

ModuFold: Semantic Authoring of Origamic Architecture

<p>Guhn Lee*</p> <p>Industrial Design</p> <p>KAIST</p> <p>Daejeon, Republic of Korea</p> <p>leeguhn@kaist.ac.kr</p>	<p>Jaewon You*</p> <p>Industrial Design</p> <p>KAIST</p> <p>Daejeon, Republic of Korea</p> <p>ampersand328@kaist.ac.kr</p>	<p>Haeri Kim</p> <p>Industrial Design</p> <p>KAIST</p> <p>Daejeon, Republic of Korea</p> <p>kimhaeri@kaist.ac.kr</p>	<p>Andrea Bianchi</p> <p>Industrial Design</p> <p>KAIST</p> <p>Daejeon, Republic of Korea</p> <p>andrea@kaist.ac.kr</p>
---	--	--	---

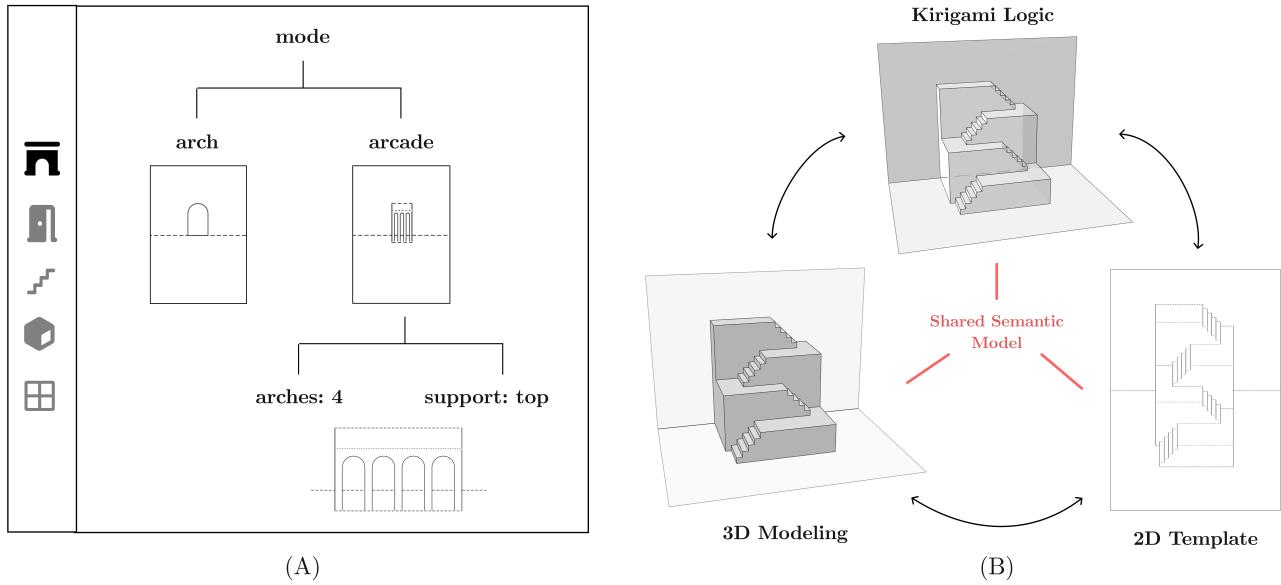


Figure 1: MODUFOLD supports a continuous workflow from design to fabrication: (A) compose architectural designs from parameterized modules, (B) inspect and refine them across synchronized 3D, deployed kirigami, and 2D template views, and (C) fabricate the resulting design as a physical model. A shared semantic model keeps edits synchronized across representations.

Abstract

Designing origamic architecture requires coordinating a three-dimensional form with its deployed panel structure and planar fabrication pattern. We present MODUFOLD, an authoring environment that connects these representations through a shared semantic design model. Makers compose parameterized architectural modules using familiar 3D interactions, while supported edits in the conventional modeling, deployed kirigami, and template views update the same underlying design. A Rhino-based geometry engine converts supported compositions into panels, hinges, openings, and an A4 cut-and-fold pattern. By preserving module identities and parameters across this process, MODUFOLD supports a continuous workflow from architectural construction to physical fabrication.

*These authors contributed equally to this research.

CCS Concepts

• **Human-centered computing** → **Interactive systems and tools**; *Systems and tools for interaction design*; • **Applied computing** → *Computer-aided design*.

