Tomaž Berčič:

PROCES POSTOPNIH SPREMEMB: OBLIKOVNA SLOVNICA V PARAMETRIČNIH ORODJIH

THE GRADUAL PROCESS OF CHANGE: INTEGRATING SHAPE GRAMMARS IN PARAMETRIC TOOLS

DOI: https://doi.org/10.15292/IU-CG.2024.12.046-054 ▮ UDK: 72:004.925.8:37.091.3 ▮SUBMITTED: October 2024 / REVISED: November 2024 / PUBLISHED: December 2024

1.01 Izvirni znanstveni članek / Scientific Article

POVZETEK

Metodologija oblikovne slovnice je učinkovito orodje za študente in strokovnjake v oblikovalskih disciplinah. Ta članek prikazuje integracijo oblikovne slovnice s sodobnimi parametričnimi orodji, pri čemer se zavestno izogiba uporabi namenske programske opreme, da bi omogočil rabo oblikovnih slovnic v praksi.

Glavni cilj raziskave je razširiti vedenje o praktičnih koristih oblikovne slovnice v izobraževalnem in profesionalnem okolju ter raziskati načine njene implementacije, ki izboljšujejo uporabniško izkušnjo in integracijo. Obstoječa orodja - interpretatorji za oblikovno slovnico pogosto niso dovolj uporabna zaradi slabe integracije z uveljavljenimi programskimi platformami. Raziskava prikazuje, kako lahko oblikovno slovnico učinkovito vključimo v parametrično programsko okolje, kot je Rhinoceros 3D, s pomočjo razširitve Grasshopper.

Ta pristop sicer ni nujno preprostejši od namenskih interpretatorjev, vendar dokazuje njegovo izvedljivost v poznanem parametričnem oblikovalskem okolju. Študija vključuje šest primerov uporabe oblikovne slovnice v Rhinoceros 3D z razširitvijo Grasshopper. Rezultati kažejo, da je metodologijo oblikovne slovnice mogoče brez težav prilagoditi standardnim orodjem, saj omogoča intuitivno oblikovanje množice variant, prilagajanje pravil in učinkovito vrednotenje rešitev.

KLJUČNE BESEDE

oblikovna slovnica, parametrično oblikovanje, Grasshopper, parametrična orodja, arhitektura, izobraževanje

ABSTRACT

The shape grammar methodology presents a compelling tool for students and professionals in design disciplines. This article illustrates the integration of shape grammar with contemporary parametric tools, explicitly avoiding the reliance on specialised custom software to facilitate its application in routine practices. The study's primary objective is to broaden the discourse on the practical benefits of shape grammar in both educational and professional settings while exploring methods of implementation that enhance user interaction and integration.

Current shape grammar tools often fall short in usability due to insufficient integration within commonly used software platforms. This research demonstrates how shape grammar can be implemented within established parametric software, such as Rhinoceros 3D, using Grasshopper. This approach does not claim to be simpler than dedicated interpreters but demonstrates its feasibility within a familiar parametric design environment. The study provides six examples showcasing the application of shape grammar within Rhinoceros 3D, utilising the Grasshopper parametric extension. The findings reveal that the shape grammar methodology can be seamlessly adapted to align with standard tools, facilitating further modifications, re-creation, and evaluation of design data. The adaptability of the shape grammar system underscores its potential applicability across diverse design fields. Using a Visual Programming Language (VPL) embedded directly in the software environment enhances functionality, thereby addressing a significant limitation of shape grammar theory by enabling the evaluation of generated design alternatives.

KEY-WORDS

shape grammar, parametric design, Grasshopper, parametric tools, architecture, education

1. INTRODUCTION

In architectural education, students often rely on predefined solutions, limiting their creative exploration. This study examines how computational tools inspired by unstructured play principles like Fröbel gifts (Figure 1) can foster creativity by generating diverse design alternatives with minimal direct intervention. Shape grammar represents such a potential tool. The concept of shape grammar is still rarely taught today. In the late 1990s while studying architecture in Slovenia, Fröbel toys were only briefly mentioned in the curriculum. These wooden boxes containing building blocks, known as Fröbel building gifts, can be perceived as an analogue counterpart to the shape grammar system, embodying core principles of design iteration and modularity.

Figure 1: Fröbel gifts—a set of wooden educational blocks designed by Friedrich Fröbel, the founder of the kindergarten movement, to encourage spatial reasoning, creativity, and early architectural thinking.

Friedrich W. A. Fröbel was a 19th-century German educator who introduced innovative approaches to the education of young children. Fröbel gifts include six boxes filled with essential wooden elements (sets 2–6) that children can arrange in numerous ways. They encourage children's spatial awareness at a very young age as intended. The Fröbel system may work as envisioned for young children as they are still unformed individuals, discovering basic solid shapes and relationships between them.

However, young architecture students have moved beyond this early development stage and already have their preconceived ideas. They feel burdened by knowledge and societal norms, constantly overwhelmed by filtered information that caters to their interests. They are, therefore, often unable to return to the zero state that they experienced when they were children (Stiny, 1980a). The core principles of this toy have much potential as an additional learning tool for architecture students and a potential analytical instrument for studying design concepts. By observing an already constructed building or structure, a design student can recreate an existing composition from the given initial shapes and reflect on the relationships between them to understand the architectural reasons why the composition was put together in that way. Students could also use them to propose and argue for a better alternative in a similar or different context based on the rules of the original design. In addition, the method provides students with a concrete methodology for approaching architectural and urban design with infinite solutions while keeping a consistent design language.

The pursuit of new learning structures is due to the pedagogically unique impacts of digital design. Various researchers and educators have begun addressing the need to integrate digital design in architectural and urban design education, examining various pedagogical approaches. Generative and digital design have played a role in shaping the theoretical, computational, and cognitive methods developed by various researchers as a basis for design education and pedagogy. (Knight, 2000; Oxman, 2004, 2006; Cuff, 2001; Knight & Stiny, 2001; Grasl & Economou, 2018)

Shape grammars are rule systems containing an initial shape and transformational shape rules. By repeatedly applying those shape rules to the initial shape, a set of forms that are part of the same family or belong to a particular style can be generated. The kindergarten method is one of the original shape grammars (Figure 2). Stiny (1980b) described the studio-based method as the most effective educational paradigm in architecture and design schools today, comparing the relationship between a young designer and their mentor to that of a child and their mother in a kindergarten.

