

ConstraintHg: A Kernel for Systems Modeling and Simulation

John Morris  ¹¶

1 Department of Mechanical Engineering, Clemson University, Clemson, South Carolina, United States
¶ Corresponding author

DOI: [10.21105/joss.09131](https://doi.org/10.21105/joss.09131)

Software

- [Review](#) 
- [Repository](#) 
- [Archive](#) 

Editor: [Alexandr Konovalov](#) 

Reviewers:

- [@fjebaker](#)
- [@GJHSimmons](#)

Submitted: 08 August 2025

Published: 19 January 2026

License

Authors of papers retain copyright and release the work under a Creative Commons Attribution 4.0 International License ([CC BY 4.0](#)).

Summary

ConstraintHg is a systems modeling kernel used for parsing constraint hypergraphs. Constraint hypergraphs are a mathematical formalism embodying the constraint-based approach to representing behavior (Willems, 2007). Any executable model, whether database schema, plant controller, or ecological forecaster, can be represented as a constraint hypergraph (J. Morris, Mocko, & Wagner, 2025). Once combined, the unified structure shows how all elements in the system are related to each other. In addition to representing constraint hypergraphs, ConstraintHg provides methods for traversing them, equivalent to simulating the system. While most system simulations must be imperatively defined, ConstraintHg enables simulations to be constructed declaratively. As a result, knowledge about a system can be discovered autonomously, transforming a general system representation into an agentic information provider.

Statement of Need

Every thing in the world is a system: a collection of distinguishable elements that together espouse some unique behavior (Cellier, 1991). To determine that behavior, scientists describe systems using models, which express how different elements in a system interact. Models are ubiquitous: they are used to describe phenomena including weather patterns (Hoffmann et al., 2023), economic policies (Garicano & Rayo, 2016), ecological events (Wang et al., 2025) and immunological responses (Sahal et al., 2022). For the behavior of a system to be understandable, models must adhere to an established framework such as a bond graph (Borutzky, 2011) or stock and flow diagram (Baez et al., 2023). Information in such frameworks can be difficult to connect, such as comparing energy in a bond graph to population dynamics in a stock and flow diagram. These domain-specificities makes integrating models difficult, leaving scientists, engineers, and decision-makers often unable to discern the complex, multi-physics interactions.

Approaches to Generalized System Modeling

The problem with general systems modeling is often addressed under the auspices of category theory¹ (Leinster, 2014), which provides the mathematical tools for understanding intersystem relationships (Hedges, 2018). In this context, tools such as [Psymple](#) (Simmons et al., 2025), [Decapodes](#) (L. Morris et al., 2024), [ModelingToolkit.jl](#) (Ma et al., 2022), and [Modelica](#) (Mattsson & Elmqvist, 1997) have been created. All of these employ macros to discover mathematical relationships between system entities, allowing for the simulation of dynamic systems.

¹Examples of categorical approaches to general systems modeling include (Ames, 2006; Baez & Stay, 2010; Mabrok & Ryan, 2017; Robinson, 2017; Schultz & Spivak, 2017; Zardini et al., 2021).

Constraint hypergraphs (CHGs), conversely, are a functional representation of a system where the system solution is composed from elements in the model rather than applied by external solvers. This allows systems to be represented without requiring a prescriptive interface. A CHG is a mathematical structure that represent the state variables of a system as nodes in a graph and the behaviors of the system as directed hyperedges mapping between these nodes (J. Morris, Mocko, & Wagner, 2025). Each hyperedge representing a function constraining the value of the target node. The generality of this formalism is able to flexibly capture the intricacies of any system. Consequently, independent models defined across domains, scales, or even different software platforms can all be reconciled into a cohesive structure from which behavior can be interpreted holistically.

System Simulation

A model is only as useful as it is simulatable. Simulation is the use of a model to identify unobserved information. For example, a person observing someone walk into a building with a wet umbrella might infer that it is raining outside. This inference is the simulation of a mental model associating wet umbrellas with precipitation. Simulation can be understood as a function, mapping a set of known inputs (the wetness of the umbrella) to the unobserved output (the current weather). Traditional system simulation is imperative, where the modeler specifies how the explicit processes by which the transformation should be conducted. Imperative simulations are difficult to adapt; for instance, a model might define relationships mapping wetness to precipitation, cloud cover, transit delay, temperature, etc. Describing these imperatively requires a unique sequence to be explicitly defined for each pairing of inputs to outputs. For a system with n variables, the maximum limit of simulation functions that could be defined is $n(2^{n-1} - 1)$. This exponential cap makes it untenable to fully describe all imperative simulations for models with even a moderate number of states. CHGs address this by representing a system graphically such that global simulations can be created by combining local interactions. This drastically reduces the complexity of the model by avoiding redundant redefinitions.

