

Turning environmental design simulation algorithms into parametric objects: Experimental study of BIM-compatible dynamic blocks

Michal Krížo^{1*} 

¹ Slovak University of Technology, Institute of Ecological and Experimental Architecture, Bratislava, Slovakia

*Corresponding author

E-mail: michal.krizo@stuba.sk

Article information

Sent: Feb 13, 2026

Accepted: Mar 1, 2026

Published: Mar 13, 2026

Abstract: European building legislation increasingly emphasises improved indoor comfort, higher energy efficiency, and systematic reduction of CO₂ emissions across the building life cycle. In response, environmental simulations are becoming integral to architectural design workflows, particularly within the Building Information Modelling (BIM) environments. However, simulation processes in parametric platforms such as Rhino 3D and Grasshopper typically remain dependent on active script files, limiting interoperability, scalability, and accessibility for non-expert users. This study proposes and experimentally validates a workflow that transforms Ladybug Tools-based Grasshopper environmental simulations into BIM-compatible parametric objects using VisualARQ GH-Styles. Instead of managing simulations as external or file-based scripts, the approach encapsulates simulation logic within reusable object-based elements embedded directly in the modelling environment. The restructuring of a Ladybug Incident Radiation script into a VA GH-Styles definition is documented and compared with the standard Grasshopper workflow. Quantitative evaluation across multiple simulation sensor grid sizes shows convergence of both approaches at high resolution, with deviation below one per cent, while the object-based workflow exhibits reduced sensitivity to grid variation. In addition, the proposed method removes dependence on an active Grasshopper window and simplifies variant management within the modelling environment. While the study is limited to a single case and specific software ecosystem, it demonstrates the feasibility of the proposed workflow for early-stage environmental analysis. The novelty of the study lies in demonstrating environmental simulation as a parametric BIM object rather than as a standalone script, contributing a functional interoperability strategy for early-stage environmental design integration.

Keywords: environmental design, climate-data, sustainable architecture, Rhino, VisualARQ, Grasshopper Styles, Ladybug Tools

1. INTRODUCTION

The current objective of European building legislation is to address greenhouse gas emissions from buildings across their entire life cycle, not only during operation but also before and after use. The EU's 2050 vision for a decarbonised building stock therefore extends beyond operational emissions and calls for life-cycle-based assessment, starting with new constructions. Measures must also account for climatic conditions, climate change adaptation, local factors, indoor environmental quality, and cost-effectiveness. According to official EU statistics, buildings are responsible for approximately half of the Union's primary fine particulate matter (PM_{2.5}) emissions, contributing to premature death and illness. Improving building energy performance can therefore simultaneously reduce pollutant emissions (EUR-Lex, 2024).

Environmental analyses at early design stages are critical, as initial architectural decisions strongly influence future building performance (Bernett et al., 2021). During these stages, design concepts take shape, establishing constraints and parameters that significantly affect subsequent development (Shibeika and

Oliveira, 2024). Integrating design models with simulation data throughout both the design phase and operational life enables digital twin-based assessment of building energy performance across the life cycle (Zech et al., 2025). To evaluate required performance parameters, an increasing number of digital tools and methodologies are being developed. Many are interconnected through Building Information Modelling (BIM). BIM supports the creation and management of digital information assets (i.e., digital twins) throughout the building life cycle, from planning and design to construction and operation. It offers opportunities to enhance occupant well-being, energy efficiency, flexibility, and resilience (Urbietta et al., 2023). Based on BIM models, further analyses such as Life Cycle Assessment (LCA) and Building Energy Simulation/Modelling (BES/BEM) can be conducted (Yeung et al., 2023).

2. BIM COMPATIBLE ENVIRONMENTAL SIMULATIONS: IMPORTANCE OF INTEROPERABILITY

To design buildings that meet energy demands and user comfort requirements, architects need accessible environmental data and intuitive tools that can be applied from the earliest design stages.

A recent review and comparative analysis of digital tools used within the circular economy framework identifies Rhinoceros 3D (Rhino)/Grasshopper (GH) and Revit/Dynamo as among the most widely adopted platforms in practice. The study highlights the importance of connecting scripts directly to CAD/BIM models, enabling the evaluation of multiple performance indicators and supporting comprehensive assessment of design alternatives (Dervishaj and Gudmundsson, 2024). Rhino/GH/Ladybug (LB) is also examined in a feasibility study on Building Integrated Photovoltaics (BIPV) envelopes for retrofit projects. The authors emphasise the importance of integrating CAD-based 3D modelling with simulation tools, noting that coupling parametric design with photovoltaic simulation is a key criterion in software selection (de Sousa Freitas et al., 2020).

Similarly, a recent cross-tool investigation of BIPV simulation tools identifies data exchange and interoperability as primary areas for future research and development. The authors argue that smoother workflows and multidisciplinary integration would be facilitated within an IFC-compatible environment (Yang et al., 2024). Comparable conclusions are reported in a review of BIM-integrated computational fluid dynamics (CFD) tools—an analytical framework closely related to environmental simulation. The study highlights gaps in coupling frameworks, efficient data transfer systems, and clear guidelines for BIM compatibility and integration (Zheng et al., 2023). To address these limitations, this study investigates current constraints within the Rhino/GH/LB workflow and proposes, tests, and evaluates a transferable design of a BIM-compatible “simulation-as-a-parametric-object” workflow.

3. RHINO/GRASSHOPPER/LADYBUG BASED WORKFLOW ORGANISATION AND ITS LIMITATIONS

Rhinoceros 3D (Rhino), developed by McNeel and first released in 1998, (McNeel, 2024) is widely used in architectural practice due to its flexibility and open plug-in ecosystem. This openness has enabled broad adoption across design studios of different scales, supporting computational design processes throughout all project stages. Rhino’s integration with Grasshopper (GH), a visual programming environment and a benchmark tool for parametric design (Aguilar-Carrasco et al., 2023), further expands its capabilities by enabling generative and algorithmic workflows, (Stewart, 2024) including access to the Ladybug (LB) toolset.