While shape grammar is a relatively new subject in design theory, several comprehensive attempts have already been made to use the method in painting and sculpture. Shape grammars, introduced by Stiny and Gips (1972), provide a formal method for generating and analysing geometric shapes in art and design. This approach has been applied to various fields, including architecture, painting, and sculpture. Shape grammars allow for the generative specification of non-representational artworks, offering implications for aesthetics and design theory (Stiny & Gips, 1972). Lauzzana and Pocock-Williams (1988) developed a LISP implementation of a design rule system based on shape grammars, demonstrating its application in analysing Kandinsky's Bauhaus paintings. Stiny's (2006) book *Shape: Talking About Seeing and Doing* further explores the mathematical formalism of shape grammars, providing a rigorous approach to understanding and constructing shapes across various design disciplines. This work emphasises the importance of shape grammar in calculating shapes and unpacking the rule-based dynamics of form creation. It is challenging to record an artist's way of creating work because of the complex design process, organic shapes, and many colours and shades. Likewise, describing every move and decision the artists make to create an art piece is challenging. In some cases, it can be easier when the art itself is a system, or the artist rejects classical painting or sculpture and parameterises their art. (Stiny & Gips, 1972)

Prairie grammar, introduced by Koning and Eisenberg (1981), analyses the architecture of Frank Lloyd Wright's prairie houses,

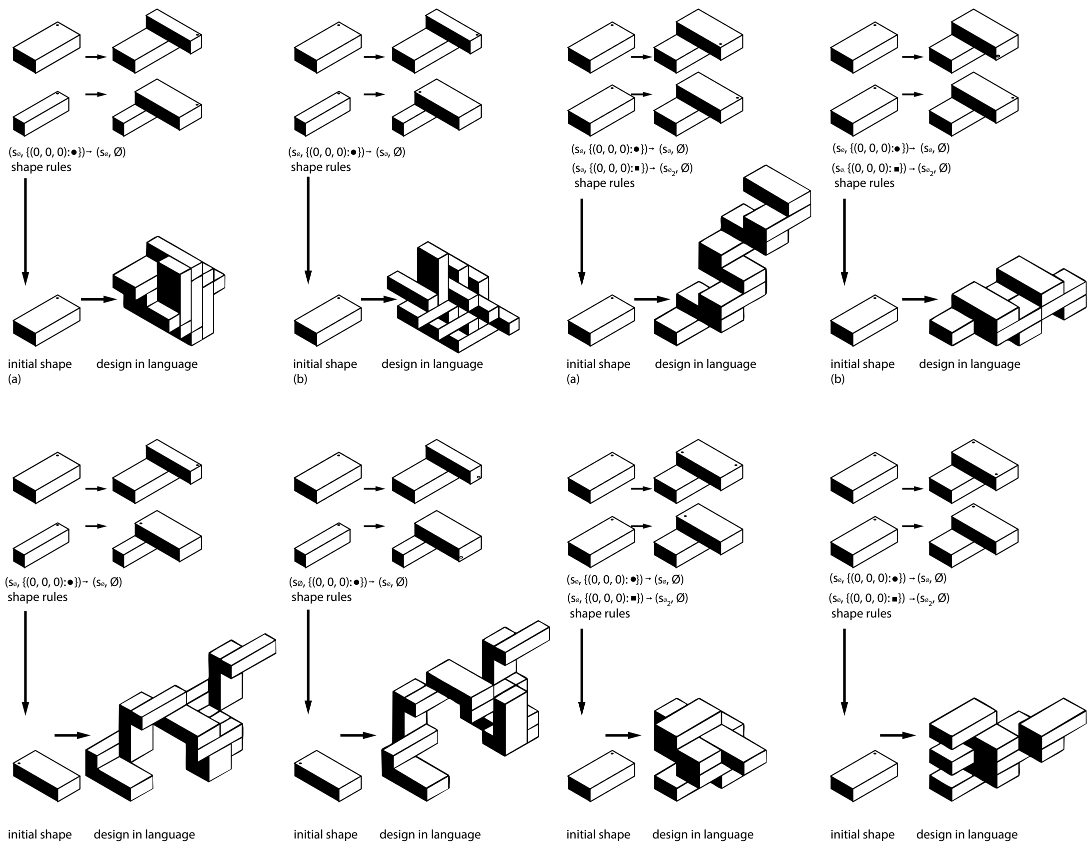


Figure 2: This figure is a redrawn composite of images 43 and 51 from Stiny's (1980b) "Kindergarten Grammars: Designing With Froebel's Building Gifts". It illustrates an example of constructing designs where volumes interpenetrate. The process starts with identical elements in the same initial configuration but applies slightly varied rules for each construction iteration, demonstrating the diversity in design outcomes achievable through procedural modifications.

offering more profound insight into an architect's design thinking for specialised architecture. A key finding from this study is that the fireplace serves as the central element in prairie-style house design. Around it, functionally distinct building blocks are systematically added and arranged to form the fundamental compositions of these homes.

Beyond prairie grammar, most research on shape grammar has been conducted in architecture. Notable examples include analytical grammar, such as those examining the geometric formation of Mughal gardens in India (Stiny & Mitchell, 1980), and Palladian grammar (Stiny & Mitchell, 1978), which generates architectural drawings incorporating wall thicknesses, column and wall placements, and window openings in Palladian villas. The existence of Palladian grammar suggests that architect Andrea Palladio likely had a structured design system, which, according to contextual analysis, guided his villa designs. The intricacy of the design rules in both Palladian grammar and the language of the prairie raises doubt about the architects using this kind of complex language to design their buildings. It raises questions concerning how architects defined their design systems, how these were implemented and with what measure of precision they were followed. Duarte (2001, 2005) demonstrated how shape grammars could be used to mass customise housing, specifically in Álvaro Siza's Malagueira project. This research provides a foundation for applying grammar in architectural practice.

