Proceedings of IASS-IWSS 2026 – September 14-18, 2026, Turin, Italy
"R-Evolution of Shapes: Sustainability, Re-Use, and New Design Paradigms"
Annual Symposium of the International Association for Shell and Spatial Structures
3rd Italian Workshop on Shell and Spatial Structures
A. Manuello Bertetto, S. Gabriele, F. Marmo, A. Micheletti (eds.)

Can AI Co-Design Structures?
An AI Agent for Early-Stage Structural Design

Ricardo MAIA AVELINO* [a,b], Mennatallah EL-ASSADY [b], Tom VAN MELE [a], Philippe BLOCK [a]

*Stefano-Franscini-Platz 1, HIB E 45, Zurich 8093, Switzerland
mricardo@ethz.ch

[a] Block Research Group, Department of Architecture, ETH Zurich
[b] IVIA Lab, Department of Computer Science, ETH Zurich

Abstract

Recent advances in large language models and agentic workflows offer a new paradigm to early-stage structural design translating design intent into structural models. Although text-to-3D approaches have demonstrated that natural language can be used to produce visual, mesh-based geometric models, structural design requires domain-specific geometric arrangements and semantically rich data types. This paper introduces a proof-of-concept workflow for prompt-to-structure generation using an AI agent connected to a COMPAS-based structural modelling API through a Model Context Protocol (MCP) server. Through controlled modelling operations, the agent creates and modifies a graph-based structural data model composed of explicit building elements. The resulting artifacts can be inspected, modified, serialised, and further connected to downstream AEC workflows. The approach is demonstrated through selected prompt-to-structure examples and by progressive and editable design sequences that resemble typical iterative structural design workflows. The results are preliminary, but demonstrate the possibility of combining natural language, agentic workflows, and domain-specific structural data structures as a first step toward AI-assisted structural co-design.

Keywords: Structural Design, Artificial Intelligence, AI agents, Co-design, Human-Computer Interaction

1. Introduction

Structural design is not only a process of satisfying performance constraints but also a creative discipline in which form, force, materials, and construction logic are negotiated together. Torroja emphasized the role of intuition and imagination in the conception of structural form [1], while Billington described structural art as the integration of efficiency, economy, and elegance [2]. From this perspective, early-stage structural design cannot be reduced to the automatic generation of final solutions, but rather is an iterative process in which designers progressively develop their final result.

Previous work on computational structural design and engineering education has focused on making this exploration more creative and interactive by introducing geometry-based approaches to structural understanding [3, 4, 5, 6], developing computational tools to explore families of structural forms [7, 8, 9] and performing topology exploration rather than just looking for a single optimal design [10, 11, 12]. Further developing this line of work, text-enhanced structural design systems have introduced language as an interface for form-finding and conceptual exploration [13], including graph- and transformer-based algorithms [14]. In this context, recent developments in large language models (LLMs) and agentic workflows offer a new paradigm to early-stage structural design where AI systems can better understand design intent, create and edit 3D artifacts towards a *co-design* process.

Recent text-to-3D systems have made substantial progress in generating visual 3D assets represented as neural fields, meshes, or learned 3D latents [15, 16, 17]. More recent work in LLM-assisted modelling

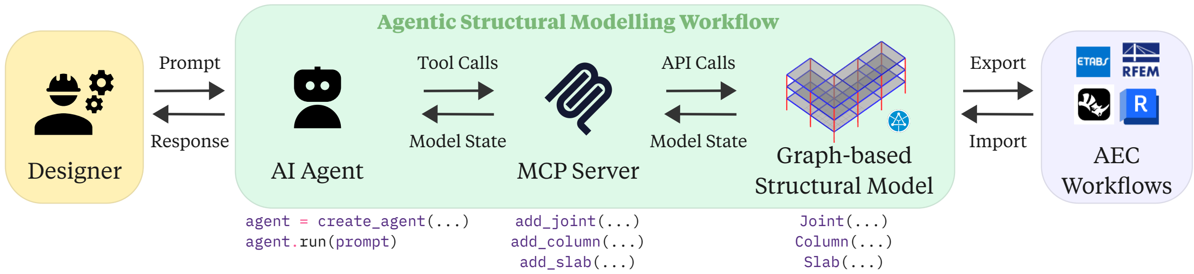


Figure 1: Agentic structural modelling workflow. A designer provides a natural-language prompt to an AI agent, which translates the prompt to MCP tool calls. The calls generate or modify a Graph-based Structural Model. The resulting artifact can be inspected, modified and further exported into AEC workflows

