Towards Bridging Shape and Graph Grammars Through Topology

Fundamental Shape Processes for Architectural Representations in Artificial Intelligence

Theodoros Dounas[1], Wassim Jabi[2]
[1,2]Heriot Watt University [2]Cardiff University
[1]T.dounas@hw.ac.uk [2]JabiW@Cardiff.ac.uk

The emergence of transformer models has revitalized AI research in architectural design despite 3D generative AI lagging behind its 2D counterparts due to dataset limitations. Lightweight frameworks like Topologic enable rapid creation of 3D datasets suitable for graph machine learning, leveraging the library's ability to produce vector-based geometric, topologic, and semantic models. Our paper presents a topology-driven graph grammar generating architecturally meaningful 3D models that contribute to generative AI training datasets. Bridging 3d shapes, topology and graphs, our system applies shape transformation rules procedurally to create 3D models from simple graphs. This approach bridges shape grammars with topology, maintaining semantics while generating novel solutions suitable for 3D generative AI.

Keywords: *topology, graph grammars, shape grammars, generative systems.*

INTRODUCTION AND BACKGROUND

Generative systems in Architectural Design present remarkable opportunities and significant challenges for the architectural, engineering, and construction (AEC) Industry. These algorithmic approaches to design, from evolutionary computation to algorithmic and parametric design, and to machine learning models, have revolutionised our ability to explore vast design spaces and to develop complex solutions that would not be otherwise possible. Within that design space, the ability to develop design variations and evaluate them against a set of complex criteria has enabled us to invent novel solutions. However, there are several critical weaknesses within artificial intelligence based generative systems, which can get overlooked within the enthusiasm that their novelty carries. One is the "optimisation trap" where values are optimised without a clear orchestration on the human experiential quality of the architecture produced (Dounas, 2020). Another issue, increasingly evident within the users of Machine Learning (ML) models is its "black box" nature where designers have no explicit concept on how their tool works, but take the output for granted, abandoning their design processes agency to the algorithm of the machine. Further, some of the ML/Artificial intelligence (ML/AI) models require disproportionate amounts of computing power compared to the benefits they bring to the design process. In many cases, these systems rely on vast amounts of data to train the ML algorithms, which results in an overabundance of ML algorithms biased towards areas where a certain type of data is more in abundance. For example, there is a plethora of diffusion models in AEC but a lot less model development in areas such as graph machine learning, probably due to the lack of datasets for developing such models

This paper is written under the lens of maintaining control of the design process, but also in using graph data and discrete transformation processes as the main vehicle of computation. Our research question is: what are the orchestrations of shape and graphs that architectural instances need, so that more efficient graph machine learning models for architecture can be built? We focus on the objective of elegantly representing both shape and graphs and their transformations. Our motivation lies within the realms of developing simple tools that advance computation as a vehicle of the design process, in this instance using discrete elements and topology, with the scope to improve graph representations and graph machine learning in architectural design. The paper focuses on the integration of graph and shape grammars at the topologic level, developing what we call Topologic Grammars.

STATE OF THE ART - REPRESENTING BUILDING WITH GRAPH STRUCTURES.

The AEC industry faces a fundamental challenge in its digital transformation: the inadequacy of current data structures for representing buildings comprehensively. While Building Information Modelling (BIM) and Digital Twins have emerged as powerful concepts, the underlying formats, particularly Industry Foundation Classes (IFC), have significant limitations. Apart from the Bonsai plugin for Blender open-source effort, IFC functions primarily as an export format rather than a common working format, fails to capture parametric information, and is limited to building-scale representations rather than component-level descriptions. These limitations severely restrict its utility in modern applications.

The integration of Artificial Intelligence with BIM presents additional complexities, as AI systems require specific data formats optimised for their architectures. For instance, convolutional neural networks are optimised for image processing, transformers for natural language processing, and graph neural networks for network-structured data. However, current approaches often prioritise the AI technique over the inherent structure of building data, leading to forced adaptations that may not optimally represent architectural information. This misalignment is further complicated by issues of data accessibility, availability, and format standardization in the AEC sector.