Recently, shape grammar research has also been applied in urban design in a city information modelling platform connected to a GIS database. Beirão and Duarte (2018) demonstrated the parametric design platform implementation called CityMaker, which translates shape grammars and converts them into parametric design patterns that can be used in CAD software. The developed computer platform is at the same time a design platform that supports the generation of design scenarios, a simulation platform that evaluates different scenarios, and a decision support platform that allows designers and stakeholders to discuss and evaluate the results of spatial solutions. The aim was not to automate urban design but to develop an assistive platform which enables urban designers to perform better. Using standard design software, such as Rhinoceros 3D and the Grasshopper visual programming interface, enables urban planners and architects to test and analyse various solutions that this tool provides.

Hong and Economou (2023) explored shape embedding in 2D CAD systems, addressing challenges in implementing shape grammars digitally and providing insights into computational applications. Their work highlighted how shape grammar interpreters could be integrated into practical software for design tasks. While ML-based generative design techniques offer flexibility in handling complex design spaces (McKay et al., 2012), they often lack the interpretability and structured rule-based approach inherent in shape grammar. This study addresses the gap by proposing a method that combines shape grammar's systematic framework with user-friendly parametric software integration.

One of the key challenges in applying shape grammar to architectural and urban design lies in accounting for the complex and interwoven urban phenomena, including policies, social dynamics, typologies, and climate. Given these intricacies, the following critical question arises: Which urban elements can be meaningfully discerned and modelled, and to what extent can their representation remain functional and feasible in producing a viable urban solution (Verovsek et al., 2013)?

This challenge is further reflected in developing a teaching methodology that requires designers to establish rules capable of generating alternative solutions. However, defining too many rules can overcomplicate the design process, mirroring the inherent difficulty of distilling complex urban conditions into a structured system. For instance, the 1961 revision of the New York City zoning ordinance introduced the floor area ratio (FAR) concept, providing an abstract yet measurable way to regulate urban form (Ažman Momirski, 2018). Similarly, students often struggle to formulate explicit rules in their design process, as indicated by the questionnaire results. This difficulty echoes broader theoretical challenges in defining urban solutions through a limited set of rules—an endeavour deemed extremely difficult, if not impossible (Duarte & Beirão, 2011). Comparable issues arise in sustainable planning, where the implementation of design rules remains complex, often requiring subjective interpretation of sustainability guidelines, particularly when addressing cultural and contextual qualities in contemporary architecture (Čok, 2014). Another use of shape grammar can be as a design tool called ChairDNA (Garcia & Menezes Leitão, 2018). Compared with other shape grammar implementations, ChairDNA uses an approach that keeps the combinatorial explosion of rule applications under control, which simplifies the use of the tool by designers who do not have experience in shape grammar.

Despite its many positive characteristics, the shape grammar theory has also been severely criticised for the grammar rules being mere recipes that are never wholly explored. The overwhelming combinatorial complexity also makes it impossible to explore what constitutes an interesting design without a preconceived idea (Stiny, 1980a). Additionally, it is argued that no definitive guideline exists for when to stop if generating, evaluating, and comparing all possible solutions is not feasible. However, thorough examination, particularly during the early stages of design, can present challenges (Fleisher, 1992). Visual and verbal categories become equivalently expressive and immediately accessible. Style can be parsed, thereby parsimoniously explained, exhibited, referenced and compared. The architectural intent is merely a matter of assigning a purpose to each grammar rule (Fleisher, 1992). Nevertheless, even the most prominent critics agree that shape grammar is worthy of further research.

As shown above, the shape grammar system can be applied to any design discipline. At the same time, the methodology is a handy educational tool for students. However, the question remains as to whether it is equally beneficial for professionals. Do they use shape grammar in their daily practice to solve spatial problems? An analogy with parametric design has shown that while using parametric tools available for research and teaching purposes is essential, in practice, they are not used as much as they could be (Hudson, 2010).

2. METHODS AND MATERIALS

To formulate a spatial grammar, the following three things are needed: the elements to which the rules apply, the rules constructed by a type of (Euclidean) transformation, and a Boolean operator (Stiny, 1980a).

To better understand shape grammars and the concepts discussed in this article, it is important to define some key terms (Garcia, 2016):

1. **Initial shape (I)** is the starting shape under which the rules are applied; it can be an empty shape.
2. **Design (D)** is a terminal shape that belongs to the language defined by a grammar; design can also be referred to as an intermediate state (partial design).
3. **Emergence (E)** is the ability to recognise and use emergent shapes not predefined in grammar but ones that emerge during the computation through the detection of sub-shapes and the relation of predefined shapes.
4. **Label (L)**; a label is a symbol associated with shapes. It is initially defined to control the rule application (transformations under which rules apply) but may also represent other aspects of shapes.
5. **Rule (R)**; a rule is from the type A ⊠ B where A and B are both shapes, A being the shape of the rule's left-hand side and B being the shape of the rule's right-hand side.
6. **Language** is a set of all designs (D) generated by the grammar of the design space. Each grammar defines one language of designs.
7. **Derivation** is the step-by-step generation of a design from an initial shape (I) to a final design (D): I =› Sn =› Sn+1 =› ... =› D. It is also referred to as computation.
8. **Transformations**; transformation τ on shape S is denoted