Ladybug Tools is a free, open-source parametric plug-in providing environmental analysis capabilities. It builds upon validated simulation engines, including Radiance, EnergyPlus/OpenStudio, Therm/Window, and OpenFOAM. Using these engines, Ladybug evaluates façade irradiation by combining geometric and environmental parameters. (Öngören and Sözer, 2026) Its radiation simulation accuracy has been validated through comparisons with field measurements in multiple studies, (Ou and Lin, 2023; Nicholson et al., 2024) reporting relative errors below $\pm 10\%$ even in unfavourable scenarios (Aguilar-Carrasco et al., 2023).

The question, therefore, is not the reliability of the simulation engines, but how to use them more effectively and how to strengthen interoperability across users, teams, and platforms. This study focuses on workflow interoperability rather than simulation engine development. Accordingly, the primary limitations of the GH/LB workflow relate to Grasshopper file structure and script management. The prototyping-oriented nature of the GH interface often results in complex and brittle scripts, which can be difficult to share, maintain, and scale—particularly for non-expert users or larger teams (Davis et al., 2011). To clarify these challenges, the following section analyses GH-based workflow management and its structural limitations.

4. LIMITATIONS OF ACTIVE GRASSHOPPER FILE-BASED WORKFLOWS

The basic principle of GH graphical algorithm development follows a left-to-right logic. Each component receives input from the left and produces output on the right (Mode Lab, 2015). To improve readability and usability, components can be custom-named and supplemented with multiline descriptions. Alignment, grouping, and scribbles can further enhance visual organisation and distinguish different parts of the algorithm. These visual strategies, however, do not affect the workflow or output and are not mandatory (Abdulwahab, 2024).

There is no strict limit to the number of connected GH components. As a result, hundreds or even thousands of interconnected elements may occupy a single canvas. Such complexity often leads to so-called “spaghetti” definitions that are difficult to manage (Payne, 2022). To reduce this complexity, functional parts of the code can be turned into reusable containers known as clusters. A selected set of components can be transformed into a single component while preserving input and output connections. Clusters are editable and duplicable, maintaining identical content across instances until intentionally detached, i.e. “disentangled”.

Functionally, clusters resemble Rhino geometry blocks. The key difference lies in their behaviour: clusters remain dynamic and responsive to inputs, whereas Rhino blocks represent fixed geometry. Native Rhino does not currently support dynamic blocks, that is, parametric elements reacting to input values or geometry in a manner comparable to AutoCAD blocks or Revit families (Chelaru et al., 2025).

All organisational strategies described above can be implemented within a single GH file. However, for complex projects it is common practice to distribute logic across multiple files to simplify debugging and reduce computational load. While several GH files can be opened simultaneously, only one can be active at a time. Built-in navigation tabs allow switching between documents but do not support parallel processing. Consequently, when working with separate GH files, for example one calculating a solar envelope and another evaluating passive solar gain, it is not possible to interact with both in real time. Inputs cannot be modified and results previewed simultaneously across files.

To export and detach results from an active GH script, two primary methods exist: “Bake” or the Content Cache component (Davidson, 2024). Baking is the traditional approach, accessible via right-click for any component containing data. The user can define layer, name, grouping, and additional parameters in Rhino. Once baked, the geometry is fully detached and no longer parametrically editable.

The Content Cache component, introduced in Rhino 8, provides a more advanced mechanism for pulling, pushing, baking, or purging geometry and data between GH and Rhino. It allows geometry in Rhino to remain linked to the GH script and updated when necessary. However, neither method eliminates dependence on the GH environment. The Grasshopper Player enables execution of a single script without opening the GH interface, but its functionality is limited and unsuitable for complex workflows.

In many cases, a more desirable approach would treat Grasshopper as a computational engine powering parametric objects independently of a specific open GH file. Such a system would rely on a background library of GH definitions rather than active file management. This principle underlies so-called Grasshopper Styles, implemented in VisualARQ, a Rhino BIM plug-in for the AEC industry.

5. VISUALARQ: AEC-FOCUSED RHINO/GH BIM PLUG-IN

Rhino 3D supports the plug-in VisualARQ (VA), which includes built-in IFC import and export, enabling the exchange of Rhino geometries and semantic data in IFC format (Khan et al., 2023; Dervishaj and Gudmundsson, 2024). VA version 3 introduces Grasshopper Styles (GH Styles), a functionality that wraps Grasshopper definitions into objects with dynamic block-like behaviour.

The absence of native dynamic block functionality in Rhino is widely recognised within the user community. A proposed solution, termed Constraints, was investigated as a system of geometric rules such as line lengths or angular relationships that maintain consistency during manipulation (Kennedy, 2022). Although developed for Rhino 8 WIP (Work-in-progress, a testable beta version of a future Rhino release), this feature was later postponed. In comparison, VA GH-Styles offer broader potential, as they encapsulate complete Grasshopper algorithms rather than operating solely on geometric relationships between curves.

Official VA examples demonstrate GH Styles as the backbone of custom architectural elements such as walls, columns, beams, and windows (Asuni, 2026). However, their application need not be limited to geometry generation. GH Styles may also serve as containers for environmental simulation workflows processed through Ladybug components.

6. ENCAPSULATING GRASSHOPPER ALGORITHMS INSIDE DYNAMIC BLOCKS WITHIN RHINO: UTILISING VISUALARQ'S GRASSHOPPER STYLES

GH-Styles enable the creation of single objects based on a Grasshopper definition while preserving the parametric logic and editability of the original script. The computational logic is encapsulated and exposed through a standardised user interface within the Rhino environment. Compared to native Rhino functionality, this approach allows dozens or even hundreds of GH-based objects to exist within a single model without repeatedly switching between GH files or manually opening the GH interface. Grasshopper algorithms can be adapted to become compatible with GH Styles, thereby eliminating dependence on an active GH window and addressing the lack of dynamic block functionality in Rhino.

