Shape Grammar

ABSTRACT

Shape grammars were invented over twenty-five years ago by Stiny and Gips. They were one of the earliest algorithmic systems for creating and understanding designs directly through computations with shapes, rather than indirectly through computations with text or symbols. Over the years, shape grammars have been explored through applications addressing a variety of design problems. The history of these applications in architecture and the arts is sketched in this paper. The roles of shape grammar applications in education and practice are outlined. New and ongoing issues concerning shape grammars in education and practice are discussed.

INTRODUCTION

Shortly after shape grammars were invented by Stiny and Gips, a two part project for shape grammars was outlined by Stiny. In a 1976 paper Stiny described “two exercises in formal composition”. These simple exercises became the foundation for the many applications of shape grammars that followed, and suggested the potential of such applications in education and practice. The first exercise showed how shape grammars could be used in original composition, that is, the creation of new design languages or styles. The second exercise showed how shape grammars could be used to analyze known or existing design languages. Both exercises illustrated the unique characteristics of the shape grammar formalism that helped motivate almost a quarter century of shape grammar work. General but simple, formal yet intuitive: qualities that continue to make shape grammar disciples and confound skeptics.

Shape grammar theory and applications are well documented and represented in the literature on design computation and related areas. A shape grammar is a set of shape rules that apply in a step-by-step way to generate a set, or language, of designs. Shape grammars are both descriptive and generative. The rules of a shape grammar generate or compute designs, and the rules themselves are descriptions of the forms of the generated designs.

Shape grammars have properties aimed at making them especially suitable for designing, without sacrificing formal rigor. First, the components of shape rules are shapes: points, lines, planes, or volumes. Shape rules generate designs using the shape operations of addition and subtraction, and spatial transformations familiar to designers such as shifting, mirroring, and rotating. In short, shape grammars are spatial, rather than textual or symbolic, algorithms. Second, shape grammars treat shapes as nonatomic entities—they can be freely decomposed and recomposed at the discretion of the designer. This liberty allows for emergence—a feature that distinguishes shape grammars from set grammars, the most common kind of formal grammar. Emergence is the ability to recognize and, more importantly, to operate on shapes that are not predefined in a grammar but emerge, or are formed, from any parts of shapes generated through rule applications. Third, shape grammars are nondeterministic. The user of a shape grammar may have many choices of rules, and ways to apply them, in each step of a computation. As a design is computed, there may be multiple futures for it that respond differently to emergent properties, or to other conditions or goals.

To the right is a two-rule grammar that illustrates these properties. The first rule shifts a square halfway along a diagonal axis of the square. The second rule shifts an L-shape, also along a diagonal axis. Registration marks in each rule show the positions of the shapes on the left-side and right-side of the rule relative to each other. The starting shape for computations, called the initial shape, consists of two L-shapes. The two rules apply to this shape and to shapes produced from it by matching the square or L-shape on the left-side of either rule with a square or L-shape in a design. The square or L-shape in either rule may be translated, rotated, reflected, or scaled in order to match a shape in a design. If a match is made, the matched shape in the design is then replaced with a shifted shape as indicated in a rule. The direction of the shift depends on the spatial transformation used to make
the match.

Below is a computation of a design using the grammar. From the second step on, the rules can apply to either emergent L-shapes or emergent squares. Also from the second step on, either the first or the second rule can be applied to a design. The user of the grammar, human or machine, must decide which rule to apply and to which shape in a design to apply the rule.

Below is another computation using the grammar. The computation is identical to the one above in the first three steps. Then it diverges and follows a different path to produce a different design. Many other computations are possible with the grammar.

Shape grammar theory has advanced over the years to include complexities of shapes and shape computations beyond what is illustrated above. Parametric shape grammars compute designs with variable or parametric shapes. Color grammars and grammars with weights compute designs with shapes and properties of shapes (such as color, material and function). Description grammars compute descriptions of designs. Structure grammars compute designs as structures or sets of shapes. Attributed grammars compute designs with attributes and constraints on attributes. Parallel grammars or grammars defined in multiple algebras simultaneously compute different shape, text, or symbolic representations of designs (for example, plans, sections, and elevations together with verbal descriptions of them). All of these extensions to the original shape grammar formalism have been developed in order to compute certain kinds of designs more easily or expressively than with a standard shape grammar. However, none add to the computational power of a standard shape grammar which is equivalent to a Turing Machine, the most powerful computational device yet defined.