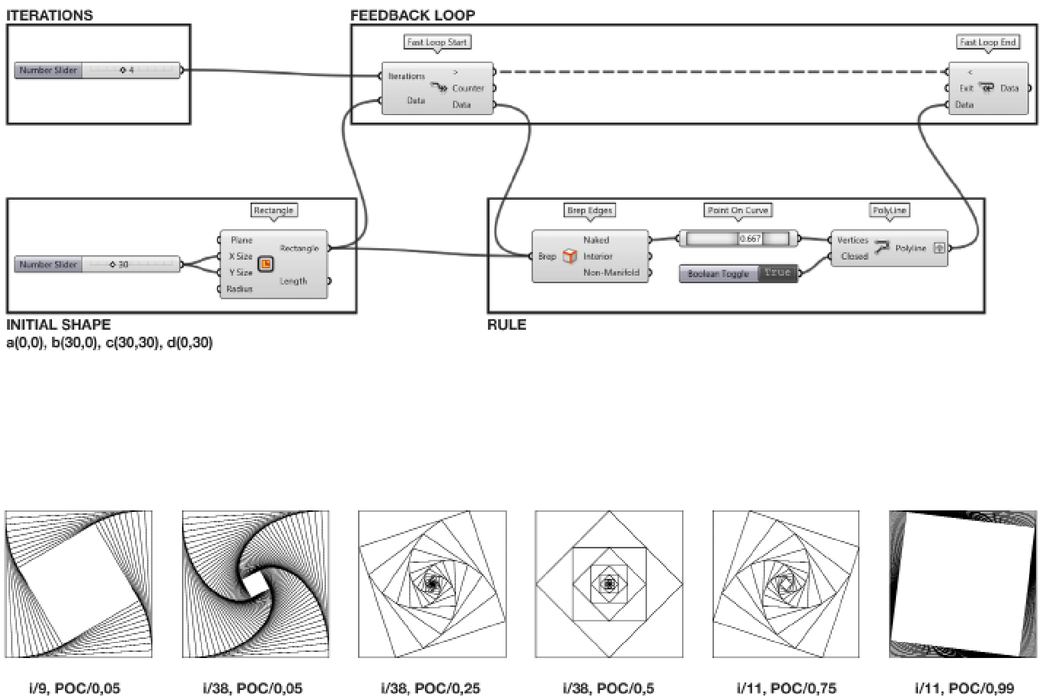
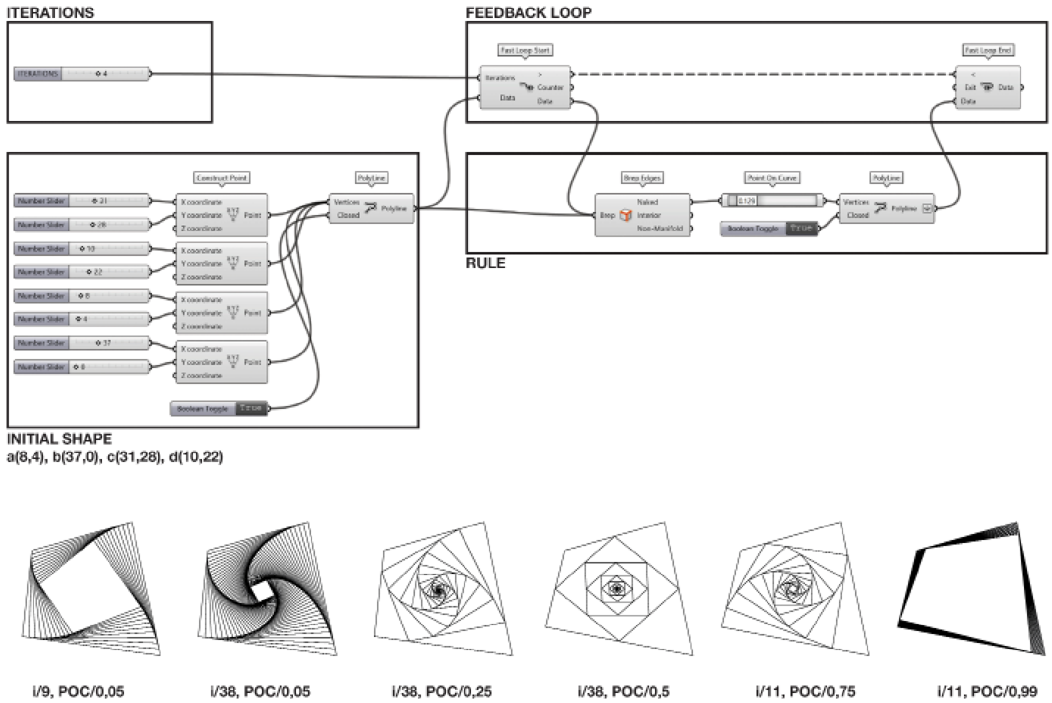


Figure 3: Proof of concept using the initial shape (rectangle) and the rule that directs the creation of a new rectangle from points in the same position on each edge of the rectangle. Instead of repeating the command manually, a feedback loop repeats the rule on each newly created rectangle n times (iterations). An additional benefit of the parametrised shape grammar method is the record for each variation. Hence, the results can always be defined, counted and recreated.

Figure 4: The same configuration rule as in Figure 3 but with an irregular shape. The quadrilateral is defined with four parametrised points, but the rule and iterations remain the same. The results are similar, but the expression of each variation includes additional information on the coordinates of corner points.



by τ(S). The similarity transformations on a shape can be one of the four Euclidean transformations—translation, rotation, reflection and scale—or the composition of at least two.

9. **Vocabulary** is a limited set of diverse shapes serving as essential building blocks for the designs created using grammar sets.

Shape grammars naturally lend themselves to computer implementations: the computer handles the calculation tasks (the representation and computation of shapes, rules and grammars, and the presentation of correct design alternatives), while the designer specifies, explores, develops design languages, and selects alternatives. Surprisingly, little effort has been directed at computer implementations despite their theoretical appeal (Tapia, 1999), and even nowadays, only a few researchers are working on further developing this topic.

Within the Rhinoceros 3D/Grasshopper parametric environment, it is possible to easily define and use the shape grammar algorithms (Figure 3) without specialised software or the need for programming knowledge to use it in the modelling software efficiently.

The McNeal Rhinoceros 3D software with the VPL module plugin Grasshopper is a multiple platform (Windows, Mac) 3D modelling environment with a built-in (from version 6 onward) Visual Programming Language module, which at first glance only supports complex 3D model manipulation and generation but can actually have a much more profound impact than that (Leitão et al., 2012).

Two custom-made shape grammar plugins developed for Grasshopper are widely available for download on the plugin portal Food4rhino. The first one is RUPA, a shape grammar design assistant by Alva Sondakh (Food4Rhino1, 2019) developed for his master's thesis. The second one is SORTALGI, a shape grammar interpreter made by Rudi Stouffs (Food4Rhino2, 2019). This supports the specification and application of both parametric and non-parametric shape rules and the generation of single or multiple (in parallel or sequence) rule application results.

However, Rhinoceros 3D directly supports all three conditions defining a shape grammar: shapes, Boolean operators and Euclidean transformations. What is lacking is the ability to form a data loop, so the transformation (rule) is repeated on the resulting geometry n times. For this purpose, an additional plugin called Anemone by Mateusz Zwierzycki (Food4Rhino3, 2019) was used in this research. The fourth instalment of the plugin allows the creation of feedback loops in Grasshopper. The basic workflow relies on two main components: Loop Start and Loop End, where Loop End sends data back to Loop Start.