Keywords

origamic architecture; kirigami; semantic authoring; computational fabrication; cross-representational editing

ACM Reference Format:

Guhn Lee, Jaewon You, Haeri Kim, and Andrea Bianchi. 2026. ModuFold: Semantic Authoring of Origamic Architecture. In *Adjunct Proceedings of the 11th ACM Symposium on Computational Fabrication (SCF Adjunct '26)*, December 06–08, 2026, Tokyo, Japan. ACM, New York, NY, USA, 3 pages. <https://doi.org/10.1145/3828643.3858022>

1 Introduction

Origamic architecture is a form of kirigami that creates three-dimensional architectural forms from a cut-and-folded sheet of paper [Chatani 1985]. Its geometry has been studied [Castle et al. 2014; Choi et al. 2019] and applied to deployable and shape-changing structures [Chang et al. 2020; Grønbæk et al. 2020; Yang et al. 2025]. Designing it, however, requires reasoning across multiple spatial



This work is licensed under a Creative Commons Attribution 4.0 International License. *SCF Adjunct '26, Tokyo, Japan*

© 2026 Copyright held by the owner/author(s).
ACM ISBN 979-8-4007-2822-8/2026/12
<https://doi.org/10.1145/3828643.3858022>

representations: conceiving a building as a three-dimensional form, understanding how it resolves into a deployed arrangement of panels, and expressing that transformation on a flat sheet through cuts and folds. These representations must remain compatible for the sheet to rise into the intended structure when opened.

Existing computational tools support this process from several directions. Simulation frameworks model how specified cut-and-fold structures move, collide, and lock during deployment [Jiang and Choi 2026]. Interactive editors help users construct geometrically valid forms and patterns [Mitani and Suzuki 2004], while conversion methods translate completed 3D CAD models into realizable paper structures [Fujikawa and Ijiri 2023; Li et al. 2010]. However, these workflows primarily expose faces, folding components, and cut lines as their authoring units. As a design moves between representations, its architectural intent becomes distributed across geometric parts, making higher-level parameters difficult to carry through subsequent revisions.

We present MODUFOLD, an authoring environment that coordinates construction, kirigami deployment, and planar fabrication through a shared semantic design model. Makers combine parameterized staircases, roofs, arcades, doors, and windows, using them additively to build structure or subtractively to create openings. The conventional modeling view supports spatial composition, the deployed kirigami view reveals how the design resolves into panels and folds, and the template view allows supported edits on the fabrication sheet. All three act on the same module records, keeping changes synchronized across three-dimensional form, folded behavior, and two-dimensional fabrication.

Table 1: Supported architectural modules and parameters.

Module	Editable parameters
Basic volume	Width, depth, height
Staircase	Step count, width, landing, tread/rise scale
Stepped roof	Step count, span, rise
Arch/Arcade	Proportions, bay count, pier thickness, support placement
Openings	Type, width, height, window panel spacing

2 System Implementation and Workflow

MODUFOLD provides a nonlinear authoring workflow across conventional 3D modeling, deployed kirigami, and planar template views. Users can move between these representations to compose architectural modules, inspect their folded behavior, and make supported edits. Each edit updates the shared design and other representations, while the resulting template can be printed and fabricated.

MODUFOLD consists of a TypeScript and Three.js¹ authoring interface connected to a Rhino 8² geometry backend. The interface maintains the parameterized module records and sends their box and cut descriptions to a local HTTP adapter, which invokes a RhinoCommon³ Python engine through Rhino MCP. The backend returns deployed panels, hinge lines, panel-source associations, and a flattened A4 pattern.

¹<https://threejs.org/>

²<https://www.rhino3d.com/8/>

³<https://developer.rhino3d.com/api/rhinocommon/>

The engine converts supported compositions of orthogonal solids into 90-degree pop-up structures. It first unites the authored solids, then extracts faces along two ordered construction directions and extends them to the corresponding fold planes. Connected coplanar regions are merged into panels, while boundaries shared by perpendicular panels form hinges. These relationships guide panel flattening, fold direction, and positioning at the selected deployment angle. Rectangular and arched openings are applied to panels of the host sheet through planar Boolean subtraction, then incorporated into the A4 cut-and-fold pattern.

Users construct designs from parameterized architectural modules that the interface translates into the solids and cuts accepted by Rhino. A staircase or stepped roof remains one editable module while being backed by an ordered set of boxes. An arcade consists of a constructive envelope and repeated arched cuts, while doors, arches, and windows become semantic openings. The expanded solids are assigned source identifiers, allowing generated panels to be associated with their contributing geometry and resolved back to the originating module. Table 1 summarizes the current module vocabulary and its editable parameters.

The modeling view operates directly on the modules, while source associations and semantic proxies connect deployed panels and flattened regions to their originating elements. Where source ownership is unambiguous, users can select, translate, and resize modules from the deployed kirigami and template views. These interactions modify the same underlying records, and the other representations regenerate from the latest design state.

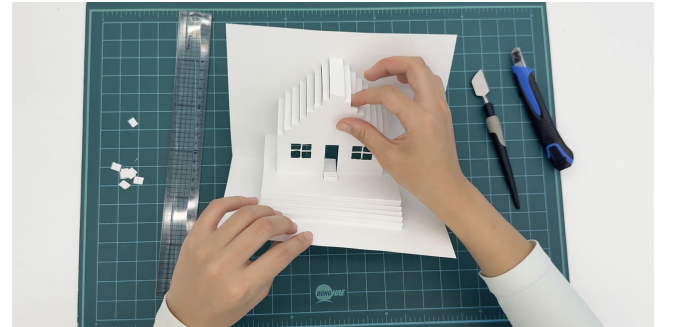


Figure 2: A physically fabricated example of an origamic architecture designed with ModuFold. Users can cut and fold along the template guide or utilize a plotter machine.