The history of shape grammar applications in architecture and the arts for the two complementary problems of original design and analysis is sketched in the first section of this paper. These two categories of applications do not have rigid boundaries, and are used mainly as a framework for discussion. An overview of the roles of shape grammar applications in education and practice is given in the second section. New and ongoing issues concerning shape grammars in education and practice are discussed in the last section.

Original Design: Shape Grammars

Interestingly, the earliest applications of shape grammars were in an area and for a purpose quickly dropped and not taken up again for a number of years. The first published paper on shape grammars by Stiny and Gips in 1972
illustrates shape grammars for original languages of paintings. The published theses of Gips and Stiny both from 1975, and the joint Stiny and Gips book *Algorithmic Aesthetics* from 1978 also illustrate the shape grammar formalism with original grammars for paintings. The shape grammars in these works are embedded in aesthetic systems for interpreting and evaluating works of art.

Stiny and Gips do not explore or explain the genesis of the original grammars they give in their early works. A specific approach for creating original grammars “from scratch” was first proposed in 1980 by Stiny in his paper, “Kindergarten grammars: designing with Froebel’s building gifts.” Stiny examines the kindergarten method of Frederick Froebel and its analogy in the studio method of designing, and then proposes a computational alternative to these mostly intuitive methods. He uses Froebel’s building blocks in the many simple and elegant shape grammars and designs created with this approach. These shape grammars are the first defined in a three-dimensional space, laying the groundwork for three-dimensional architectural grammars to come.

Stiny’s kindergarten programme for creating original grammars lay dormant for several years while analytic applications of shape grammars grew quickly. In papers beginning in 1992, Stiny’s programme was taken up by Knight in an expanded approach for creating both shape grammars and color grammars of restricted types. The approach is simple enough to be grasped by nontechnically-oriented designers, yet rich enough to serve as the starting point for complex, sophisticated designs. Knight has put this programme into practice in graduate architecture courses taught at UCLA and MIT.

With Knight’s programme, the development of a shape grammar begins with a vocabulary of shapes and spatial relations between shapes. Spatial relations constrain the ways that vocabulary elements may be combined with one another. They are simple compositional ideas and are the key to shape grammars. They provide contexts for adding and subtracting shapes to create designs. In theory, shapes and spatial relations can be anything at all and are limitless in number. In practice, the constraints of a design problem (site, economic or functional requirements, for example) and the constraints the designer brings to the problem (style or design philosophy, for example) motivate the selection of particular shapes and spatial relations. Thus, the shapes and spatial relations used to compute designs often have implicit meanings and functions in the same way that, in a conventional design process, the lines a designer puts down on paper
Spatial relations are explored through additive and subtractive shape rules. Additive rules are used to define simple shape grammars, called basic grammars. Basic grammars generate all of the simplest designs possible with one or more given spatial relations. They are defined by marking or labeling additive rules in different ways according to symmetry properties of the shapes in the rules. The basic grammars so defined apply recursively to generate different designs by instantiating the same spatial relations under different spatial transformations as directed by labels.

An example of the development of basic grammars and designs is shown to the right. A vocabulary of pillars and a spatial relation between two pillars is defined. Additive and subtractive rules that add and subtract pillars in accordance with the spatial relation are given. The pillar added in additive rule is labeled in sixteen different ways according to the sixteen operations in the symmetry group of the pillar. Each different labeling defines a different basic grammar and design.

