Knight, T W** (1983) Transformations of languages of designs *Environment and Planning B: Planning and Design* Vol 10 (part 1) 125–128; (part 2) 129–154; (part 3) 155–177

**Mitchell, W J** (1977) *Computer-aided architectural design* Petrocelli/Charter, New York 573 p

**Mitchell, W J and Kvan, Thomas** (1987) *The art of computer graphics programming: a structured introduction for architects and designers* Van Nostrand Reinhold, New York 572 p

**Mitchell, W J** (1990) *The logic of architecture: design, computation, and cognition* MIT Press, Cambridge, MA 292 p

**Thompson, D A W** (1992) On growth and form, in **John Tyler Bonner** (ed) *On growth and form* an abridged edition, Cambridge University Press, Cambridge 345 p