While many packages are provided for representing hypergraphs, such as XGI (Landry et al., 2023) or HyperNetX (Praggastis et al., 2024), ConstraintHg is the first library for performing declarative simulation using a CHG. It does this by employing a breadth-first search (BFS) algorithm that can autonomously construct an optimal simulation process between arbitrarily paired input and output variables. In addition to finding simple paths, the search algorithm is able to unravel cycles in the hypergraph by performing parallel searches on cycle branches to ensure an optimal chain is found. This allows ConstraintHg to simulate dynamic systems, where the states of a system go through some type of repeated change.

In addition to dealing with dynamic systems, ConstraintHg also provides mechanisms for dealing with partial models, such as when a modeler can provide relations for only a subset of a state's values. For example, while an umbrella being wet might imply that it is raining, if the umbrella is not wet, no prediction can be made as to current weather. These partial relations become model validity frames which, while essential to systems modeling, can be difficult to capture in a modeling framework (Malak & Paredis, 2004). ConstraintHg captures a validity frame as a boolean function which is conditional to the parser including the edge in a simulation path. This feature allows modelers to specify the conditions in which a model can be relied on, an important aspect of collaborative modeling where users of a model are often unfamiliar with its limitations.

Related Research

ConstraintHg is actively being employed in work deriving the mathematical foundations of digital twins (J. Morris, Louis, et al., 2025), as well as establishing executable digital threads for model-based enterprises (J. Morris, Mocko, Wagner, & Ramnath, 2025). It has been employed

to modeling a [naval microgrid](#) (J. Morris, 2025), an [elevator lift](#) (J. Morris, Mocko, & Wagner, 2025), a kinematically-constrained [crankshaft](#) (J. Morris, Mocko, Wagner, & Ramnath, 2025), and an [additive manufacturing machine](#).

Acknowledgements

The author acknowledges valuable testing and contributions from Evan Taylor.

References

Ames, A. D. (2006). *A categorical theory of hybrid systems* (PhD Dissertation UCB/EECS-2006-165). University of California at Berkeley. <http://www.eecs.berkeley.edu/Pubs/TechRpts/2006/EECS-2006-165.html>

Baez, J., Li, X., Libkind, S., Osgood, N. D., & Patterson, E. (2023). Compositional modeling with stock and flow diagrams. *Proceedings Fifth International Conference on Applied Category Theory*, 380, 77–96. <https://doi.org/10.4204/EPTCS.380.5>

Baez, J., & Stay, M. (2010). Physics, topology, logic and computation: A Rosetta stone. In B. Coecke (Ed.), *New structures for physics* (Vol. 813, pp. 95–172). Springer Berlin Heidelberg. https://doi.org/10.1007/978-3-642-12821-9_2

Borutzky, W. (Ed.). (2011). *Bond graph modelling of engineering systems: Theory, applications and software support*. Springer. <https://doi.org/10.1007/978-1-4419-9368-7>

Cellier, F. E. (1991). *Continuous system modeling*. Springer. <https://doi.org/10.1007/978-1-4757-3922-0>

Garicano, L., & Rayo, L. (2016). Why organizations fail: Models and cases. *Journal of Economic Literature*, 54(1), 137–192. <https://doi.org/10.1257/jel.54.1.137>

Hedges, J. (2018). *Towards compositional game theory* [PhD thesis, Queen Mary University of London]. <http://qmro.qmul.ac.uk/xmlui/handle/123456789/23259>

Hoffmann, J., Bauer, P., Sandu, I., Wedi, N., Geenen, T., & Thiemert, D. (2023). Destination earth – a digital twin in support of climate services. *Climate Services*, 30, 100394. <https://doi.org/10.1016/j.cliser.2023.100394>

Landry, N. W., Lucas, M., Iacopini, I., Petri, G., Schwarze, A., Patania, A., & Torres, L. (2023). XGI: A Python package for higher-order interaction networks. *Journal of Open Source Software*, 8(85), 5162. <https://doi.org/10.21105/joss.05162>