has begun to explore the generation of 3D CAD geometries and 2D FEA analysis from written scripts [18, 19, 20, 21, 22]. These developments suggest that language can become an increasingly powerful interface for geometry generation. However, structural design requires more than a generic mesh-based structural geometry as they must contain semantically rich objects, including joints, frames and areas, connectivity information, and domain-specific metadata.

Building on the need to develop interactive design processes and on the emerging capabilities and potential of LLMs and agentic systems, this work proposes a workflow that connects a language-model agent to a domain-specific structural modelling API through a Model Context Protocol (MCP) server. The agent interprets natural-language design intent and requests modelling actions that generate or edit a COMPAS-based graph data structure [23] that contains explicit structural objects such as joints, frames, slabs, walls, grids and levels. This object-based representation allows the generated artifact to be inspected, modified, serialized, and further connected to downstream AEC workflows.

The contribution of this paper is a proof-of-concept workflow for early-stage structural modelling with language agents. It introduces a COMPAS-based structural data structure for representing generated buildings as editable graph-based artifacts, and it demonstrates the approach through selected prompt-generated examples, a language-based editing sequence, and a progressive modelling example. By generating explicit structural data that can be inspected, modified, and connected to downstream workflows, this work paves the way to further research on interactive natural-language-based interfaces for structural design.

2. An Agentic Structural Modelling Workflow

Figure 1 summarizes the proposed workflow. A designer provides a natural-language prompt describing a desired structural model or modification. A specialised AI agent interprets this request and communicates with a structural modelling environment through a Model Context Protocol (MCP) server. The server acts as an interface between natural-language reasoning and deterministic modelling actions. These actions generate or modify a 3D structural model. The resulting model can then be reviewed and further adjusted using follow-up instructions. The individual components of this workflow are detailed below.

2.1. A COMPAS-based graph data structure

The central component of the workflow is the domain-specific data structure. The generated artifact is represented as a COMPAS [23] graph-based model which defines the main elements of the system, their connectivity and key meta-data associated with them. These elements are semantically rich and directly relate to the definitions used in structural engineering practice.

The structural API exposes a small set of Python classes corresponding to common building-scale structural elements and enable operations on them such as additions, deletions, copy, translations, and rotations. The API enables the creation of `Joints()` as points in the 3D space, `Frame()` connecting two joints, and `Area()` elements connecting 3 or more points. The `Frame()` object is further specialised into `Beam()`, `Column()` and `Brace()` to represent horizontal, vertical, and inclined elements in the structure. The `Area()` object is further specialised in `Wall()` and `Slab()` elements. Classes for storing metadata assigned to these elements are also present in the API, including `Material()` and `Section()`. Geometry constructors are also present and allow for the definition of `Grid()` and `Level()` as well as to operate geometrically on these entities.

These objects are assembled in a `Structure()` class, which stores a graph-based representation of the structural model that enables direct access to adjacent entities. In this graph, the nodes correspond to the joints, whereas the frame and area elements encode relationships among those nodes. The API facilitates common operations in building-scale modelling, including creating and modifying regular grids, adding and replicating stories, and copying and dragging objects. This API is exposed to the AI agent through an MCP server, so the operations are executed through function calls as defined below.

2.2. MCP as a bridge between AI agent and structural model

Recently, the Model Context Protocol (MCP) was introduced [24] as an open protocol for connecting AI assistants to external tools, data sources, and services. In AEC, recent work has begun to use MCP-like interfaces to allow LLMs to query, generate, and modify IFC or BIM models [25, 26] and generate OpenSees and ETABS structural analysis models [27].