This concept leads to the central research question. If GH scripts can be transformed into parametric objects, can the same principle be applied to scripts containing Ladybug environmental simulation components? How should inputs be referenced and processed? How should outputs be represented? What design potential would such simulation-as-object workflows offer to architects? Based on the current literature review, no documented use of VisualARQ as a GH/LB simulation wrapper has been identified. Therefore, this study conducts an experimental demonstration of a BIM-integrated simulation workflow by encapsulating a GH/LB environmental simulation script within a VisualARQ parametric object.

7. EXPERIMENT MATERIALS AND METHODS

The experiment examines the practical use of Rhino/GH/LB for environmental data-driven architectural design in combination with a workflow enabling the encapsulation of GH/LB scripts into VA parametric objects using VA GH-Styles. The objective is to increase software interoperability and enable live environmental data representation directly on BIM models, while eliminating the need to maintain complex active GH files.

7.1. Research questions and objectives

The main research questions are following:

1. Is it possible to encapsulate a functional GH/LB environmental simulation within a VA parametric object?
2. Will the resulting simulation object be:
 - independent of an active GH window?
 - editable per object instance?
 - project-independent?
 - operable by non-expert users?
 - easily duplicable, scalable and shareable?

The objectives of the research are twofold:

1. To develop an exemplary interoperability workflow dedicated to early-stage environmental simulation that overcomes identified GH/LB limitations using VA GH-Styles.
2. To demonstrate and document the experimental process, and to analyse, compare, and discuss the advantages, limitations, interoperability gains, and future potential of the proposed workflow.

7.2. Tools and simulation inputs

The experiment is conducted in Rhinoceros 3D (Rhino 8 SR27 2026-1-19), selected as a representative and widely adopted platform relevant to both individual practitioners and collaborative architectural teams. Grasshopper (build 1.0.0008) serves as the visual programming interface, hosting Ladybug (version 1.8.0) environmental simulation components. VisualARQ 3 (version 3.7.1.20537) is employed for its ability to encapsulate GH scripts into parametric objects via VA GH-Styles. A simplified urban 3D model of the Student Dormitory Jura Hronca, an accommodation facility of the Slovak Technical University in Bratislava, Slovakia, located in the Old Town district, serves as the input geometry for the Ladybug Incident Radiation simulation (Fig. 1). A Bratislava weather file in .epw format is loaded for climate data input. The simulation analysis period is set to a full year (1 January to 31 December).

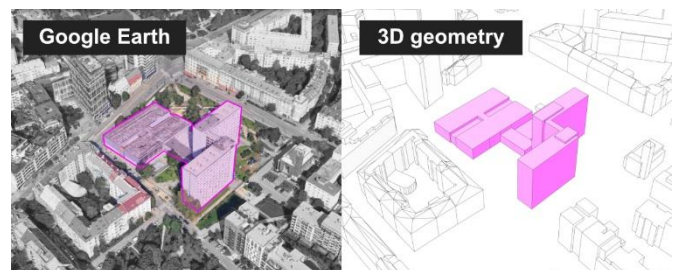


Fig. 1. Satellite image and simplified geometric representation of the input object. (Source: Google Earth, 2025; modified by author, 2026)

7.3. Methods

First, a standard Grasshopper script implementing a Ladybug Incident Radiation simulation is developed based on existing literature (Workflow A). Second, the script is restructured according to VA GH-Styles conventions. The input and output logic is iteratively reformulated until a functional encapsulated prototype is achieved (Workflow B). Finally, both workflows are compared and evaluated based on identical simulation input settings. The benchmark metric is total incident solar energy (kWh) as calcu-

Tab. 1. Workflow A: LB Incident Radiation simulation results: Total incident solar radiation (kWh) across sensor point grid sizes. (Source: Author, 2026)

Grid Size (m)	Workflow A (kWh)
10	14,188,028
8	13,731,375
5	13,767,704
3	14,197,941
2	14,416,877
1	14,670,403

7.3.2. Workflow B: Experimental GH/LB/VISUALARQ GH-styles definition

In Workflow B, the validated GH/LB Incident Radiation script from Workflow A is restructured to comply with VA GH-Styles conventions. The objective is to encapsulate the environmental simulation into a parametric VA object while preserving its computational logic (Fig. 4).

Workflow B scheme

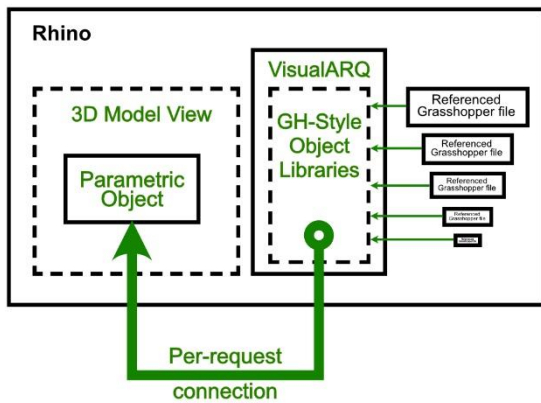


Fig. 4. Schematic representation of Workflow B, illustrating the per-request connection (i.e., recalculated only when input values change) between a VA parametric object and VA-managed GH-Styles object libraries. The number of GH-based library objects is not limited. (Source: Author, 2026)

To successfully interpret GH script by VA GH-Styles, specific input and output constraints must be respected. On the input side, only predefined parameter types are supported, such as Number Slider, Boolean Toggle, Value List, Text, and standard geometric types including Brep, Mesh, Surface, and Curve. Internal calculations within the GH definition remain unrestricted. On the output side, geometric data such as Geometry, Brep, Mesh, Curve, Extrusion, Geometry Text, and Hatch are interpreted as visible 3D objects, while numeric and textual outputs such as Integer, Number, Boolean, and Text are exposed as read-only object information.