Hoopsnake by Yannis Chatzikonstantinou (Food4Rhino4, 2019) is another available feedback loop plugin with a similar functionality. In principle, it creates a copy of the data it receives as its input upon user request and stores it locally. This duplicate is made available through a standard Grasshopper parameter output. The component's input includes programming to escape Grasshopper's recursive loop avoidance check.

In Figures 3 and 4, an elementary principle is developed in Grasshopper with two shapes on a 2D plane. In Figure 3, the initial shape is a rectangle; in Figure 4, the shape is an irregular quadrilateral defined by four parametric points. With the Grasshopper plugin Anemone, a feedback loop is created and serves the specified rule in combination with the iteration slider. The rule states that each side of the rectangle or irregular polygon has a point assigned in the same position relative to its length. Then, from the four new points, a new polygon is created on which the rule repeats until all iterations are fulfilled. The results in the lower row demonstrate that the basic method for shaping grammar with regular modelling tools is possible. It can be easily implemented without special software or programming knowledge and can be accomplished using the VPL plugin. As part of the parametrisation of the shape grammar, a code is expressed that enables the result to be parametrically recreated if the need arises.

3. RESULTS

The research sought to demonstrate that the shape grammar method offers architecture students and professionals a tangi-

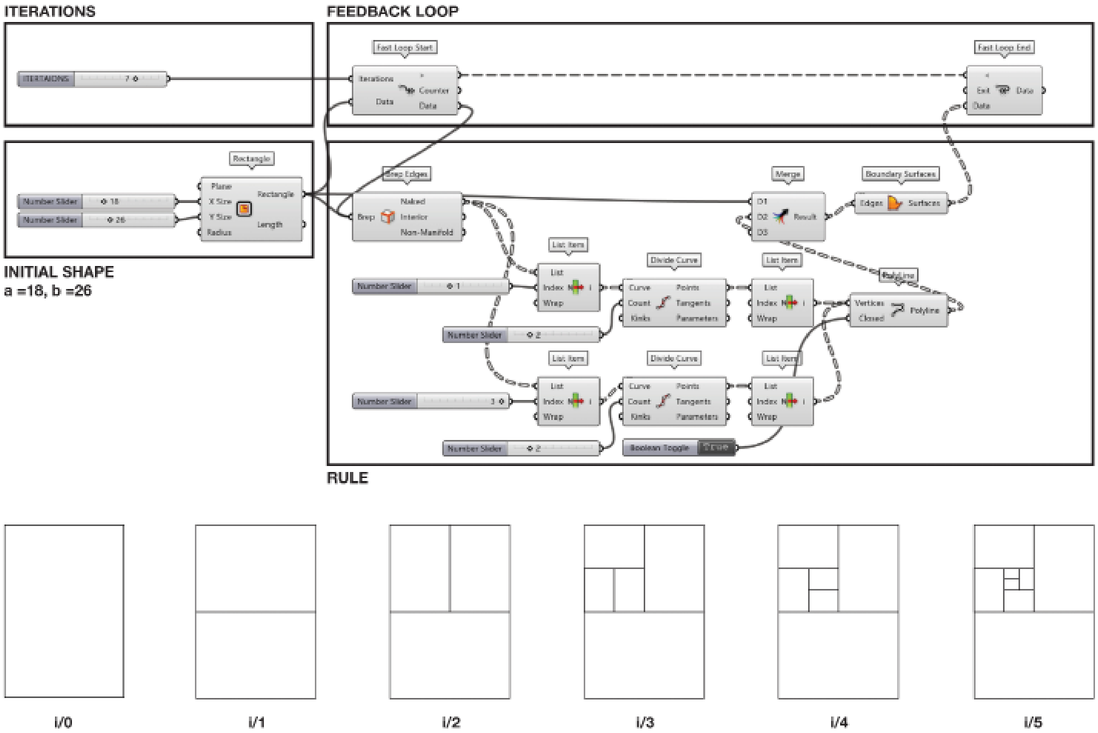


Figure 5: The Grasshopper definition of the rectangle with the recurring rule to divide the first and each subsequent rectangle in half. The number of iterations represents the number of divisions of the successive rectangles.

ble approach to applying shape grammar in architectural and urban design. This method allows them to explore various design solutions while consistently maintaining a coherent design language. To establish a strong and consistent design language, we have decisively chosen to develop the entire methodology exclusively on a 2D plane. Building on the design definition in Figure 4, a rectangle is defined, and a simple rule is applied to create a new rectangle that is exactly half the size of the initial one and to repeat that rule n times (Figure 5). The iterations can be defined dynamically. Furthermore, additional parameters concerning the division of the sides of the rectangles are defined but not used in this case.

The essential elements of shape grammar in a two-dimensional space deal with points, lines and polygons. Additional attributes, such as colours and weights, can be added to each essential element. However, weights are attributed only to lines. In the case of a visual representation of architectural drawings, they are an important tool. They can be added or mixed in various ways (Figure 6), expanding the two-dimensional grammar with line weights and creating a weight grammar.

For colours, colour grammar can be developed and added to polygons but can also be attributed to lines or points. With the introduction of overlays, an additional system can be developed to manage how colours mix and interact. This system can