In classroom teaching, basic grammars are presented in a linear, stage-by-stage way. In design projects, though, students develop grammars in nonlinear ways that correspond to traditional studio design processes. Shapes, spatial relations, and rules are continually modified and redefined until the designs generated satisfy the general goals of a project. These designs are then fine-tuned by detailing basic grammars to produce more complex grammars, or by using traditional studio methods. Below are three design projects developed with basic grammars. Simple spatial relations between three-dimensional shapes are the basis for these projects.
Original Design: Color Grammars

Basic color grammars are developed in a similar way as basic (shape) grammars. In a color grammar, rules have a color component. Colors in rules may stand for colors in generated designs. More often they are used as indices for other attributes—for example, materials, architectural elements such as doors and windows, or even changes to the geometries of shapes. Basic color grammars can be developed from scratch, starting with a vocabulary of colored shapes and spatial relations between them, or they can be developed from predefined basic (shape) grammars. In the latter case, basic grammars are developed first to explore alternative forms, color is then added to selected grammars to explore ways of articulating and elaborating these forms. Color is then used to explore ways of articulating and elaborating these forms.

An example of the development of basic color grammars and designs is shown to the right. The vocabulary and spatial relation are the same as those in the basic (shape) grammar example above, but with color added. An additive color rule can be defined from the spatial relation. The two pillars in the rule can be repositioned in different ways so that the geometric relationship between the pillars remains the same while color relationships change. Different repositionings of the pillars lead to different color grammars and designs in the same way that different labelings of the uncolored pillar in the example above lead to different grammars and designs. In general, the number of different repositionings of colored shapes in an additive rule depends on the symmetries of the shapes both with and without color. Here, there are $16 \times 16 = 256$ different ways of repositioning the colored pillars in the additive rule. Each different repositioning defines a different basic color grammar. Some of these color grammars generate different spatial forms. Some generate the same form but with different colorings.
Below are several examples of student design projects developed with basic color grammars. Underlying each project are one or two spatial relations between colored forms. Color is used in each project to work out interior and exterior details of buildings, and to satisfy programmatic and aesthetic constraints.

Basic grammars are a restricted type of shape grammar with limited computational power. They are additive, set grammars with no capabilities for emergence. Yet they can produce unexpected, novel, and complex designs. Computer programs implementing basic grammars are useful for rapidly exploring numerous design possibilities. A program by an MIT graduate student, Yufei Wang, allows users to define any spatial relation between three-dimensional orthogonal shapes, and then automatically defines and implements basic grammars from the spatial relation. Output from the program can be sent to a rapid prototyping machine for physical fabrication of designs.
Automated grammars together with rapid prototyping are especially suited for the design of complicated, irregular, or nonorthogonal structures that are difficult or time-consuming to conceive of and model by hand. (The architecture of Frank Gehry is a good example.) Links to other computer-aided design and engineering systems and to advanced building techniques can unite different phases of a computationally-driven design process. Shape grammars provide an ideal front end for such a process and for various innovative design and manufacturing technologies now being explored in practice.

**Original Design: Transformations of Grammars**

The approach for developing original grammars and design languages starts “from scratch” in theory only. In practice, original languages or styles are not created from scratch, but from past or known ones. Original design generally involves some degree of analysis or understanding of known designs—one’s own or those of others—as the impetus for new designs. Indeed, students who use shape grammars in studio projects do not define grammars from scratch. Choices of shapes, relations, and rules are influenced by those seen in other conventional or grammatical projects. Often, grammars are used as an idea generator when a project, begun in a conventional way, is at a standstill. To push a project on, a student analyzes her design or some part of it, extract rules, and then plays with these rules to generate new design possibilities. One or more possibilities may then be selected for further development in the project.

Knight’s work on grammatical transformations formalizes processes like these. In the 1980s, Knight proposed a model for developing new design languages on the basis of existing ones. Languages are created by transforming the rules underlying grammars for existing languages. In other words, a known style is first analyzed by inferring a grammar for it, the rules of the grammar are transformed, and then the transformed rules become the basis for a new grammar and style. Knight’s model had a dual purpose. It could be used to characterize the historical evolution of known styles into succeeding ones. It could also be used to innovate new styles on the basis of given ones. In Knight’s book, *Transformations in Design*, the model is applied to analyze stylistic changes in the work of Frank Lloyd Wright, in De Stijl painting, and in ancient Greek ornamental designs.
In a 1990 paper, Flemming proposed a model similar to Knight’s for teaching architectural composition. General architectural languages based on vernacular or “high-style” traditions are introduced to students. These languages include wall architecture, mass architecture, panel architecture, layered architecture, structure/infill architecture, and skin architecture. Grammars underlying these architectures are also presented. Students use the grammars to learn about the languages, then modify them to generate their own new languages. Thus, Flemming’s strategy is both analytic and creative.