A graph is a simple discrete mathematical structure comprising a set of nodes and edges, where some or all the nodes are connected by edges. Representing building data as a graph can take an either a labelled directed graph structure (LDG) or a Resource description framework (RDF) – a type of graph that uses triples (subject-predicate-object) to replicate and exchange data on the web. Both have been adopted by researchers in various domains in computation on AEC. RDF graphs are used to build knowledge graphs representing data of a building. The adoption by W3C of the Resource Description Framework (RDF) structures which can be used in linked data to present entities in the modern web3.0. is further evidence of their usability in building representation. In other examples (Alymani et al., 2021) LDGs have been used extensively to facilitate the topological representation of a building with uses in ML/AI or energy analysis, or to encode the position of discrete building components for circular economy (Dounas et al., 2021).While the linked data paradigm with the Resource Description Framework offers a promising approach by representing building information in graph format with designed ontologies, it too has limitations. The conversion process often sacrifices important information, particularly geometric data of building elements. The fundamental challenge stems from the multidimensional nature of buildings, which intersect various scientific disciplines, each with its own data requirements and conventions. Though graph-based data structures appear theoretically ideal for orchestrating this complexity, the industry still lacks a comprehensive data structure suitable for modern informatics applications (Mavrokapnidis et al., 2021).

This context frames the motivation for our paper, an exploration of a new approach combining topology frameworks, specifically Topologic (2021), with discrete component analysis and synthesis, where graph data structures are used throughout the development of a building model. Indeed, the goal is to use topology driven graph data structures as drivers for architectural synthesis. This integration promises a more rigorous classification of topological relationships in spatial and component analysis, while maintaining the advantages of graph data structures.

We demonstrate a basic shape grammar workflow through Topologic, and the development of a particular case study on a Topologic Graph Grammar of the Dom Narkomfin building, a significant 20th-century architectural work that has not previously been subjected to grammatical analysis.

Generative Representations: Shape and Graph Grammars

Shape Grammars are production systems (SCL, 2025; Stiny, 1980) that provide excellent generative description, evaluation and interpretation of existing designs (and styles) and the creation of new design (styles). (Grasl et al. 2018). A constant criticism of Shape grammars has been their lack of adoption by the computational community in automation processes, which stands in contrast with their adoption for the theoretical analysis of design styles. Eloy et al, present a set of papers that include generic interpreters, produced in Rhino/GH but also a couple of grammatical examples that attempt to generalise the generation of shape grammars, such as the one created by Beirao and Duarte (2018) on encoding design principles that are general and common to large sets of designs.

Graph Grammars have been used to represent Shape Grammars (Grasl et al., 2018) or integrated with them in hybrid production systems (Swap architekten & GRAPE) (Dounas, 2020). Grasl and Economou have provided a succinct and complete classification of techniques that can be used to map shape grammars to graph grammars. Most mapping techniques start with the shape and provide an interpretation of the shapes into LDGs. The most straightforward one is "Points to Nodes" (Figure 1) and "Lines to Edges" (Figure 2) where the resulting graph looks the same as the generating shape. This technique does not account for maximal lines, resulting in poor performance in recognising emergent shapes (Gero et al., 1993). Second is the "Lines to Nodes" and "Intersections to Edges", based on an inversion of the first one. While this approach takes care of the maximal lines issue, and allows for emergence, it is troublesome in the cases of grammars where more than two lines intersect. As such it is a good fit for grammars that result in orthogonal designs.

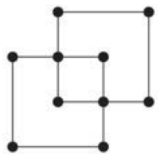

Figure 1
Points to Nodes and lines to Edges mapping

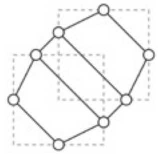

Figure 2
Lines to Nodes and Intersections to Edges

A third mapping is to translate both points and lines to nodes. This needs a label for the nodes, increasing the complexity of the graph, and the amount of data needed to store the shape, but also adds flexibility as one could attach any element to a node, from solids to spaces or colours.