In the present work, an MCP server is created to expose the COMPAS-based structural modelling API to the agent. When triggered by the AI agent, the MCP functions execute deterministic operations and return the updated state of the model. These controlled actions populate and modify a graph-based structural data structure. As a result, an explicit 3D structural artifact is produced that can then be inspected visually and modified through follow-up instructions. The AI agent is pre-configured to use the MCP definitions to build the structural artifacts and relies on Google Gemini 3.1 Pro [28] for reasoning on the user input.

2.3. Import and Export

The generated structures are stored as explicit COMPAS objects whose data can be serialised and stored as a structured JSON file. Based on the COMPAS ecosystem interoperability features, the geometric data underlying the model can be imported and exported to common geometry formats. The development of different connectors to common AEC software is planned as future work such that the generated models could be directly transformed into AEC software models typically used in structural practice.

The next section demonstrates the use of the presented workflow to generate, edit, and progressively design complex structural building artifacts that can serve engineers and architects working on early-stage structural design.

3. Prompt-to-Structure Demonstrations

3.1. Generating 3D Structural Models from Natural Language

Figure 2 presents five selected examples of prompt-generated structural models that show the potential of the current workflow to generate a range of structural building models. The examples move from simple to more complex artifacts. P01 demonstrates the basic creation of columns, beams, and slabs from span and height information. P02 introduces a circular plan, repeated perimeter columns, and a core wall. P03 uses a notional grid and removes bays to form an L-shaped plan. P04 extends the task to a terraced multi-story structure with different floor extents across levels. P05 introduces geometric rotating stories along the height of the building. Together, these examples suggest that the agent can generate models that mix different structural arrangements and geometric operations. Furthermore, the agent can handle domain-specific vocabulary used to describe building-scale models.

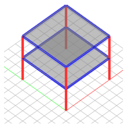
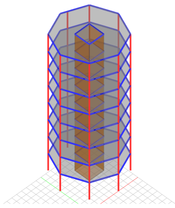
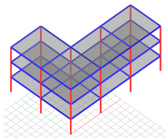
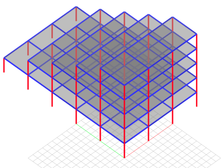
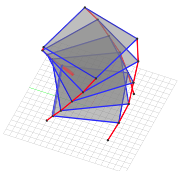
	3D Model	Prompt
P01		Draw a 2-story building with one bay in x and one bay in y. Use spans of 6.0m in both x and y directions and a story height of 3.0m. Add columns at all corners and beams along the perimeter at each level. Add a slab at each floor level.
P02		Draw a circular tower with 8 stories above ground, each 3.5m high. The base of the tower has a circular layout with radius 6.0m. Place 8 perimeter columns evenly spaced on the circle. Add beams connecting adjacent perimeter columns at each level and add a circular slab at each floor. Create a central square core wall of side length 3.0m running from the base to the top.
P03		Draw a 3-story building with an L-shaped plan. The building should have a notional 3x2 bays with 6m spacing being 18m long in x and 12 long in y. The two top-right bays should be removed to create the L-shaped plan. Add slabs, beams and columns the gridlines to support the slabs.
P04		Draw a 5-story terraced building. The base footprint is a 5 by 3 bay grid with 5.0m spacing in both directions and story height 3.0m. Each higher story steps back by one bay from the west side, while the east facade remains vertical.
P05		Draw a tower with 5 rotating stories. The tower should rotate 15 degrees at each story. The tower can be simplified by a square floorplan with 10m span and columns at the edges. Please add the slabs at each floor and beams in the perimeter.

Figure 2: Selected prompt-to-structure examples. P01 shows a simple two-story one-bay frame; P02 introduces a circular tower with a central core; P03 generates an L-shaped footprint; P04 introduces a multi-story building with setbacks; and P05 shows a rotating tower. Each model is generated from a direct natural-language prompt and stored as an editable structural artifact.