Others have adopted similar teaching strategies. Julie Eizenberg, an award winning architect and coauthor of a shape grammar for Wright prairie houses, has introduced shape grammars in her studio teaching at UCLA, Harvard, MIT, Yale, and elsewhere in an analysis® transformation® synthesis process. Students analyze the buildings of an architect, extract rules, then play with these rules to formulate their own rules for buildings that satisfy a given program.

Analysis

The first two decades of shape grammar applications focused almost exclusively on analysis. Through this work, shape grammars became an established paradigm in design theory, computer-aided design, and related fields. The first analytic exercise with shape grammars was given by Stiny in his 1977 paper, “Ice-ray: a note on the generation of Chinese lattice designs”. The grammar laid out in this paper set the standards for the shape grammars that followed. It is the first parametric shape grammar, showing the power and necessity for parameters to describe and generate existing design languages. With five simple rules, the grammar captures the compositional conventions of lattice designs, generates existing lattice designs and an infinite number of new, hypothetical designs in the same style.

The second analytic application of shape grammars, the Palladian grammar by Stiny and Mitchell from 1978, initiated work on more ambitious and complex parametric shape grammars for architectural styles that continues
today. Included in this work are shape grammars for the architecture of Giuseppe Terragni, Frank Lloyd Wright, Glenn Murcutt, Christopher Wren, and Irving Gill, for the vernacular styles of Japanese tearooms, bungalows of Buffalo, Queen Anne houses, and Taiwanese traditional houses, and for the landscape architecture of Mughul gardens. The Wright grammar is notable for being the first three-dimensional architectural grammar—motivated in part by Stiny's earlier work on kindergarten grammars and the alleged influence of Froebel on Wright's architecture. Shape grammars for styles in the arts more generally include ones for the paintings of Richard Diebenkorn, Georges Vantongerloo, and Fritz Glarner, the chair designs of Hepplewhite, the window designs of Frank Lloyd Wright, and ornamental designs on ancient Greek pottery. The stylistic range of this work, from formal and geometric to more amorphous and organic, is evidence of the generosity of the shape grammar formalism.

An exciting new kind of analysis application is emerging that departs from the pure analysis applications above. Three doctoral students in the Design and Computation program at MIT are working on analysis applications that extend into original design in much the same way that original design applications draw on analysis. Each student is working on a shape grammar that is based on a past or contemporary architectural style. Unlike earlier analytic grammars, these grammars are being developed with very specific practical or pedagogical goals in mind. They are not just meant to be “read”. They are meant to be used. Each grammar will have some degree of flexibility built in so that potential users of the grammar will not only be able to understand and generate designs in the original style, they will be able to generate new designs in an extension of the style. To achieve these goals, the grammars will incorporate new grammatical or other devices such as description grammars, parallel grammars, color grammars, or multiple algebras.

Birgul Colakoglu is writing a grammar for a traditional housing type in Bosnia. The grammar is intended to be used by design students to teach them about this type. It is also intended to be used to generate “interpolations” of the type that can be built in a contemporary Bosnian context. Andrew Li’s shape grammar is for a Chinese building system documented in the 12th century building manual Yingzao Fashi. The grammar is being structured in such a way that it can be used by students to generate and explore variations of the system. Jose Duarte is developing a grammar for a housing system designed by the Portuguese architect, Alvaro Siza. This work is unique in that it encapsulates the work of a living architect as well as a “living style”—that is, a style that continues to be built today. Duarte has the enthusiastic support of Siza in developing his grammar, and anticipates that Siza will test and may ultimately use a computer-implementation of the grammar to develop new houses.
Original Design

Applications in original design have implications for both education and practice. In his early work, Stiny is unreserved in his ambitions for shape grammars, setting his approach in opposition to traditional, “romantic” approaches to design. Shape grammars provide the foundation for a “science of design” and for a “theory of architectural composition.” Moreover, his constructive approach “is proposed in the belief that something like it will ultimately replace the kindergarten method both in the studio and in practice . . . Using rules instead of intuition, the designer need no longer rely on ‘creative inspiration’, the ‘inventive flash’, or ‘individual genius’. Once these barriers to clear thinking in design have been removed, we can begin to answer that persistent query: ‘Where do designs come from?’”