A fourth method is provided by Wang et al. (2018) to create tractable designs, where the shapes are first translated using the first technique, but then a compass graphical device is used to establish rules of transformation, where each node gets assigned cardinal directions. This suggests that potential three-dimensional grammars and their transformations might benefit from a coordinate and cardinal navigation system.

Generative Representations: Topologic

Topologic is a software modelling library (2021) that supports a comprehensive conceptual framework for the hierarchical spatial representation of buildings based on non-manifold topology (NMT). It integrates geometry, topology, and semantics to enable architects and designers to reflect on their design decisions in the early stages of conceptual design, before the complexities of building information modelling (BIM) set in. Topologic aims to bridge the gap between the conceptual design phase and the BIM-based design development phase by bringing forward building performance analyses, including spatial reasoning.

Topologic's classes include Vertex, Edge, Wire, Face, Shell, Cell, CellComplex, Cluster, Topology, Graph, Aperture, Content, Context, and Dictionary. These classes act as geometric abstractions and placeholders for more detailed representations, allowing designers to think conceptually and strategically about their design.

One of the key features of Topologic is its ability to conduct formal spatial reasoning and semantics-based geometrical and topological operations. It allows users to associate semantic information with topologies through dictionaries on graphs, which can be used to affect geometric operations. For example, Boolean operations can be manipulated by a topology's semantic information, allowing users to experiment with different results and optimize their design by altering the semantics in the model. Topologic also supports graph-based analyses, such as shortest path calculations and space integration/centrality studies, which can be useful for understanding the spatial relationships within a building design.

METHODOLOGY

We developed a set of basic shape grammar operations that would allow us to work with architectural topological entities. These basic operations include splitting or segmenting a volume (horizontally or vertically), creating sub volumes and their Boolean operations with the initial volume, and transformations. Topologic automatically generates the underlying labelled directed graphs. Within this initial stage we did not yet use the graphs to drive the grammars but use them in this paper to illustrate the possibility to do so. Additionally, we tested and validated successfully the basic operations by developing a grammar for Dom Narkomfin, a building which has not been studied grammatically before.

TOPOLOGIC GRAMMARS: BASIC OPERATIONS

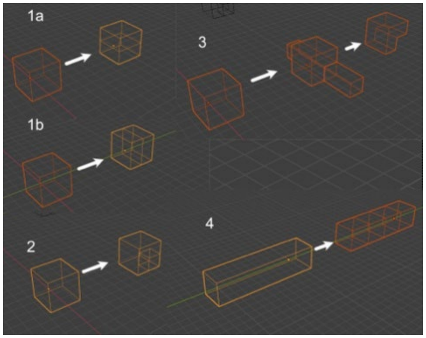

Figure 3
Basic Operations in Topologic

This section details the methodology implemented in the Python script for replacement rules that depend on evaluating the similarity of topologies, and deriving the transformation matrix needed to map them. The script is based on several functions from the TopologicPy (2021) library to perform canonical representation of 3D polyhedral, similarity checks based on largest faces and longest edges, derivation of coordinate systems for each object, computing a transformation matrix, and final vertex congruence checks. Below is a breakdown of the key functions and their underlying concepts.

Figure 4
Grammar Workflow in Topologic

The process begins by defining shape grammar replacement rules each with a Left-Hand Side (LHS) topology and a Right-Hand Sides (RHS) topology that can be spatially synchronized through an optional transformation matrix (Figure 3). Then, a design is introduced with several topologies in it. The system proceeds by looping through the shapes in the design and determining if they match any LHS topology. If matched, the algorithm returns the rules and the needed transformation matrices to map the RHS topology to the detected shape in the design (Figure 4).

The "Topology.IsSimilar" function determines whether two topologies are geometrically similar, regardless of their position and orientation in Euclidian space by comparing their features. After excluding clearly dissimilar topologies (e.g. ones with different number of vertices, edges, or faces, or compactness factors), the function begins by identifying the largest faces in each topology, which serves as the basis for deriving local coordinate systems. For each face comparison, two coordinate systems are created: one aligned with the largest face of the first topology and another with the largest face of the second topology. Obviously, the larger number of faces with equal area, the more time this process will take to complete. Within each face comparison, a third vertex is required to establish the coordinate system. Again, the longest edges are used to determine this vertex. Using permutations of directional axes, the function tests different transformations to align the two topologies. The alignment process relies on "Matrix.ByCoordinateSystems", which computes the transformation matrix required to map one coordinate system to the other. Once aligned, the function compares the transformed geometries to assess similarity.