Accordingly, several structural modifications were implemented. The weather file reference, originally defined via a Panel, was converted to a Text input to allow editing directly from Rhino User Interface (UI) (Fig. 5). Numerical parameters were converted to integer sliders and systematically renamed using indexed naming conventions to ensure consistent ordering within the VA user interface. Functional input groups were defined and labelled, as VA GH-Styles interpret/translate grouped inputs as structured sections in the object properties panel (Fig. 6).

Referencing geometry required additional adaptation. Direct Brep or Mesh inputs allow selection of only a single object instance. To enable referencing multiple geometries, both the analysed building and surrounding urban context were organised

into Rhino blocks. The GH definition then retrieves geometry from block definitions based on block names (Fig. 7). The building is extracted as a closed and valid Brep, while surrounding geometry is retrieved as meshes. This approach removes the single-object limitation and enables scalable geometric input management.

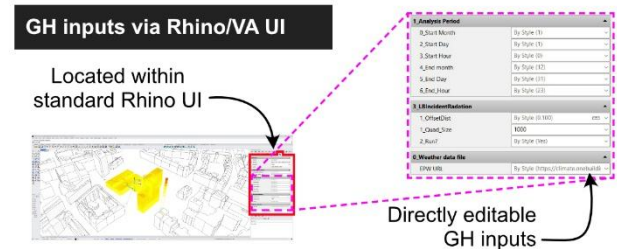


Fig. 5. Screenshot of the Rhino/VisualARQ user interface showing Grasshopper input parameters (e.g., text and numeric fields) exposed and editable through VA GH-Styles. (Source: Author, 2026)

GH UI vs. Rhino/VA UI – GH Styles translator

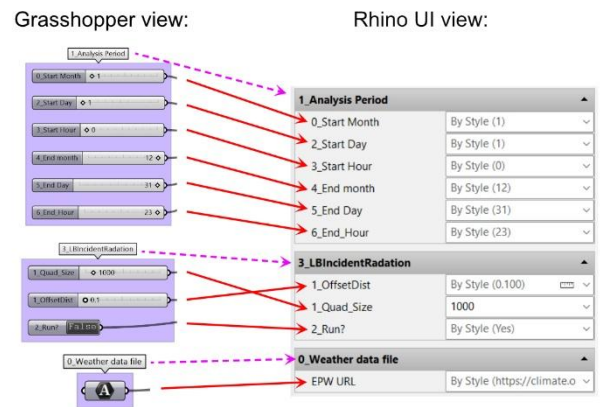


Fig. 6. Comparison of the Grasshopper user interface and the Rhino/VisualARQ interface, illustrating how GH input parameters are translated into editable fields within the Rhino environment through VA GH-Styles. (Source: Author, 2026)

Referencing Rhino blocks into VA GH Styles

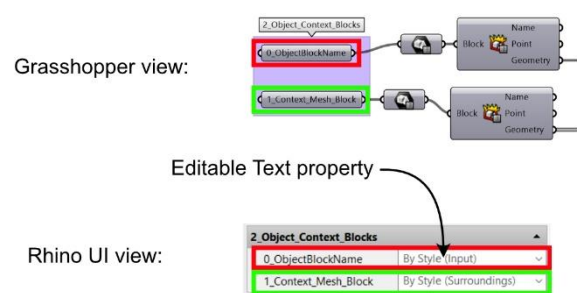
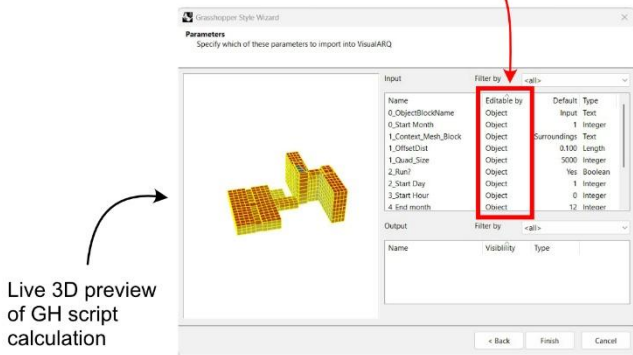


Fig. 7. Name-dependent referencing of Rhino blocks in VA GH-Styles, enabling multi-object input within a single GH-Style definition (Source: Author, 2026)

For consistency with Workflow A, simulation grid size is implemented as a numerical slider input. Grid size defines the spacing between sensor points on analysis surfaces, expressed in millimetres, and determines the resolution of the radiation calculation. This allows testing across identical grid values ranging from 1m to 10m. A Boolean Toggle is introduced to activate or deactivate the simulation solver, providing controlled execution of the calculation.

Rhino/VA GH Styles creation window

All parameters need to be editable by "Object"



Live 3D preview of GH script calculation

Fig. 8. Rhino/VA GH-Styles Creation Window. All input parameters are set to "Per object" editability, ensuring that each instance of the GH-Style simulation object within the Rhino model can operate independently with distinct weather file references and simulation settings parameters. (Source: Author, 2026)

After restructuring, the modified GH file is saved separately from the original Workflow A definition. A new VA GH-Styles object is then created within the VA Element Library as a new GH-Style object and is linked to the corresponding GH file. Finally, all input

parameters are set to "Per object" editability (Fig. 8), allowing each instance of the simulation object within the Rhino model to operate independently, including distinct weather file references and simulation settings.

8. RESULTS

The Ladybug Incident Radiation simulation runs within VA GH-Styles without errors or computational problems (Fig. 9). The encapsulated simulation behaves as a parametric VisualARQ object and produces comparable numerical outputs to the original GH-based workflow. Certain limitations were identified during testing. Text legends generated inside Grasshopper are not automatically preserved in the GH-Styles output. This can be addressed by extending the script to transfer text information explicitly to supported output types.