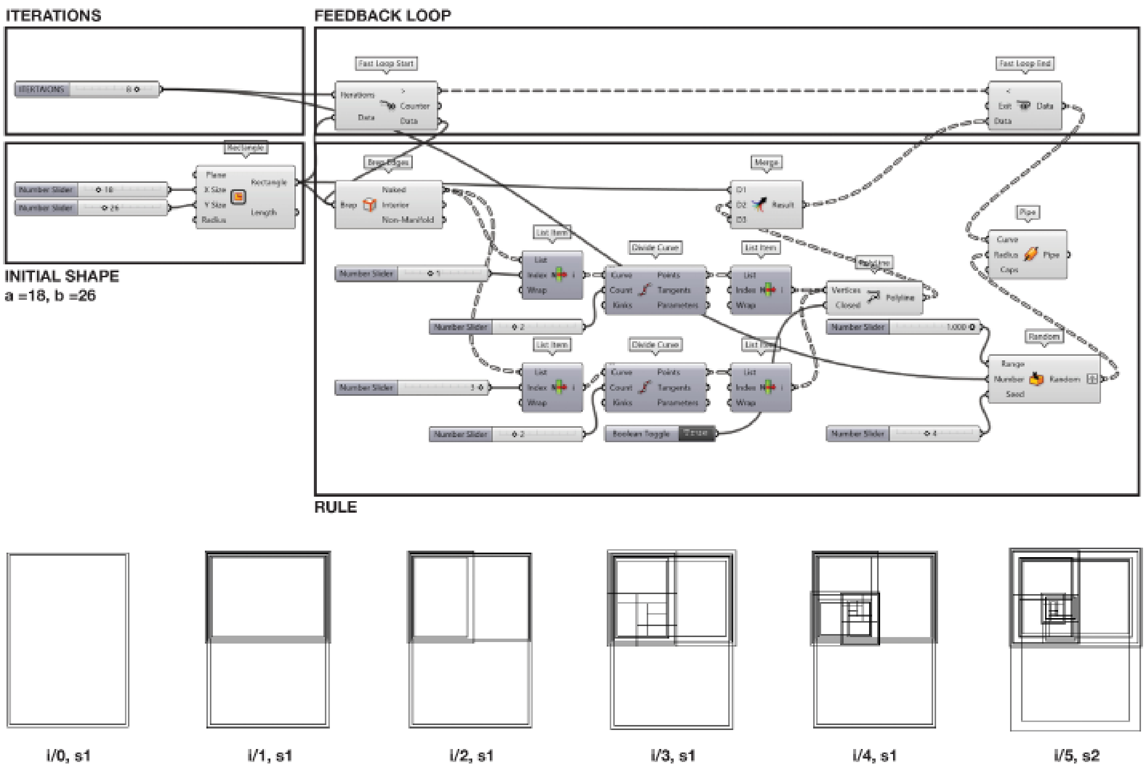


Figure 6: The Grasshopper definition based on the definition in Figure 5; the division of each subsequent iteration has an additional rule for defining the width of the lines.

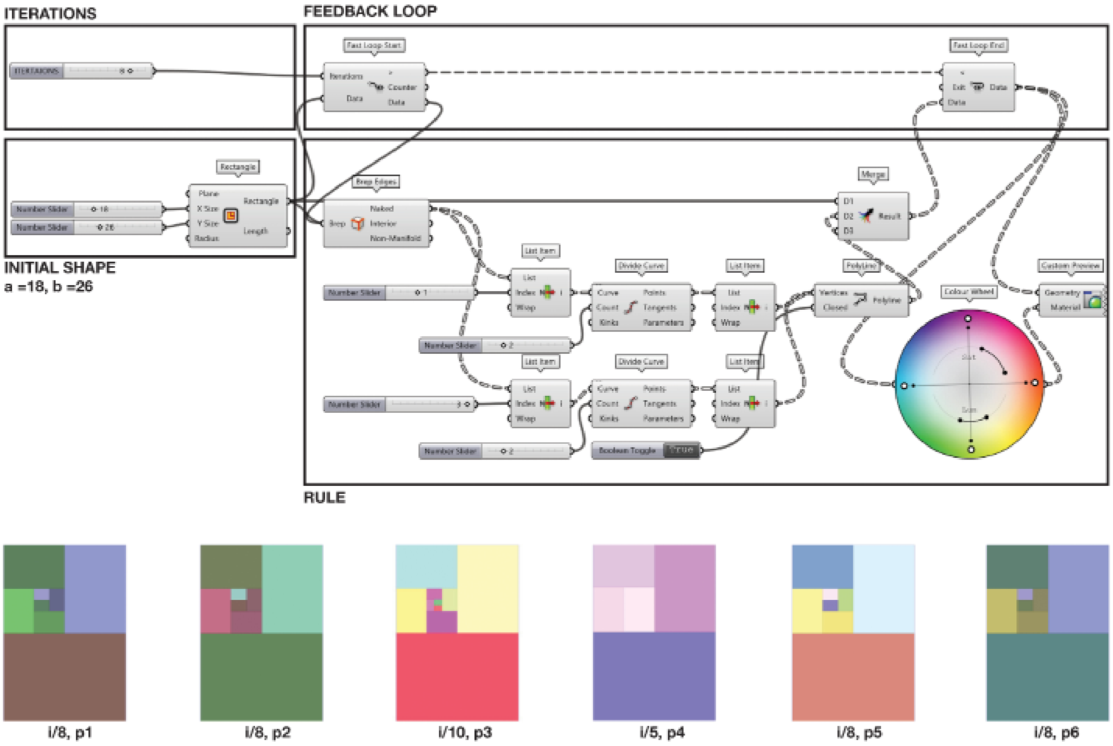


Figure 7: The Grasshopper definition based on the definition in Figure 6; the division as each subsequent iteration has an additional rule for defining the colours of the divided polygons, thus enabling the shape grammar rule to attribute a random colour to each polygon.

be based on layers, colour theory, or other predefined rules. Colour is another variable that can be attributed to the rules of composition for designs (Figure 7). The exact definition, as used in Figure 6, is expanded with the Grasshopper Random colour component, creating a simple colour grammar definition.

To research or use the design language of a painter with a design system, the painting *Curves* by Georges Vantongerloo painted in 1938 was selected. The composition of the painting seems random but is based on a previously developed system. However, other compositions can be identified if the author's design rules—and the design language they follow—have been determined. A new composition that follows the same design language can be generated with consistent implementation of design rules with a custom shape grammar. Grammatical transformation is the gradual process of changing the artist's (in this case) painting style evolution. This can be done with gradual or abrupt changes to the basic style. Ultimately, however, it gradually leads to an evolution of painting styles.