3.2. Editing object-based structural models

Figure 3 illustrates how a generated artifact can be modified through follow-up language instructions. Starting from the circular tower in P02, the user asks for an extension to the first three floors. The AI agent first queries about the current state of the model, reasons about the required addition or modification, and the maps the MCP tools that need to be called to achieve it. For this example, the agent responds to the user prompt by adding beams and columns creating the extension as requested. The user then follows up with another modification, requesting the addition of slabs on the newly created floors. The updated version adds the slab elements to the designated floors.

This example reflects a common situation in early-stage design, where new ideas or design modifications are proposed to an already existing structure, which then need to be reflected in an updated model. By allowing for this interaction, the workflow caters to the practical needs of structural designers as they adapt their models towards the final design solution.

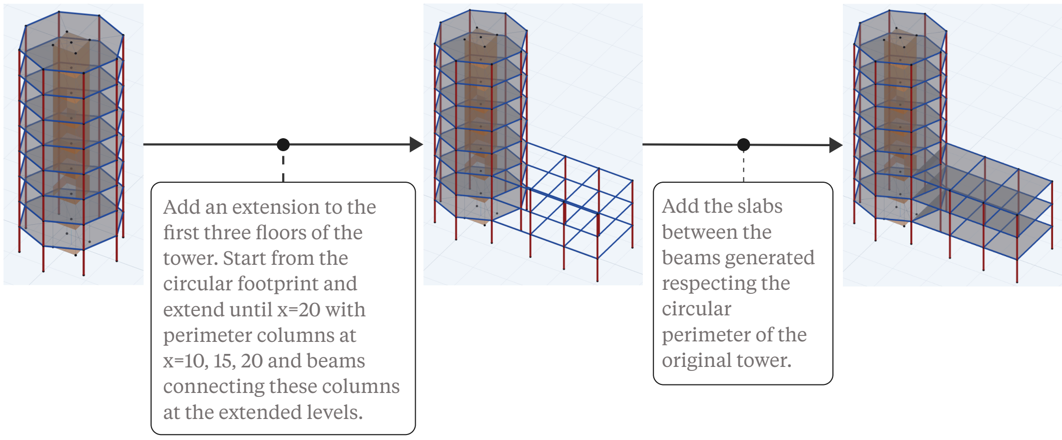


Figure 3: The circular tower generated by prompt P02 has the lower floors extended through a follow-up natural-language instruction.

The example demonstrates that object-level modifications can be requested and executed on an existing model, but also reveals limitations in geometric interpretation. In Figure 3, the agent was unable to correctly draw the slab adjacent to the initial circular footprint as requested, resulting in an unnecessary overlap between the added rectangular slab and the existing central tower slabs. Further prompt instructions or manual edits would then be needed to update this particular location as intended. Geometric errors can indeed occur due to current LLMs' limited spatial reasoning or to ambiguity on the user prompting. This limitation highlights the need for manual inspection and verification.

3.3. Towards progressive structural co-design

Figure 4 presents a second sequence in which the designer deliberately starts with a simple artifact and increases its complexity over several prompts. The initial request creates two towers. A subsequent prompt connects the towers through an elevated skybridge, and a later prompt adds inclined cables to support the bridge. This sequence illustrates a different mode of use in which the user does not provide the complete design brief at once, but develops the artifact progressively. Such progressive modelling resembles early structural design practice, where designers often begin with a simple organizational idea and even generate simplified preliminary models that then become more complex over time.

By exposing the AI agent to the model state at all times, this progressive design is possible. This example connects the current workflow to the broader idea of AI-assisted structural co-design where the AI agent is able to rapidly materialize intermediate structural artifacts, allowing the designer to inspect, revise, and redirect the model. This generation and edition of the models is a prerequisite for future structural co-design workflows.

By reducing the effort required to create preliminary structural models and removing the friction of structural modelling, such workflows can support teams to achieve higher-performing design alternatives. The opportunities and limitations of the findings presented here are discussed in the next section.