Many of Stiny's early goals have yet to be realized. Certainly, shape grammars are well-suited for teaching composition and visual correlates such as proportion and symmetry. To the author’s knowledge, no comprehensive pedagogy using shape grammars in this way, with the exception of Flemming’s course, has been instituted in a design curriculum. Behind the polemics of Stiny’s comments, are important questions about the relationship between shape grammars and traditional design practice. With a shape grammar approach, one may begin to answer the question “Where do designs come from?” by pointing to rules that generate them. However, this question is then replaced with an equivalent, equally difficult question “Where do rules come from?” It is precisely this question that students ask, often indirectly, when attempting to write a grammar for a design project. Rules do not circumvent intuition, inspiration, and so forth. These “powers” are simply transferred to a different level of design—the design of rules rather than the design of a single object. Just like any other creative process, the design of rules involves intelligence and discipline on the one hand and intuition, imagination and guesswork on the other.

Given that creative abilities, whatever they may be, are a part of grammatical design, what then is the value of rules as opposed to more conventional approaches, particularly for design education? Basically, the value of using rules to design is two-fold. First, rules make explicit or externalize a student’s design ideas so that they can be examined, changed, and communicated more readily. Second, rules make possible multiple design solutions rather than a single solution. However, it is not the multiple solutions themselves that are important. Rather, it is the possibility of choosing between different solutions that is important. The process of evaluating and selecting among different designs again brings into focus a student’s design intentions.

Analysis-driven approaches to grammatical design, such as Flemming’s, have additional benefits. With these approaches, multiple skills are taught in a coordinated way. Students learn about the work of accomplished designers or their own work in progress, about ways of designing, and about ways of developing their own work.
Analysis

Pure analysis applications have much educational potential. There is no better way to learn about styles or languages of designs (at least compositionally) than by either studying shape grammars already written for languages or by writing grammars oneself. Good analytic grammars are both parsimonious and descriptive. They are eye-openers, revealing simplicity or regularities behind designs seemingly complex or random. They reveal the thoughtfulness, the “individual genius”, behind designs that students might otherwise take as unfathomable.

Analytic grammars also embody general design strategies that students can learn from and use in their own work. Different grammars for very different languages (temporally, culturally, geographically) often use common design strategies. For example, in a number of shape grammars, designs are based on an abstract grid or parti. Spaces are delineated within the grid, and then finer details are added within these spaces. The Palladian grammar, the Japanese tearoom grammar, and Li’s Yingzao Fashi grammar all work in this fashion. A number of grammars use subdivision as the basis for designs. This strategy is useful when designs in a language have the same, regular boundary. The ice-ray grammar, the Hepplewhite chairback grammar, the Siza grammar, all of the painting grammars (Vantongerloo, Glarner, and Diebenkorn), and to some degree, the bungalows of Buffalo grammar, work in this way. Other grammars use an additive process for generating designs. This strategy is useful when designs in a language have irregular or diverse kinds of boundaries. With this approach, designs are generated beginning with one part (the core) of a design to which other parts are successively added. The Wright grammar, the Queen Anne grammar, and Colakoglu’s Bosnian house grammar follow this approach.

The general design strategies that cut across different grammars and the more specific ones embodied in particular grammars are theories of composition and may not correspond to historical fact. That is, a grammar may have nothing to do with the way designs were originally conceived or the process by which they were originally created. In fact, many analytic grammars do not even make sense from a design process point of view when specific details of rules are examined (see section on authoring a grammar below). In this sense, a grammar may be unbelievable. Yet, there is no way of verifying whether a grammar is historically accurate, even with the testimony of a living designer. Thus, shape grammars are not, to use AI terminology, strong theories of styles, but weak theories of styles. The more compelling a grammar is, though, the more it may seem to correspond to historical reality. Stiny’s ice-ray grammar is a good case in point. Stiny suggests in his paper that it is easy to imagine that Chinese artisans constructed ice-ray window grilles using exactly the process encoded by his grammar: “Indeed, the steps in the ice-ray lattice generation . . . could well comprise the frames in a motion picture of the artisan creating his design!” The grammar for Greek ornamental designs is believable in the same way.