Also, when simulation geometry is output as Brep, colour information is lost due to current GH-Styles material limitations. This issue is avoided by using Mesh output, which preserves colour information. However, Rhino display modes might affect mesh colour interpretation and therefore represent a potential validity threat in visual result evaluation. When "Ambient occlusion" is active, the simulation mesh colours appear altered, which may lead to misinterpretation of radiation intensity. For consistency and reproducibility, "No lighting", "Default lighting", or "Scene lighting" are recommended to display consistent results (Fig. 10).

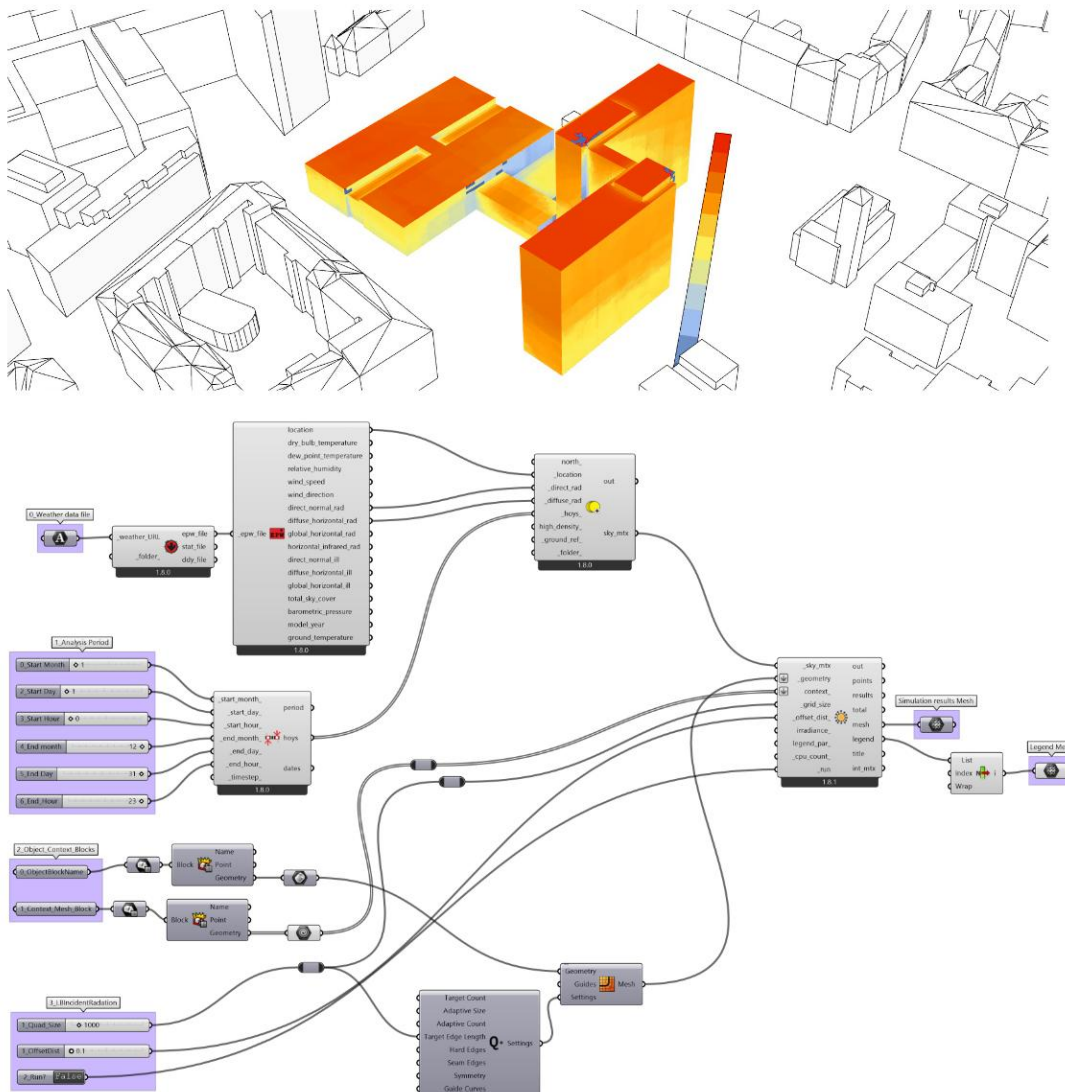


Fig. 9. Workflow B: LB Incident Radiation simulation – novel method setup and visual result interpretation (Source: Author, 2026)

Display mode lighting influence on simulation results

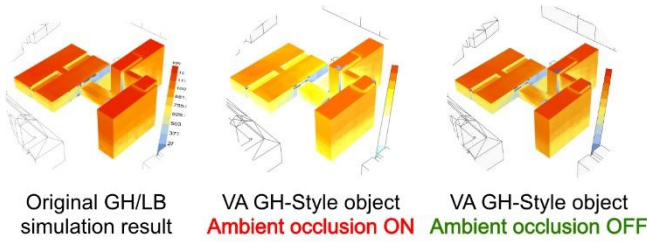


Fig. 10. Influence of display mode scene lighting on colour interpretation of GH-Style-based simulation results. (Source: Author, 2026)

A further limitation concerns referencing multiple geometries per object. Direct per-object referencing is not supported. A workaround using Rhino blocks referenced by block name allows multiple geometries to be included but introduces a potential validity threat related to spatial consistency. If block origins are modified unintentionally, referenced geometry position within LB calculations may change without any warning or possibility of visual detection, affecting simulation input conditions and compromising results.