4. Opportunities and Limitations

The workflow presented suggests several opportunities for early-stage structural design. First, prompt-based model generation can reduce the friction associated with repetitive model setup, allowing designers to test structural ideas more quickly. By lowering the effort required to create preliminary models, the workflow can enable greater design freedom and unlock more exploratory forms of creativity. Second, the use of explicit structural elements and access to the current state of the model using MCP servers make such that the resulting artifact can be inspected, modified and exported. Third, future AI-assisted

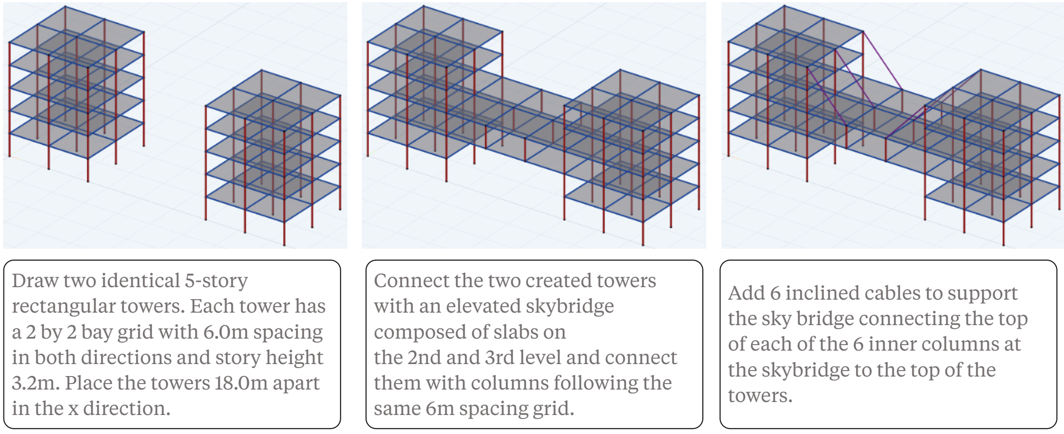


Figure 4: Progressive language-based structural modelling. A designer progressively complexifies the building through a sequence of prompts: two towers are generated, then connected by a skybridge which is later supported with inclined cable elements.

modelling could support the design of more resource-efficient or sustainable structures by guiding the user on modifications, additions, or deletions that can have a positive outcome on the structure metrics, including material consumption and embodied carbon accounts.

The approach may also be extended through multimodal input. Structural designers and architects do not communicate through text alone: sketches, annotations, reference images, or preliminary models are also important [29]. Future systems could therefore combine these multimodal inputs to bring human and AI understanding closer during model interaction.

The current prototype also has important limitations. The examples shown in this paper are proof-of-concept rather than a formal evaluation. The generated models do not guaranty structural correctness, code compliance, or analysis readiness. Prompt ambiguity, hallucination, misinterpretation of design intent and robustness across similar prompts still needs to be assessed. Furthermore, current LLMs are known to have limited spatial reasoning [30], such that geometric modifications can be wrongly executed by agents. The size and complexity of the models studied here are also limited. These limitations motivate the need for further research and development of benchmarks, datasets, and validation protocols for AI-assisted structural modelling.

5. Conclusions

Building on existing research that formalizes structural design as an exploratory and creative discipline, this paper presented a proof-of-concept workflow to generate editable early-stage structural building models from natural-language prompts. The workflow presented combines language-agent reasoning with a COMPAS-based structural modelling API exposed through an MCP server. Through selected examples, the paper shows that language agents can create editable structural artifacts that can assist in early structural design workflows. This work argues that LLMs introduce a new design paradigm in structural design and make it possible to establish new modes of interaction between human designers and structural models. The contribution is preliminary and a series of limitations exist regarding the size of the models, analysis readiness, and the risk of intent misinterpretation. Nevertheless, the results suggest that structural design is a relevant and challenging domain for future AI-assisted modelling workflows, where the goal is not autonomous design but the reduction of friction between design intent and editable structural models towards more creative and efficient designs.

Acknowledgements

This research was supported by the Design++ Postdoctoral Fellowship awarded to the first author in collaboration with Halter AG.

References

[1] E. Torroja, J. J. Polivka and M. Polivka, *Philosophy of Structures*. University of California Press, Dec. 1958.