Most shape grammar authors, though, do not view historical truth or practicability as goals for their grammars. A
well-crafted grammar, believable or not, may be used to classify designs and to predict unknown or hypothetical ones successfully. And it can serve as the platform for theories of style that go far beyond compositional issues, even so far as to explore historical issues.

Pure analytic grammars have yet to be used in design practice. In architecture and the arts, fields with a heavy emphasis on originality and novelty, it seems unlikely that a designer would implement any grammar but her own (original) grammar. However, the new, hybrid analysis/original design applications being worked on by researchers in architecture (for example, Duarte’s and Colakoglu’s grammars) may take these grammars out of the classroom very shortly. These applications seem to have the most immediate promise for architectural practice. In engineering and product design, either pure or hybrid analytic grammars have strong practical possibilities. Agarwal and Cagan’s grammar for coffeemaker designs is a good example of a grammar that could be put into practice very effectively.

NEW AND ONGOING ISSUES IN EDUCATION AND PRACTICE

Shape grammars are more than twenty-five years old, but their potential in education and practice is still far from being realized. Shape grammar theory is now far in advance of practical applications. Why? What can be done to narrow this gap?

The number of design (architecture, arts, engineering) schools that include shape computation or shape grammars as part of their curriculum has grown steadily as graduates of shape grammar programs find teaching positions worldwide and establish their own shape grammar courses. Shape grammars now have a much wider reach than they have ever had, but practical accomplishments have grown slowly. Some issues and questions need to be addressed for continued and more rapid progress. A few of these questions are considered in this section.

Pedagogy

Questions of pedagogy -- how to teach shape grammars -- are important to address. Shape grammars, like their Turing equivalent, can compute almost anything. To understand and deploy the full capabilities of shape grammars, a unique combination of technical, spatial, and intellectual abilities and interests is required. A number of graduate students and researchers with these abilities and interests continue to push the boundaries of shape grammar theory. However, there is a wider population that can enjoy, learn from, or use shape grammars very profitably that needs to be reached. Introducing shape grammars to this population raises questions about teaching strategies.

Should the teaching of shape grammars be theory-based or practically-based? In other words, should shape grammars be taught through their mathematical and philosophical foundations, should they be taught through concrete, practical applications, or through some combination the two? Similarly, should the shape grammars first introduced to students be abstract ones like the kindergarten grammars or interpreted ones in which shapes have definite meanings or functions? An advantage to abstract grammars is that they demonstrate easily the mechanics of shape grammars. Also, they do not limit possible interpretations of the designs they generate. Designs can be read in multiple ways leading to multiple applications. For beginning design students, though, abstract grammars can have disadvantages. Students with limited or no design experience may not be able to see beyond, or generalize from, what they physically have in hand (a 2” x 1” x 1/2” wooden block, for example), thus limiting their understanding of what grammars can be used for. Interpreted shape grammars, such as Flemming’s wall and panel grammars, may be more instructive or motivational for beginning students.

Regardless of whether the teaching of grammars is theory-based or practically-based, the spatial abilities needed to grasp shape grammars, especially in three dimensions, require practice even for experienced designers. In this author’s experience, the best way for students to understand how grammars work is through concrete and
collaborative exercises. Hands-on implementations of grammars with physical 2D or 3D shapes, that are manipulated manually in space, or with 2D drawings of shapes (on trace) is essential. Because shape grammars are unlike anything most students have ever seen before, students call on a wide range of skills to understand what they are doing. Collaborative work allows students to share their different learning strategies.

Computer Implementation

Strongly tied to pedagogical questions, are questions of computer implementation. When and how are computer implementations of shape grammars useful for students or practitioners? What should these implementations look like?