Once encapsulated, the simulation behaves as a standard Rhino object. It can be duplicated, moved within the 3D scene and assigned independent parameter values, including grid size, .epw data file or reference geometry. Moving the simulation object does not break the relative spatial connection to referenced geometry. Simulation objects can be organised using layers and included in layout drawings. After the VA GH-Styles definition is created, Grasshopper does not need to be reopened. This lowers the barrier for team collaboration and allows non-expert users to interact with simulation parameters directly through the Rhino/VA UI.

Tab. 2. Workflow B: LB Incident Radiation simulation results: Total incident solar radiation (kWh) across sensor point grid sizes. (Source: Author, 2026)

Grid Size (m)	Workflow B (kWh)
10	14,677,267
8	14,651,567
5	14,748,749
3	14,788,684
2	14,791,074
1	14,782,840

A quantitative comparison between Workflow A (Tab. 1) and Workflow B (Tab. 2) was conducted using total incident solar radiation values across grid sizes ranging from 1m to 10m. At the finest grid size of 1m, Workflow A produced 14,670,403 kWh and Workflow B 14,782,840 kWh, corresponding to a difference of 112,437 kWh, or approximately 0.76 per cent. The values are therefore similar, although not identical.

As grid size increases, the behaviour of the two workflows differs (Fig. 11). Workflow A shows stronger variation. For example, at 5m grid size, Workflow A yields 13,767,704 kWh, which is approximately 902,699 kWh lower than its 1m reference value. At 8m, the deviation from the 1m benchmark exceeds 900,000 kWh. In contrast, Workflow B remains comparatively stable. At 5m, the deviation from its 1m value is 34,091 kWh, and at 8m it remains below 140,000 kWh. The absolute deviation between workflows decreases as resolution increases, with the largest difference observed at 5m and the smallest at 1m. The deviation is calculated as the direct subtraction of total incident solar radiation values

(Workflow A – Workflow B) for each grid size (Tab. 3, Fig. 12). The results indicate that both workflows converge at high resolution, while Workflow B shows reduced sensitivity to grid coarsening.

Comparison of simulation results: Workflow A vs. Workflow B

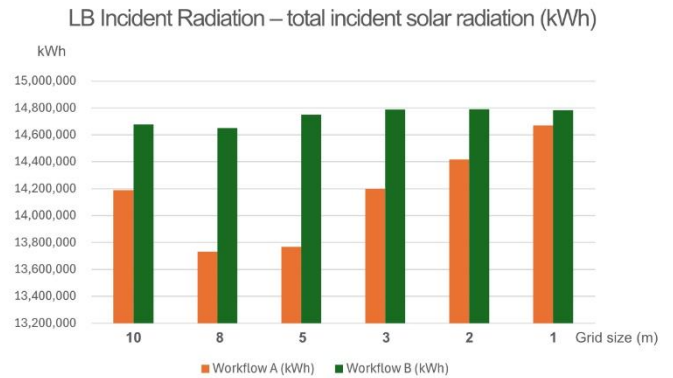


Fig. 11. Comparative results of total incident solar radiation (kWh) obtained from the LB Incident Radiation simulation for Workflow A and Workflow B. (Source: Author, 2026)

Tab. 3. Table presenting deviation values calculated as the direct difference (Workflow A – Workflow B) in total incident solar radiation (kWh). (Source: Author, 2026)

Grid Size (m)	Workflow A (kWh)	Workflow B (kWh)	Deviation A-B (kWh)
10	14,188,028	14,677,267	-489,239
8	13,731,375	14,651,567	-920,192
5	13,767,704	14,748,749	-981,045
3	14,197,941	14,788,684	-590,743
2	14,416,877	14,791,074	-374,197
1	14,670,403	14,782,840	-112,437

Comparison of simulation results: Workflow A vs. Workflow B

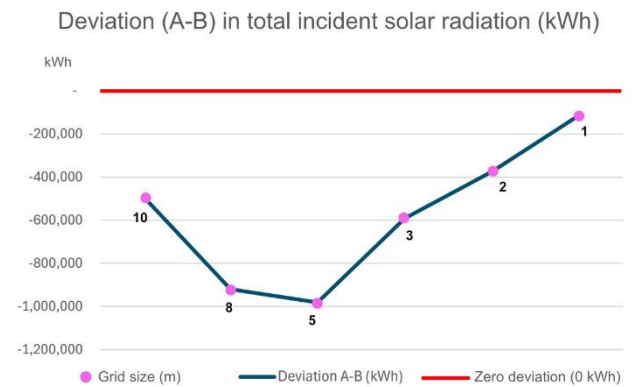


Fig. 12. Chart illustrating the values from Tab. 3, showing convergence toward zero deviation at the 1m grid resolution. (Source: Author, 2026)

9. DISCUSSION

The convergence of results at the 1 m grid size confirms that encapsulating the GH/LB script into VA GH-Styles does not alter the underlying simulation logic in a substantial way. The difference below one per cent indicates that the functionality of the original definition is preserved after transformation into a parametric object. The cause of the remaining difference remains speculative. It may be related to differences in mesh pre-processing, as block-based geometry referencing in Workflow B could influence mesh

generation or triangulation behaviour. Additionally, encapsulation via VA GH-Styles may introduce implicit geometry cleaning or more stable geometry handling.

A noticeable difference appears in the behaviour across grid sizes. Workflow A shows stronger variation as grid resolution decreases, while Workflow B remains comparatively stable. This does not prove higher physical accuracy of Workflow B, as no external validation data were used. It does, however, indicate lower numerical sensitivity to grid coarsening. For early-stage design, where larger grid sizes are often used to reduce computational time, this behaviour may be relevant from a practical point of view.

In relation to existing BIM-integrated simulation approaches, the proposed workflow does not introduce a new simulation method but reorganises how simulation logic is embedded into the modelling environment. Instead of relying on file-based script management or export-based data exchange, the simulation is treated as a parametric object within a BIM-compatible context. This shifts the focus from tool comparison toward functional interoperability.