[2] D. P. Billington, *The Tower and the Bridge: The New Art of Structural Engineering*. Princeton, NJ: Princeton University Press, 1983.

[3] J. Teng, C. Y. Song and X. F. Yuan, "Fostering creativity in students in the teaching of structural analysis", *International Journal of Engineering Education*, vol. 20, no. 1, pp. 96–102, 2004.

[4] E. Allen, W. Zalewski and J. Iano, *Form and Forces: Designing Efficient, Expressive Structures*. Hoboken: John Wiley & Sons, 2010.

[5] T. Van Mele, L. Lachauer, M. Rippmann and P. Block, "Geometry-Based Understanding of Structures", *Journal of the International Association for Shell and Spatial Structures*, vol. 53, no. 174, pp. 285–295, 2012.

[6] R. Maia Avelino, J. Lee, T. Van Mele and P. Block, "An Interactive Implementation of Algebraic Graphic Statics for Geometry-Based Teaching and Design of Structures", in *International Fib Symposium - Conceptual Design of Structures 2021*, pp. 447–454, (Attisholz-Areal, Switzerland), Sept. 2021.

[7] P. O. Ohlbrock and J. Schwartz, "Combinatorial Equilibrium Modeling", *International Journal of Space Structures*, vol. 31, no. 2–4, pp. 177–189, 2016. doi: `10.1177/0266351116660799`.

[8] J. Lee, C. Mueller and C. Fivet, "Automatic Generation of Diverse Equilibrium Structures through Shape Grammars and Graphic Statics", *International Journal of Space Structures*, vol. 31, no. 2–4, pp. 147–164, 2016. doi: `10.1177/0266351116660798`.

[9] K. Saldana Ochoa, P. O. Ohlbrock, P. D'Acunto and V. Moosavi, "Beyond Typologies, Beyond Optimization: Exploring Novel Structural Forms at the Interface of Human and Machine Intelligence", *International Journal of Architectural Computing*, vol. 19, no. 3, pp. 277–299, 2021. doi: `10.1177/1478077120943062`.

[10] C. T. Mueller, *Computational Exploration of the Structural Design Space*. PhD thesis, Cambridge, 2014.

[11] R. Oval, M. Rippmann, R. Mesnil, T. Van Mele, O. Baverel and P. Block, "Feature-Based Topology Finding of Patterns for Shell Structures", *Automation in Construction*, vol. 103, pp. 185–201, July 2019. doi: `10.1016/j.autcon.2019.02.008`.

[12] R. Danhaive and C. T. Mueller, "Design subspace learning: Structural design space exploration using performance-conditioned generative modeling", *Automation in Construction*, vol. 127, p. 103664, 2021. doi: `10.1016/j.autcon.2021.103664`.

[13] Z. Guo, K. Saldana Ochoa and P. D'Acunto, "Enhancing Structural Form-Finding through a Text-Based AI Engine Coupled with Computational Graphic Statics", in *Proceedings of IASS Annual Symposia*, vol. 2022, pp. 1–11.

[14] L. Bleker, Z. Guo, K. Smith, K.-M. M. Tam, K. S. Ochoa and P. D'Acunto, "Text2Structure3D: Graph-based generative modeling of equilibrium structures with diffusion transformers", *Results in Engineering*, vol. 31, p. 111375, 2026. doi: `10.1016/j.rineng.2026.111375`.

[15] B. Poole, A. Jain, J. T. Barron and B. Mildenhall, "DreamFusion: Text-to-3D Using 2D Diffusion", in *The Eleventh International Conference on Learning Representations*, Sept. 2022. doi: 10.48550/arXiv.2209.14988.

[16] C.-H. Lin, J. Gao, L. Tang, T. Takikawa, X. Zeng, X. Huang, K. Kreis, S. Fidler, M.-Y. Liu and T.-Y. Lin, "Magic3D: High-Resolution Text-to-3D Content Creation", in *2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 300–309, June 2023. doi: 10.1109/CVPR52729.2023.00037.