For teaching purposes, computer implementations may not be as effective as by-hand applications of grammars. Slow, by-hand applications of rules require careful thinking about how rules work. In the long term, this results in a better understanding of grammars and better quality design work. Computer implementations of grammars can encourage mindless defining and testing of rules. Interesting or useful designs may be arrived at, but by chance and with no understanding of how the designs were generated or how to generate equally good or better results. Also, computer implementations provide only two-dimensional representations of generated designs. Three-dimensional grammars can generate complex designs that are difficult to comprehend without a physical, 3D model. A two-dimensional representation of a three-dimensional object, no matter how sophisticated, cannot compete with the object itself. Linking rapid prototyping technologies to computer implementations may be a solution to this problem.

There are, of course, overwhelmingly strong reasons for having computer implementations. Computer implementations are good demonstration tools for showing novices the range and power of shape grammars. They can allow students and designers who do not wish to deal with the technicalities of grammars, to develop or use shape grammars with success. For advanced shape grammarians, who understand how shape grammars work, they allow for rapid explorations of rules and design possibilities. Shape grammars are powerful devices and the power of computers is needed to explore their limits.

Currently though, there are few computer implementations that are useful for students or practitioners. Most do not have interfaces that make them easy for nonprogrammers to use. More efforts have gone to computational problems than to interface ones. Implementations of simple, restricted grammars that require only graphic, nonsymbolic, nonnumerical input are needed. Successively more general and powerful implementations can be built from these. Tapia’s program for two-dimensional, nonparametric shape grammars is a great start. Still simpler programs for three-dimensional set grammars developed by Duarte and Simondetti and by Wang at MIT have done much to promote the use of shape grammars among graduate design students at MIT.

Authoring a Shape Grammar

As more students and researchers explore the possibilities of putting shape grammars into practice, questions of authoring a shape grammar for practical applications are emerging. These questions are different for original design applications and for analysis (pure or hybrid) applications.

In original design applications, the author and user of a grammar are the same: a designer. Are some kinds of shape grammars more suitable than others for designers to work with? Like computer implementation issues, consideration must be given to the power and generality of a grammar on the one hand, and ease of use on the other. Knight has proposed that simple, restricted types of grammars with a minimum of shape grammar
paraphernalia (parameters, labels, etc.) may be best suited for the early, conceptual stages of design. Restricted grammars are easy to design, easy to understand, and they can generate a multitude of innovative design possibilities. Generated designs can be elaborated either by elaborating the grammar or by traditional means.

One critical problem in authoring an original grammar is how to develop a grammar that meets the goals and constraints of a particular project. A commonly asked question by design students writing grammars for a project is “How should I start?”. At some point in the process of developing a grammar—if not at the start—a connection must be made between rules that describe spatial form, and the goals of a project that may describe anything from function to meaning to aesthetics and so on. Making this connection is not an easy task because shape grammars are in general unpredictable.

Seemingly simple rules can produce surprisingly complex results. Take, for example, the rule below which shifts a triangle (from Lionel March). The rule applies recursively to two triangles to produce unexpected results.

![Recursive application of rule](image1)

Conversely, seemingly complex rules can produce surprisingly simple results. Take, for example, the rule below and the result of applying the rule recursively to the shape on the left side of the rule.

![Recursive application of rule](image2)

Different approaches to connecting grammars and goals have been suggested. One approach is direct. It involves writing rules with the foreknowledge that the generated designs will meet, or start to meet, given goals. In order to do this, the behaviors and outcomes of rules must be predictable in some way. In a 1987 paper, Flemming recognized the problem of predictability, writing “There is, at the present time, no body of theory available that would allow us to predict the properties of shapes generated by a grammar solely from an inspection of its rules.”

Recently though, Knight has attempted to establish just such a theory. Assuming a lack of theory, Flemming suggested: “In order to assure that a grammar is properly constructed, we often have to enumerate a substantial number, if not all of the shapes it generates. This process is tedious and error-prone if done manually and could clearly gain from automation.” This implies an alternate, indirect approach to connecting grammars and goals that has also been researched in recent years. With this alternate approach, grammars are developed without a clear idea of their outcomes. An automated search and test strategy is then used to explore the space of designs generated, sampling designs and testing them to see if they meet given goals. Cagan’s shape annealing technique is a successful example of this approach.