To determine which transformation matrix permutation to use, the source topology is transformed using each matrix and the "Topology.IsVertexCongruent" function is called to determine whether the transformed topology and the target topology are vertex congruent by comparing their vertex sets. It checks if both topologies contain the same location of vertices within a given epsilon tolerance. The function uses a KD-Tree for efficient nearest-neighbour search to calculate the ratio of unmatched vertices' coordinates, rounding to a specified mantissa for precision. If the unmatched ratio is within the tolerance, the topologies are considered congruent. A simple interface has been built to showcase the selection of the rules by a human (figure 5).

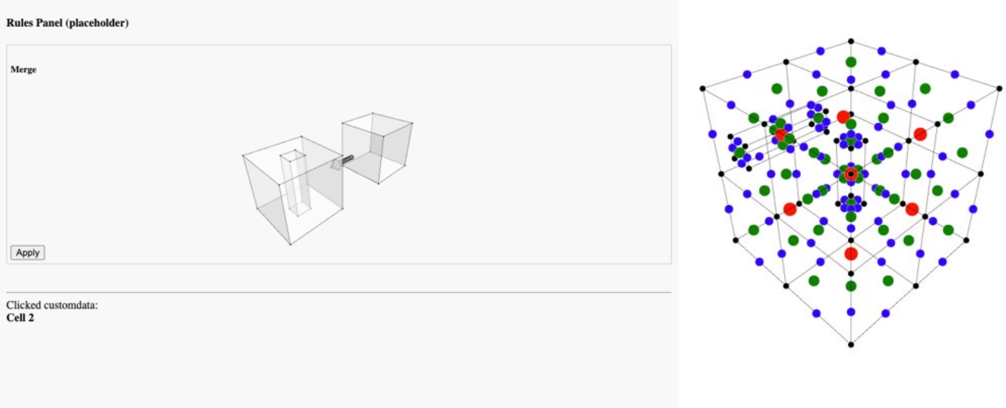

Figure 5
Simple Graphical User Interface to apply shape grammar rules in Topologic

Topologic Grammars: The Dom Narkomfin Grammar

The Narkomfin Building (Hwa-Yeon et al., 2016; Stevens et al., 2016), completed in 1930, stands as one of the most significant examples of Constructivist architecture in Moscow. Designed by architects Moisei Ginzburg and Ignaty Milinis for the Soviet Ministry of Finance (Narkomfin), this experimental communal housing project embodies the social and architectural aspirations of the early Soviet period.

Architectural Form and Structure

The building consists of two main blocks connected by an enclosed bridge-like passage. The primary residential block, elevated on pilotis (supporting columns), stretches horizontally for approximately 85 meters and rises six stories high. Its clean, modernist lines are punctuated by continuous horizontal ribbon windows that wrap around the façade, creating a striking visual rhythm and flooding the interior spaces with natural light.

The secondary communal block, shorter in height but equally modernist in design, housed shared facilities including a gymnasium, cafeteria, and library. This separation of functions between blocks reflects the building's experimental social program while creating a distinctive architectural composition. The communal block was named as a social condenser, where all social life would be housed and condensed. Several later designs in the 20th century would use the idea of the condenser as an architectonic strategy to develop their spatial arrangements.

Interior Organisation: The residential units were designed in two main types:

K-type units: Small apartments with compact kitchens and F-type units: Larger duplex apartments without kitchens, intended to encourage use of communal dining facilities.

By applying the rules developed within Topologic, we developed an abstract, topologically correct representation of the Dom Narkomfin, along with all the step-by-step graph representations of each stage of development (figure 6 and 7). Essentially, this is simple formal system allowed the development of conceptually and topologically sound architectural representations, where the advancement of the shape rules dictate the changes in the graphs. We have not yet tested the mirror technique which would be driving the shape grammar through extending the graph grammar. Topologic was used through its python bindings in TopologicPy, while Blender 4.3 was used for visualisation.