[17] J. Xiang, Z. Lv, S. Xu, Y. Deng, R. Wang, B. Zhang, D. Chen, X. Tong and J. Yang, "Structured 3D Latents for Scalable and Versatile 3D Generation", *arXiv preprint arXiv:2412.01506*. doi: 10.48550/arXiv.2412.01506.

[18] Y. Du, S. Chen, W. Zan, P. Li, M. Wang, D. Song, B. Li, Y. Hu and B. Wang, "BlenderLLM: Training Large Language Models for Computer-Aided Design with Self-Improvement", *arXiv preprint arXiv:2412.14203*, Dec. 2024. doi: 10.48550/arXiv.2412.14203.

[19] S. Lu, G. Chen, N. A. Dinh, I. Lang, A. Holtzman and R. Hanocka, "LL3M: Large Language 3D Modelers", *arXiv preprint arXiv:2508.08228*, Aug. 2025. doi: 10.48550/arXiv.2508.08228.

[20] X. Li, Y. Sun and Z. Sha, "LLM4CAD: Multimodal Large Language Models for Three-Dimensional Computer-Aided Design Generation", *Journal of Computing and Information Science in Engineering*, vol. 25, no. 2, p. 021005, 2025. doi: 10.1115/1.4067085.

[21] M. S. Khan, S. Sinha, T. U. Sheikh, D. Stricker, S. A. Ali and M. Z. Afzal, "Text2CAD: Generating Sequential CAD Designs from Beginner-to-Expert Level Text Prompts", in *Proceedings of the 38th International Conference on Neural Information Processing Systems*, vol. 37 of *NIPS '24*, pp. 7552–7579, (Red Hook, NY, USA), Curran Associates Inc., Dec. 2024. doi: 10.5555/3737916.3738158.

[22] H. Liang, M. T. Kalaleh and Q. Mei, "Integrating Large Language Models for Automated Structural Analysis", *arXiv preprint arXiv:2504.09754*, Apr. 2025. doi: 10.48550/arXiv.2504.09754.

[23] T. Van Mele, "COMPAS: A Framework for Computational Research in Architecture and Structures", https://compas.dev, 2017. Accessed on 2026-05-28.

[24] Anthropic, "Introducing the Model Context Protocol", https://www.anthropic.com/news/model-context-protocol, 2024. Accessed on 2026-05-28.

[25] B. K. Nithyanantham, T. Sesterhenn, A. Nedungadi, S. Peral Garijo, J. Zenkner, C. Bartelt and S. L"udtke, "MCP4IFC: IFC-Based Building Design Using Large Language Models", *arXiv preprint arXiv:2511.05533*, 2025. doi: 10.48550/arXiv.2511.05533.

[26] T. Heimig-Elschner, C. Du, A. Scheuvens, A. Borrmann and J. Beetz, "A Modular Reference Architecture for MCP-Servers Enabling Agentic BIM Interaction", *arXiv preprint arXiv:2601.00809*, 2026. doi: 10.48550/arXiv.2601.00809.

[27] C. Avila, D. Ilbay and D. Rivera, "Human–AI Teaming in Structural Analysis: A Model Context Protocol Approach for Explainable and Accurate Generative AI", *Buildings*, vol. 15, p. 3190, Jan. 2025. doi: 10.3390/buildings15173190.

[28] Google, "Gemini 3.1 Pro - Model Card", https://deepmind.google/models/model-cards/gemini-3-1-pro/. Accessed on 2026-05-28.

[29] R. Maia Avelino, R. Sevastjanova, T. Van Mele, P. Block and M. El-Assady, "Creativity

from Friction: Human-AI Interaction for Exploratory Structural Design", *arXiv preprint arXiv:2607.07521*, 2026. doi: `10.48550/arXiv.2607.07521`.

[30] J. Yang, S. Yang, A. W. Gupta, R. Han, L. Fei-Fei and S. Xie, "Thinking in Space: How Multimodal Large Language Models See, Remember, and Recall Spaces", in *2025 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 10632–10643, June 2025. doi: `10.1109/CVPR52734.2025.00994`.