The contribution of this study therefore lies in demonstrating that environmental simulation can be encapsulated into reusable BIM-oriented objects without loss of functionality. Measurable aspects include correct calculation of numerical results at both high and lower resolution, reduced sensitivity across grid sizes, removal of dependency on an active Grasshopper window, and simplified duplication and comparison of simulation variants. These aspects relate to usability, efficiency, and interoperability rather than algorithmic advancement.

The study is limited to the Rhino, Grasshopper, Ladybug, and VisualARQ ecosystem. The general principle, however, is not inherently software-specific. Any environment that allows parametric scripting to be encapsulated into object-based BIM structures could potentially adopt a similar approach. The practical transferability depends on the availability of comparable object frameworks and input-output control mechanisms in other platforms.

Current limitations are mainly technical. VA GH-Styles are restricted to predefined Architectural Object categories and impose specific input-output constraints. Geometry referencing through blocks introduces potential risks related to spatial consistency. These constraints reflect the present development state of the tools rather than a conceptual limitation of the simulation-as-object approach.

Future research should focus on refining the practical robustness of the proposed workflow. Improvements in VA GH-Styles input and output flexibility would reduce the need for workarounds such as block-based geometry referencing. Further investigation is required to better understand the observed differences in grid sensitivity between Workflow A and Workflow B. Controlled testing of geometry handling and data structure behaviour may clarify the source of these variations.

The simulation-as-object principle should also be tested with additional environmental analyses beyond incident radiation. Applying the workflow to other Ladybug components would help evaluate its general validity and practical relevance for early-stage design. Improving interoperability through structured IFC import and export of simulation-enabled objects represents another key step. This would allow simulation parameters and results to be exchanged more reliably across BIM platforms and extend the concept beyond a single software ecosystem. Finally, broader testing across different project types, team settings and BIM-compatible environments would help assess the usability and interoperability gains in real design practice.

10. CONCLUSION

This study demonstrates that a GH/LB environmental simulation can be encapsulated into a BIM-compatible parametric object using VA GH-Styles. The transformation from an active Grasshopper definition to an object-based implementation preserves simulation functionality and produces comparable numerical results. At 1m grid resolution, both workflows converge with less than one per cent deviation, confirming computational consistency. Across 2 to 10m grid sizes, the GH-Styles-based workflow shows reduced variation compared to the standard GH implementation. While this does not imply higher physical accuracy, it indicates lower sensitivity to grid resolution changes, which is relevant for early-stage design.

The experimental results confirm that the resulting simulation object operates independently of an active Grasshopper window once the VA GH-Styles definition is created. All input parameters can be configured as “Per object,” enabling independent editing for each instance within the model. The object is project-independent in the sense that it can be reused across files provided that the referenced GH-Style and required inputs are available. It is easily duplicable and scalable within the Rhino environment, allowing multiple simulation instances with distinct settings. While basic operation and parameter adjustment can be performed without direct interaction with Grasshopper, the initial setup and modification of the underlying GH definition still require advanced user knowledge.

The contribution of the study lies in demonstrating a functional interoperability strategy in which environmental simulation is embedded as a reusable BIM-oriented object. This approach reduces dependency on active scripting workflows, simplifies variant management, and supports broader usability within collaborative design environments.

Acknowledgements

This article was developed with the support from the Green STU university project through the STU Grant scheme for excellent teams of young researchers.

Declaration of generative artificial intelligence in writing

During the preparation of this work the author used GPT-5.2 to improve text consistency, clarity, and flow. After using this tool, the author reviewed and edited the content as needed and takes full responsibility for the content of the publication. As the final step, the paper has undergone proofreading by journal professionals.

References

- Abdulwahab, A. (2024) “CAD.GH Computational aided design via Grasshopper”, Amazon Fulfillment, Wroclaw, Poland. ISBN 979-8325824135
- Aguilar-Carrasco, M.T., Díaz-Borrego, J., Acosta, I., Campano, M.Á., Domínguez-Amarillo, S. (2023) “Validation of lighting parametric workflow tools of Ladybug and Solemma using CIE test cases”, *Journal of Building Engineering*, Vol. 64, 105608. <https://doi.org/10.1016/j.jobe.2022.105608> [Accessed: 15 Jan 2026]
- Asuni (2026) “Welcome to VisualARQ” (VisualARQ Help), VisualARQ – Flexible BIM for Rhino, Asuni. [online] Available at: <https://help.visualarq.com/> [Accessed: 25 Jan 2026]
- Bernett, A., Kral, K., Dogan, T. (2021) “Sustainability evaluation for early design (SEED) framework for energy use, embodied carbon, cost, and daylighting assessment”, *Journal of Building Performance Simulation*, 14(2), pp. 95-115. <https://doi.org/10.1080/19401493.2020.1865459> [Accessed: 25 Feb 2026]
- Chelaru, B., Onutu, C., Ungureanu, G., Volf, I., Serbanoiu, A.A. (2025) “BIM and cloud-based tools integration for BREEAM in energy-efficient building