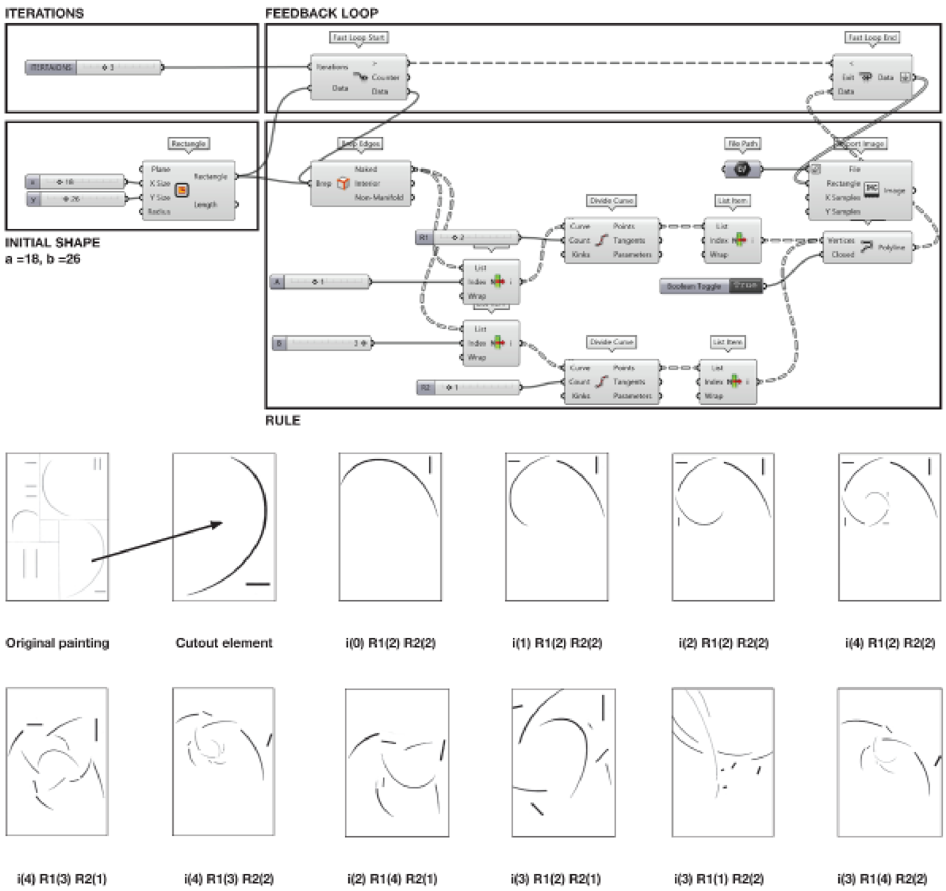


Figure 8: The final definition based on the first Grasshopper definition from Figure 5. This time, instead of line thickness or colour, an element of the painting by Georges Vantongerloo is used. Different kinds of compositions using consistent design language are derived by manipulating the attributes of the rule (how a polygon is divided) and the number of iterations.

With just a minor change of the Grasshopper definition (Figure 8), a part of the painting by Georges Vantongerloo was used to fill the emerging design. The same code was used as in all the previous examples (Figures 2–8), but new transformative parameters were used to define a more complex rule. Initially, the results in the top row of Figure 8 follow the same principle as the previous results in Figures 4–6. However, the bottom row in Figure 8 uses slightly changed parameters relating to the division of the sides of the newly created quadrilaterals. With little effort, completely new designs emerge in the context of the same design language.

4. CONCLUSIONS

There is no doubt that shape grammar has practical applications and is a fantastic tool. It is practical for showing students, professionals and others involved in the design process how a solution was generated and other possible alternatives. Shape grammar systems produce design alternatives with non-definitive formal solutions, keeping a consistent design and spatial language, which is positive. The shape grammar solutions shown in the research are easily implemented using the described software tools. The use of a VPL enables the user to create new shape grammar vocabularies or exceptional synthetic grammar with great ease.

All design professionals employ their methodologies to arrive at their solutions and only in rare cases create a single solution for a specific problem. Spatial solutions are the result of versioning and iteration, leading to the selection of the optimal alternative. There is also the question concerning the quality of the proposed designs, especially with shape grammar, which can quickly generate countless alternatives. Systems that can direct or limit the stream of alternatives that meet the defined criteria must be established. To choose the best (or a range of several) solutions, new technologies such as multiparametric decision systems (Bercic et al., 2018; 2021; Berčič et al., 2024), machine learning and artificial intelligence based on an extensive database of good practical examples and heuristic support must be implemented in the classical tools that professionals use to take full advantage of the shape grammar methodology.

However, as Kahneman (2011) explains, humans rely on instinctive, bias-prone first-tier thinking. In contrast, surprising research (Goh et al., 2024) demonstrates that by objectively evaluating criteria, computational systems are more effective at predicting the "best" outcome, allowing human expertise to intervene only after an optimised selection has been generated.

Shape grammars align seamlessly with parametric systems, as computers excel at computation and iterative tasks. The future lies in leveraging their ability to generate vast design alternatives while employing decision-based models to refine and select the most promising solutions, thus maximising efficiency and design quality.

ACKNOWLEDGEMENTS

I would like to thank Prof. José Pinto Duarte for introducing me to the shape grammar theory as part of my PhD studies at the University of Ljubljana, Faculty of Architecture, in the course Introduction to Shape Grammars that he taught at Penn State College of Arts and Architecture in the 2018 fall semester. A considerable part of the research contained in this article was part of my final report on the subject.

I gratefully acknowledge Prof. George Stiny from MIT for granting written permission to use images 43 and 51 from his 1980 work "Kindergarten Grammars: Designing With Froebel's Building Gifts" published in Environment and Planning B to create Figure 1. I appreciate Prof. Stiny's support and the opportunity to build upon his significant contributions to the field.

I also gratefully acknowledge Melissa Abendroth for granting permission to use the image of Fröbel gifts (https://www.paradise-ofchildhood.com/content/images/size/w1000/2022/10/IMG_1537.JPG) in this article. The image illustrates the concept of Fröbel gifts in the context of architectural education.

The research presented in this paper was conducted in the scope of the research program "Sustainable Planning for the Quality Living Space" and is funded by the Slovenian Research and Innovation Agency (ARIS) under grant no. P5-0068. This support is gratefully acknowledged.

REFERENCES

Ažman Momirski, L. (2018). Negotiating dynamic variables in urban regeneration process: A case study of the degraded Kranj railway station area. *Prostor*, *26*(1 (55)), 156–169. https://doi.org/10.31522/p.26.1(55).12

Beirão, J., & Duarte, J. P. (2018). Generic grammars for design domains. *AI EDAM*, *32*(2), 225–239.

Bercic, T., Bohanec, M., & Ažman Momirski, L. (2018). Role of decision models in the evaluation of spatial design solutions. *Annales: Anali za istrske in mediteranske študije – Series Historia et Sociologia*, *28*(3), 621–636.