In analysis applications of shape grammars, the author and the users of a grammar are different. Traditional analytic shape grammars were intended for a diverse audience from historians to designers who use the grammars for educational purposes—to understand a particular style. Criteria for authoring a successful analytic grammar for a style were spelled out early by Stiny and Mitchell. Requirements of a grammar are that: (1) it should clarify the underlying commonality of structure and appearance manifest for the buildings in the corpus; (2) it should supply the conventions and criteria necessary to determine whether any other building not in the original
corpus is an instance of the style; and (3) it should provide the compositional machinery needed to design new buildings that are instances of the style."

In newer, hybrid analysis/original design applications of grammars, the author and the users of a grammar are also different. However, unlike pure analytic grammars, these grammars are intended for an audience who use the grammars for practical design purposes as well as educational ones. New criteria for authoring these new grammars are called for. These criteria depend in part on the desired level and kind of user interaction with the grammar. For example, computations in a grammar can be automated and controlled minimally and indirectly by a user through the input of constraints or goals. These constraints can guide computations to produce certain designs, or they can be used to evaluate and select certain designs that are output by computations. Maximum user interaction might involve a user choosing which rule to apply and how to apply it in each step of a computation. In this case, the user's role more closely approximates the role of a designer.

When a user has liberal control over computations, then the grammar must be structured in a designernly way in order to be practicable. Most traditional analytic grammars are not structured in this way. In other words, the way in which choices are presented to a user may not make sense from a design point of view. This is particularly true for choices relating to parameters. With some architectural grammars, for example, a user must decide the dimensions of individual spaces in a plan before the arrangement of the spaces with respect to one another is decided. In the first stage of the Palladian grammar, for instance, the modules of a grid plan are generated individually to define the underlying plan of a villa. As each module is added, it must also be dimensioned. Thus, the user must decide the dimensions and proportions of individual modules before knowing (generating) the size and proportions of the overall grid. In the Wright prairie-house grammar, the dimensions and proportions of some spaces in the core unit of a house must be decided before the functions of these spaces are assigned. Problems such as these arise because shape grammars do not easily allow for the separation of dimensioning from other choices. More generally, dependencies among different properties of designs are sometimes difficult to structure practically in a shape grammar. Choices about local properties must often be made prior to the determination of global properties.

In his work on the Yingzao Fashi building system, Li is exploring a solution to problems such as these with the use of parallel grammars. Parallel grammars allow different properties and representations of designs to be separated into different computations, while allowing for these computations to communicate with and influence one another in appropriate ways. Duarte and Colakoglu are also exploring the use of parallel grammars in their work, not only as a solution to the parameter problem but as a way of generating multiple representations of designs.

When a user has minimal control over computations in a grammar, then different authoring issues arise. For example, if a user's only interaction with a grammar is through the input of constraints or goals, then mechanisms need to be built into the grammar to find or generate just those designs that satisfy the constraints. (This issue is similar to the problem of connecting original grammars with goals discussed above.) For example, Duarte's grammar for Siza houses is intended to be used by Siza or his clients. For clients with particular requirements (a number of bedrooms, for example), a mechanism is needed to find or generate just those houses in the language that satisfy the requirements. Duarte is exploring a number of automated mechanisms either embedded within the grammar or external to it that will perform this task. One solution to the general problem of searching through languages to find particular designs involves the use of description grammars. A shape grammar can be linked with a parallel description grammar so that a design and a description of the design are generated in parallel. Every design generated by the shape grammar thus has a description (including number of bedrooms, for example). Given some requirements, the rules for generating a description that includes the requirements can be used as input to run the shape grammar. This input specifies which shape rules to apply to generate a design with that description.
These are just a few of the practical issues that have emerged in recent years. More work than ever before is now being directed toward these issues. Recent shape computation workshops, a new shape grammar website, and a generally more active and widespread dialogue among researchers and students internationally may help to place shape grammars in mainstream education and practice in the near future.