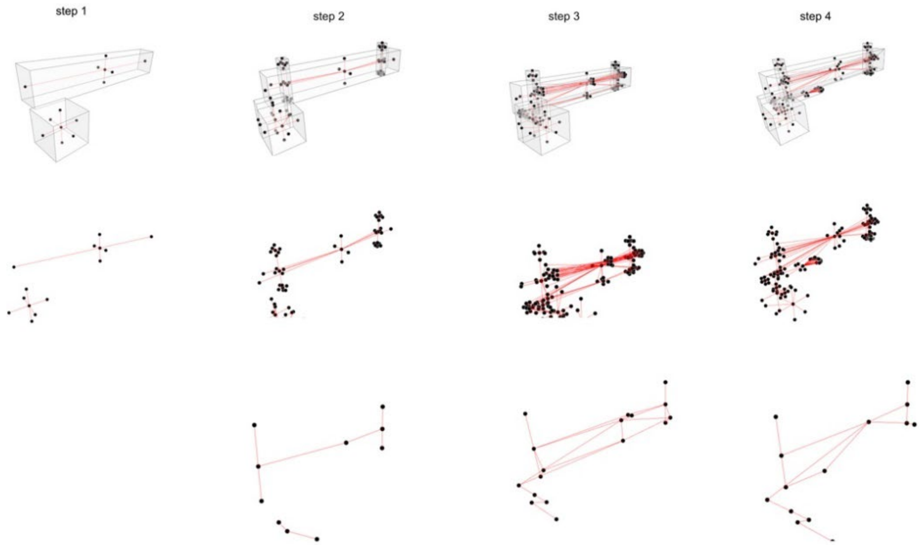

Figure 6
The development of the Dom Narkomfin Grammar, step by step with extracted graph (through their shared topologies) and, the simpler, direct graphs