3 Conclusion and Future Work

We introduce MODUFOLD, a semantic authoring environment that connects conventional modeling, deployed kirigami, and planar fabrication via persistent, parameterized building modules. The current system is limited to orthogonal geometry and a 90-degree host fold, with its expressive range bounded by the available parameters. Future work will support additional fold angles, geometric configurations, and module variants, as well as image-guided authoring that extracts reviewable architectural modules and spatial relationships from building photographs. Together, these extensions would broaden the path from architectural references and spatial composition to editable fabrication instructions.

Acknowledgments

This work was supported by the IITP (Institute of Information & Communications Technology Planning & Evaluation) - ITRC (Information Technology Research Center) grant funded by the Korea government (Ministry of Science and ICT) (IITP-2026-RS-2024-00436398).

OpenAI GPT-5.6 Sol and GPT-5.6 Luna were used to develop and refine the frontend interface and its communication layer to the in-house Rhino geometry engine. The Rhino engine and its geometry-processing logic were developed by the authors without AI assistance. The tools were used to proofread and improve the clarity of the manuscript. The authors checked, tested, and revised all AI-assisted material and remain responsible for the final content, code, and reported work.

References

- Toen Castle, Yigil Cho, Xingting Gong, Euiyeon Jung, Daniel M. Sussman, Shu Yang, and Randall D. Kamien. 2014. Making the Cut: Lattice Kirigami Rules. *Physical Review Letters* 113, 24 (2014), 245502. doi:10.1103/PhysRevLett.113.245502
- Zekun Chang, Tung D. Ta, Koya Narumi, Heeju Kim, Fuminori Okuya, Dongchi Li, Kunihiro Kato, Jie Qi, Yoshinobu Miyamoto, Kazuya Saito, and Yoshihiro Kawahara. 2020. Kirigami Haptic Swatches: Design Methods for Cut-and-Fold Haptic Feedback Mechanisms. In *Proceedings of the 2020 CHI Conference on Human Factors in Computing Systems*. ACM, New York, NY, USA, 1–12. doi:10.1145/3313831.3376655
- Masahiro Chatani. 1985. *Pop-Up Origamic Architecture*. Ondorisha Publishers, Tokyo, Japan. <https://search.worldcat.org/title/12618172> OCLC 12618172.
- Gary P. T. Choi, Levi H. Dudte, and L. Mahadevan. 2019. Programming shape using kirigami tessellations. *Nature Materials* 18, 9 (2019), 999–1004. doi:10.1038/s41563-019-0452-y
- J. Fujikawa and T. Ijiri. 2023. Fabricatable 90° Pop-ups: Interactive Transformation of a 3D Model into a Pop-up Structure. *Computer Graphics Forum* 42, 7 (2023), e14954. arXiv:<https://onlinelibrary.wiley.com/doi/pdf/10.1111/cgf.14954> doi:10.1111/cgf.14954
- Jens Emil Grønbaek, Majken Kirkegaard Rasmussen, Kim Halskov, and Marianne Graves Petersen. 2020. KirigamiTable: Designing for Proxemic Transitions with a Shape-Changing Tabletop. In *Proceedings of the 2020 CHI Conference on Human Factors in Computing Systems* (Honolulu, HI, USA) (CHI '20). Association for Computing Machinery, New York, NY, USA, 1–15. doi:10.1145/3313831.3376834
- Qinghai Jiang and Gary P.T. Choi. 2026. PyKirigami: An interactive python simulator for Kirigami structures. *Computer Physics Communications* 328 (Nov. 2026), 110349. doi:10.1016/j.cpc.2026.110349
- Xian-Ying Li, Chao-Hui Shen, Shi-Sheng Huang, Tao Ju, and Shi-Min Hu. 2010. PopUp: automatic paper architectures from 3D models. *ACM Trans. Graph.* 29, 4, Article 111 (July 2010), 9 pages. doi:10.1145/1778765.1778848
- J. Mitani and H. Suzuki. 2004. Computer aided design for Origamic Architecture models with polygonal representation. In *Proceedings Computer Graphics International, 2004*. 93–99. doi:10.1109/CGI.2004.1309197
- Yue Yang, Chuang Chen, Yulu Chen, Ge Gao, Lei Ren, Bolan Yao, Lingyun Sun, Ye Tao, and Guanyun Wang. 2025. KirinInflate: Fabricating Cross-Scale Inflatables with Large Contraction and Tunable Stretchability for Tangible Interaction. In *Proceedings of the 38th Annual ACM Symposium on User Interface Software and Technology (UIST '25)*. Association for Computing Machinery, New York, NY, USA, Article 99, 15 pages. doi:10.1145/3746059.3747766