- design: A case study", *Results in Engineering*, Vol. 26, 105493. <https://doi.org/10.1016/j.rineng.2025.105493> [Accessed: 18 Jan 2026]
- Davidson, S. (2024) "Content Cache Updated Guide", *Rhinoceros Forums*. [online] Available at: <https://discourse.mcneel.com/t/content-cache-updated-guide/181883> [Accessed: 20 Jan 2026]
- Davis, D., Burry, J., Burry, M. (2011) "Understanding Visual Scripts: Improving Collaboration through Modular Programming", *International Journal of Architectural Computing*, 9(4). <https://doi.org/10.1260/1478-0771.9.4.361> [Accessed: 28 Feb 2026]
- Dervishaj, A., Gudmundsson, K. (2024) "From LCA to circular design: A comparative study of digital tools for the built environment", *Resources, Conservation and Recycling*, Vol. 200, 107291. <https://doi.org/10.1016/j.resconrec.2023.107291> [Accessed: 28 Feb 2026]
- EUR-Lex (2024) "Directive (EU) 2024/1275 of the European Parliament and of the Council of 24 April 2024 on the energy performance of buildings (recast) (Text with EEA relevance)", Document 32024L1275, Publications Office of the European Union. [online] Available at: <https://eur-lex.europa.eu/legal-content/EN/ALL/?uri=CELEX:32024L1275> [Accessed: 20 Jan 2026]
- de Sousa Freitas, J., Cronemberger, J., Soares, R.M., Amorin, C.N.D. (2020) "Modeling and assessing BIPV envelopes using parametric Rhinoceros plugins Grasshopper and Ladybug", *Renewable Energy*, Vol. 160, pp. 1468-1479. <https://doi.org/10.1016/j.renene.2020.05.137> [Accessed: 27 Feb 2026]
- Google Earth 10.102.78.2 (2025) "Student Dormitory Jura Hronca", 48°09'19.80"N 17°07'14.76"E, elevation 456m. [online] Available at: <https://earth.google.com/web/> [Accessed: 25 Jan 2026]
- Kennedy, J. (2022) "Rhino 8 WIP Feature: Constraints", *Rhinoceros Forums*. [online] Available at: <https://discourse.mcneel.com/t/rhino-8-wip-feature-constraints/138737> [Accessed: 25 Jan 2026]
- Khan, M.S., Kim, I.S., Seo, J. (2023) "A boundary and voxel-based 3D geological data management system leveraging BIM and GIS", *International Journal of Applied Earth Observation and Geoinformation*, Vol. 118, 103277. <https://doi.org/10.1016/j.jag.2023.103277> [Accessed: 15 Jan 2026]
- McNeel (2024) "The History of Rhino", *McNeel Wiki*. [online] Available at: <https://wiki.mcneel.com/rhino/rhinohistory> [Accessed: 25 Jan 2026]
- Mode Lab (2015) "The Grasshopper Primer: Third Edition V3.2", *Mode Lab, GitBook*. [online] Available at: <https://modelab.gitbooks.io/grasshopper-primer/content/> [Accessed: 25 Jan 2026]
- Nicholson, S., Nikolopoulou, M., Watkins, R., Löve, M., Ratti, C. (2024) "Data driven design for urban street shading: Validation and application of ladybug tools as a design tool for outdoor thermal comfort", *Urban Climate*, Vol. 56, 102041. <https://doi.org/10.1016/j.uclim.2024.102041> [Accessed: 15 Jan 2026]
- Öngören, Ö., Sözer, H. (2026) "Optimizing building integrated photovoltaic facades: A parametric approach for enhanced energy efficiency and daylighting performance", *Solar Energy*, Vol. 306, 114279. <https://doi.org/10.1016/j.solener.2025.114279> [Accessed: 15 Jan 2026]
- Ou, H., Lin, T. (2023) "Effects of orientation and dimensions of shading structures on thermal comfort", *Building and Environment*, Vol. 243, 110715. <https://doi.org/10.1016/j.buildenv.2023.110715> [Accessed: 15 Jan 2026]
- Payne, A. (2022) "What is Hops", *RhinoDeveloper*. [online] Available at: <https://developer.rhino3d.com/guides/compute/what-is-hops/> [Accessed: 25 Jan 2026]
- Shibeika, A., Oliveira, S. (2024) "Environmental analysis in UK modern methods of construction (MMC) housing: insights from early-stage architectural design process", *Archnet-IJAR: International Journal of Architectural Research*. <https://doi.org/10.1108/ARCH-08-2024-0356> [Accessed: 20 Feb 2026]
- Stewart, J. (2024) "Exploring the Evolution and Impact of Rhino Software in Contemporary Architectural Design", *ResearchGate*. [online] Available at: https://www.researchgate.net/publication/382067016_Title_Exploring_the_Evolution_and_Impact_of_Rhino_Software_in_Contemporary_Architectural_Design [Accessed: 25 Jan 2026]
- Urbietta, M., Urbietta, M., Laborde, T., Villarreal, G., Rossi, G. (2023) "Generating BIM model from structural and architectural plans using Artificial Intelligence", *Journal of Building Engineering*, Vol. 78, 107672. <https://doi.org/10.1016/j.jobee.2023.107672> [Accessed: 25 Feb 2026]
- Yang, R.J., et al. (2024) "Digitalising BIPV energy simulation: A cross tool investigation", *Energy and Buildings*, Vol. 318, 114484. <https://doi.org/10.1016/j.enbuild.2024.114484> [Accessed: 27 Feb 2026]
- Yeung, J., Menacho, A.J.H., Marvuglia, A., Gutiérrez, T.N., Beach, T., Rezgui, Y. (2023) "An open building information modelling based co-simulation architecture to model building energy and environmental life cycle assessment: A case study on two buildings in the United Kingdom and Luxembourg", *Renewable and Sustainable Energy Reviews*, Vol. 183, 113419. <https://doi.org/10.1016/j.rser.2023.113419> [Accessed: 15 Jan 2026]
- Zech, P., Hammes, S., Goldin, E., Geisler-Moroder, D., Brey, R., Pfluger, R. (2025) "From BIM to Digital Twin: A transformation process through advanced control modeling and automated commissioning using daylight and artificial lighting as examples", *Energy and Buildings*, Vol. 329, 115184. <https://doi.org/10.1016/j.enbuild.2024.115184> [Accessed: 15 Jan 2026]
- Zheng, L., Lu, W., Wu, L., Zhou, Q. (2023) "A review of integration between BIM and CFD for building outdoor environment simulation", *Building and Environment*, Vol. 228, 109862. <https://doi.org/10.1016/j.buildenv.2022.109862> [Accessed: 28 Feb 2026]