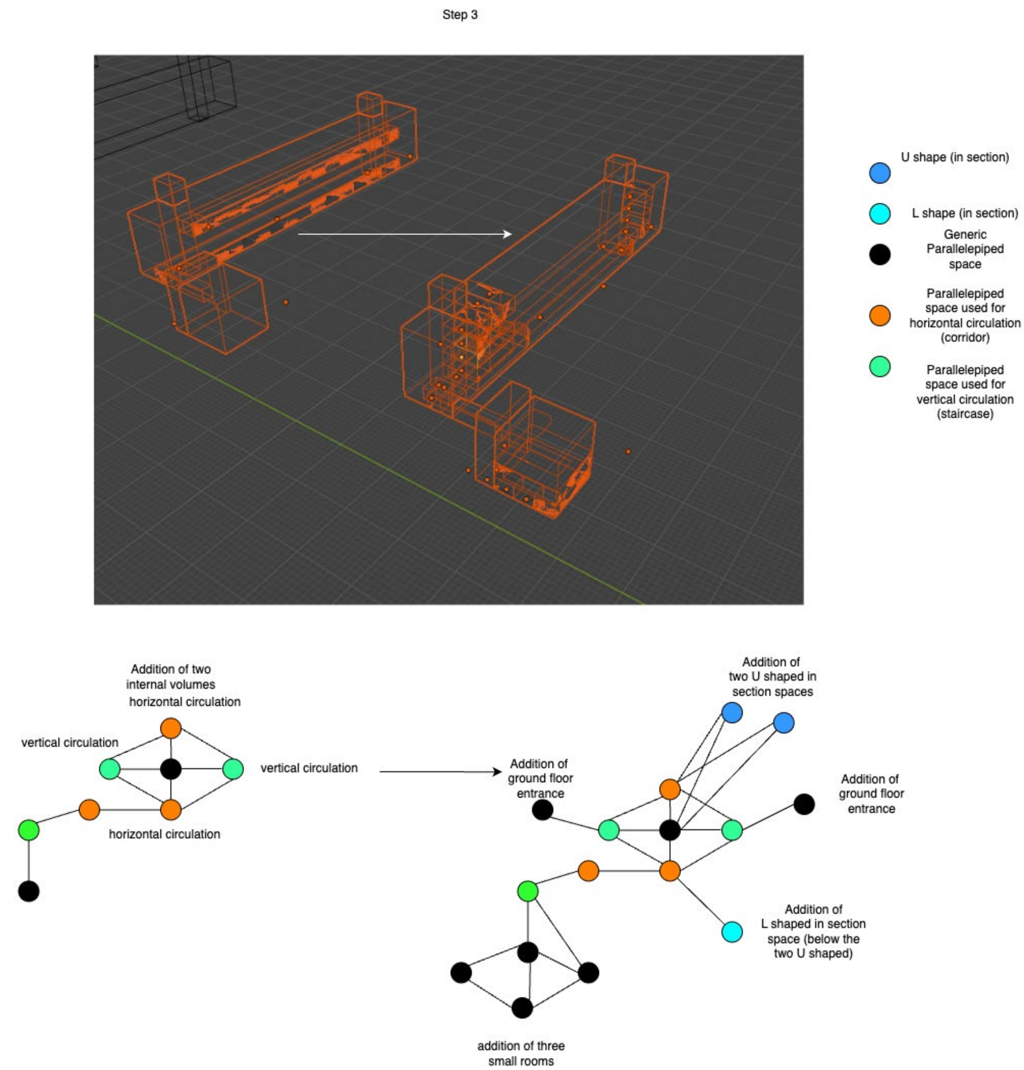

Figure 7
Step 3 in the developed Dom Narkomfin Topologic Grammar

RESULTS, LIMITATIONS AND DISCUSSION

We have shown the development of a shape and graph grammar in abstract conceptual architectural designs, that due to their inherent Topology, develop in an integrated manner. A shape grammar class was developed for Topologic, where the selection of the rule and its application is user driven. We then validated the use of the grammar using the Dom Narkomfin case study, to develop a conceptual abstract yet topologically correct model. The Dom Narkomfin Grammar follows a top-down, volumetric approach, where the initial two volumes are segmented or further subdivided using the basic operations to arrive at the end result. This approach does not allow for the development of bottom-up grammars, where operations develop from constituent, discrete parts towards a whole. An interesting further test would be the assembly of the Dom Narkomfin using such a grammar through its basic constituent parts. Additionally, through the process of developing the design of the building, a series of graphs is developed that can be used as the backbone of semantic data, through dictionaries. This means that for example while developing the emergent shapes one could assign functions, colour, materials to them, but also link them to other graphs and linked data further enriching the model. This can be useful not only in Artificial Intelligence applications but also towards an integrated method to develop Digital Twins from Design data. Topologic's capacity to generate graphs for conceptual models, but also dual graphs where both the spaces and the discrete architectonic elements are represented promises that we can build even more elaborate grammars based on this functionality. Even with this simple grammar one can see that simple conceptual models can generate a multitude of variants in architecture, taking advantage of the clarity and robustness of the replacement rules, while not sacrificing at all the understanding of how the model and its variants have been generated. These generated models can then create much larger data pools out of which one can train machine learning algorithms in full 3d or can provide the backbone of a data for the creation of robust digital twins, using linked data. The novelty of our method is that it always works in three dimensions, and that it takes into account the conceptual topology of the architecture, a consequence of Topologic's robust Topology classification.

CONCLUSIONS

Shape Grammar implementations via graph grammars have generated a variety of grammar interpreters that have a good chance of developing into generative systems. We believe that our approach of developing a grammar based on topology, that can operate inherently in three dimensions and integrate graph and shape grammars at the topology level, has the potential to provide in generative systems the needed architectural qualities, beyond the optimisation trap, while being able to explain the generation of the architectonic solutions, beyond the boundaries of "black boxes". We believe thatThese new grammatical additions to Topologic's approach will be of worthy use to other researchers in building complex, data-driven solutions for the Architecture-Engineering-Construction industry.