Berčič, T. (2021). *Vrednotenje prostorskih rešitev z večparametrskimi modeli = Evaluation of spatial solutions through multiparametric models* [[T. Berčič]]. https://repozitorij.uni-lj.si/IzpisGradiva.php?id=137203

Berčič, T., Bohanec, M., & Ažman Momirski, L. (2024). Integrating multi-criteria decision models in smart urban planning: A case study of architectural and urban design competitions. *Smart Cities*, *7*(2), 786–805. https://doi.org/10.3390/smartcities7020033

Cuff, D. (2001). Digital pedagogy: An essay. *Architectural Record*, (9), 200–204.

Čok, G. (2014). Residential buildings and sustainable development in Slovenia = Stambene zgrade i održivi razvoj u Sloveniji. *Prostor – znanstveni časopis za arhitekturu i urbanizam*, *1*(47), 134–147.

Duarte, J. P. (2001). *Customising mass housing: A discursive grammar for Siza's Malagueira houses* [Doctoral dissertation, Massachusetts Institute of Technology].

Duarte, J. P. (2005). A discursive grammar for customising mass housing: The case of Siza's houses at Malagueira. *Automation in Construction*, *14*(2), 265–275.

Duarte, J. P., & Beirão, J. (2011). Towards a methodology for flexible urban design: designing with urban patterns and shape grammars. *Environment and Planning B: planning and design*, *38*(5), 879-902.

Fleisher, A. (1992). Grammatical architecture? *Environment and Planning B: Planning and Design*, *19*, 221–226.

Food4Rhino1. (2019). RUPA (Shape Grammar Design Assistant). [Online]. Available at: https://www.food4rhino.com/app/rupa-shape-grammar-design-assistant [Accessed 16 Oct. 2019].

Food4Rhino2. (2019). SortalGI Shape Grammar Interpreter. [Online]. Available at: https://www.food4rhino.com/app/sortalgi-shape-grammar-interpreter [Accessed 16 Oct. 2019].

Food4Rhino3. (2019). Anemone. [Online]. Available at: https://www.food4rhino.com/app/anemone [Accessed 16 Oct. 2019].

Food4Rhino4. (2019). Hoopsnake. [Online]. Available at: https://www.food4rhino.com/app/hoopsnake [Accessed 16 Oct. 2019].

Garcia, S. (2016). Classifications of shape grammars. In *Proceedings of the Design Computing and Cognition conference, DCC 2016* (pp. 259–278). Chicago, United States.

Garcia, S., & Menezes Leitão, A. (2018). Shape grammars as design tools: An implementation of a multipurpose chair grammar. *Artificial Intelligence for Engineering Design, Analysis and Manufacturing*, *32*(2), 240–255. https://doi.org/10.1017/s0890060417000610

Goh, E., Gallo, R., Hom, J., Strong, E., Weng, Y., Kerman, H., Cool, J. A., Kanjee, Z., Parsons, A. S., Ahuja, N., Horvitz, E., Yang, D., Milstein, A., Olson, A. P. J., Rodman, A., & Chen, J. H. (2024). Large language model influence on diagnostic reasoning: A randomized clinical trial. *JAMA Network Open*, *7*(10), e2440969.

Grasl, T., & Economou, A. (2018). From shapes to topologies and back: An introduction to a general parametric shape grammar interpreter. *Artificial Intelligence for Engineering Design, Analysis and Manufacturing*, *32*(2), 208–224. https://doi.org/10.1017/s0890060417000506

Hong, T. C. K., & Economou, A. (2023). Implementation of shape embedding in 2D CAD systems. *Automation in Construction*, *146*, 104640. https://doi.org/10.1016/j.autcon.2022.104640

Hudson, R. (2010). *Strategies for parametric design in architecture: An application of practice-led research* [Doctoral dissertation, University of Bath].

Kahneman, D. (2011). *Thinking fast and slow*. Farrar, Straus and Giroux.

Knight, T. (2000). Shape grammars in education and practice: History and prospects. *International Journal of Design Computing*, *2*(67).

Knight, T., & Stiny, G. (2001). Classical and non-classical computation. *Arq: Architectural Research Quarterly*, *5*(4), 355–372.

Koning, H., & Eisenberg, J. (1981). The language of the prairie: Frank Lloyd Wright's prairie houses. *Environment and Planning B*, *8*, 295–323.

Lauzzana, R. G., & Pocock-Williams, L. (1988). A rule system for analysis in the visual arts. *Leonardo*, *21*(4), 445–452. https://doi.org/10.2307/1578709

Leitão, A., Santos, L., & Lopes, J. (2012). Programming languages for generative design: A comparative study. *International Journal of Architectural Computing*, *10*(1), 139–162.

McKay, A., Chase, S., Shea, K., & Chau, H. H. (2012). Spatial grammar implementation: From theory to usable software. *Artificial Intelligence for Engineering Design, Analysis and Manufacturing*, *26*, 143–159. https://doi.org/10.1017/S0890060412000042

Oxman, R. (2004). Think-Maps: Teaching design thinking in design education. *Design Studies*, *25*(1), 63–91.

Oxman, R. (Ed.). (2006). Digital design [Special issue]. *Design Studies*, *27*(3).

Stiny, G. (1980a). Introduction to shape and shape grammars. *Environment and Planning B*, *7*, 343–351.

Stiny, G. (1980b). Kindergarten grammars: Designing with Froebel's building gifts. *Environment and Planning a* 125–135.

Stiny, G. (2006). *Shape: Talking about seeing and doing*. MIT Press.

Stiny, G., & Gips, J. (1972). Shape grammar and the generative specification of painting and sculpture. In C. V. Freiman (Ed.), *Information Processing 71* (pp. 1460–1465). North-Holland.

Mitchell, W., & Stiny, G. (1978). The palladian grammar. Environment and Planning B, 5, 5-18.

Stiny, G., & Mitchell, W. J. (1980). The grammar of paradise: On the generation of Mughul gardens. *Environment and Planning B*, *7*, 209–226.

Tapia, M. (1999). A visual implementation of a shape grammar system. *Environment and Planning B: Planning and Design*, *26*(1), 59–73.

Verovsek, S., Juvancic, M., & Zupancic, T. (2013). Understanding and interpreting urban space (in)formation. *International Journal of Architectural Computing*, *11*(2), 135–155. https://doi.org/10.1260/1478-0771.11.2.135