REFERENCES

Alymani, A. and Jabi, W. (2024) 'A Graph-Based Computational Tool for Retrieving Architectural Precedents of Building and Ground Relationship (BGR Tool)', *International Journal of Architectural Computing*, 23(2), pp. 517-532 Available at: https://doi.org/10.1177/14780771241260853

Beirão, J. and Duarte, J. P. (2018) 'Generic grammars for design domains', *Artificial Intelligence for Engineering Design, Analysis and Manufacturing*, 32(2), pp. 225-239. Available at: https://doi.org/10.1017/S0890060417000452

Dounas, T. (2020) 'Animation as a computational framework for architectural design composition', *Architectural Science Review*, 63(2), pp. 222–232. Available at: https://doi.org/10.1080/00038628.2019.170903 7

Eloy, S., Pauwels, P. and Economou, A. (2018) 'AI EDAM Special Issue: Advances in Implemented Shape Grammars: Solutions and Applications', *Artificial Intelligence for Engineering Design, Analysis and Manufacturing*, 32(2), pp. 131-137. Available at: https://doi.org/10.1017/S0890060417000634

Gero, J. S., Yan, M. (1993) 'Discovering Emergent Shapes Using a Data-Driven Symbolic Model', in *CAAD Futures '93 Conference Proceedings*. Pittsburgh, USA, pp. 3-17.

SWAP Architekten ZT. (2023) *Sort Machines*. Available at: http://plus.swap-zt.com/publikation/sort-machines/ (Accessed 4 May 2024).

GRAPE For Net - Shape Grammar Interpreter (2023) Available at: http://grape.swap-zt.com/ (Accessed 4 May 2024).

Grasl, T. and 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), pp. 208-224. Available at: https://doi.org/10.1017/S0890060417000506

Grasl, T. and Economou, A. (2011) 'GRAPE: Using Graph Grammars to Implement Shape Grammars', in SpringSim '11, Boston, MA, USA, 3-7 April 2011. pp. 21-28. Available at: https://dl. acm .org/doi/10.5555/2048536.2048539

Mavrokapnidis, D., Katsigarakis, K., Pauwels, P., Petrova, E., Korolija, I. and Rovas, D. (2021) 'A Linked-Data Paradigm for the Integration of Static and Dynamic Building Data in Digital Twins', in *Proceedings of the 8th ACM International Conference on Systems for Energy-Efficient Buildings, Cities, and Transportation*, Coimbra, Portugal, pp. 369-372. Available at: https://doi.org/10.1145/3486611.3491125.

Seo, H. and Sohn, S. (2016) 'A Study on The Relationship of Architectural Ideology and Characteristics of Spatial Composition between Narkomfin Communal House and Unite d'Habitation', *Journal of the Korean Housing Association*, 27, pp. 37-44. Available at: https://doi.org/10.6107/JKHA.2016.27.5.037.

Shape Computation Lab - Georgia Tech. (2023) *Shape Machine Symposium.* Available at: https://shape.design.gatech.edu/Symposium/index.html (Accessed 4 May 2023).

Stevens, R., Petermans, A. and Vanrie, J. (2016) 'Bit by Byt towards a Contemporary Conjoint Interior: From Dom Narkomfin to a New Social Condenser', *Interiors: Design/Architecture/Culture*, 7(2-3), pp. 135-154. Available at: https://doi.org/10.1080/20419112.2016.1201908

Stiny, G. (1980) 'Introduction to Shape and Shape Grammars', *Environment and Planning B: Planning and Design*, 7(3), pp. 343-351. Available at: https://doi.org/10.1068/b070343.

Wang, X., Liu, Y., Li, J. and Zhang, K. (2018) 'Generating Tractable Designs by Transforming Shape Grammars to Graph Grammars', in *Proceedings of the 11th International Symposium on Visual Information Communication and Interaction (VINCI '18)*, pp. 41-48. Available at: https://doi.org/10.1145/3231622.3231637

Wassim, J. (2021) *Topologicpy: An Advanced Spatial Modelling and Analysis Software Library for Architecture, Engineering, and Construction* Available at: http://doi.org/10.5281/zenodo.11555172.