Leinster, T. (2014). *Basic category theory*. Cambridge University Press. <https://doi.org/10.1017/CBO9781107360068>

Ma, Y., Gowda, S., Anantharaman, R., Laughman, C., Shah, V., & Rackauckas, C. (2022, February 9). *ModelingToolkit: A composable graph transformation system for equation-based modeling*. <https://doi.org/10.48550/arXiv.2103.05244>

Mabrok, M. A., & Ryan, M. J. (2017). Category theory as a formal mathematical foundation for model-based systems engineering. *Applied Mathematics & Information Sciences*, 11(1), 43–51. <https://doi.org/10.18576/amis/110106>

Malak, R. J., & Paredis, C. J. J. (2004). On characterizing and assessing the validity of behavioral models and their predictions. *Volume 3a: 16th International Conference on Design Theory and Methodology, 2004*, 325–336. <https://doi.org/10.1115/DETC2004-57452>

Mattsson, S. E., & Elmquist, H. (1997). Modelica - an international effort to design the next generation modeling language. *IFAC Proceedings Volumes*, 30(4), 151–155. <https://doi.org/10.21105/joss.09131>

[//doi.org/10.1016/S1474-6670\(17\)43628-7](https://doi.org/10.1016/S1474-6670(17)43628-7)

Morris, J. (2025). *MicrogridHg* (Version v1.0). Clemson, Naval Postgraduate School. <https://doi.org/10.5281/zenodo.15447062>

Morris, J., Louis, E., Van Bossuyt, D. L., Mocko, G., & Wagner, J. (2025, July). *Constraint hypergraphs as a unifying framework for digital twins* (No. 2507.06494). <https://doi.org/10.48550/arXiv.2507.05494>

Morris, J., Mocko, G., & Wagner, J. (2025). Unified system modeling and simulation via constraint hypergraphs. *Journal of Computing and Information Science in Engineering*, 25(6), 061005. <https://doi.org/10.1115/1.4068375>

Morris, J., Mocko, G., Wagner, J., & Ramnath, S. (2025, August 17). Declarative integration of CAD software into multi-physics simulation via constraint hypergraphs. *Volume 2A: 45th Computers and Information in Engineering Conference (CIE)*. ASME IDETC-CIE 2025. <https://doi.org/10.1115/DETC2025-168376>

Morris, L., Baas, A., Arias, J., Gatlin, M., Patterson, E., & Fairbanks, J. P. (2024). Decapodes: A diagrammatic tool for representing, composing, and computing spatialized partial differential equations. *Journal of Computational Science*, 81, 102345. <https://doi.org/10.1016/j.jocs.2024.102345>

Praggastis, B., Aksoy, S., Arendt, D., Bonicillo, M., Joslyn, C., Purvine, E., Shapiro, M., & Yun, J. Y. (2024). HyperNetX: A Python package for modeling complex network data as hypergraphs. *Journal of Open Source Software*, 9(95), 6016. <https://doi.org/10.21105/joss.06016>

Robinson, M. (2017). Sheaf and duality methods for analyzing multi-model systems. In I. Pesenson, Q. T. Le Gia, A. Mayeli, H. Mhaskar, & D.-X. Zhou (Eds.), *Recent applications of harmonic analysis to function spaces, differential equations, and data science* (pp. 653–703). Springer International Publishing. https://doi.org/10.1007/978-3-319-55556-0_8

Sahal, R., Alsamhi, S. H., & Brown, K. N. (2022). Personal digital twin: A close look into the present and a step towards the future of personalised healthcare industry. *Sensors*, 22(15), 5918. <https://doi.org/10.3390/s22155918>

Schultz, P., & Spivak, D. I. (2017, December 23). *Temporal type theory: A topos-theoretic approach to systems and behavior*. <https://doi.org/10.48550/arXiv.1710.10258>

Simmons, G., Stern, D., Osang, G., Ponti, L., Gutierrez, A. P., Facciolà, C., & Hosgood, T. (2025). Psymple: A Python package for complex systems modelling. *Journal of Open Source Software*, 10(109), 7364. <https://doi.org/10.21105/joss.07364>

Wang, X., Wu, B., Zhou, G., Wang, T., Meng, F., Zhou, L., Cao, H., & Tang, Z. (2025). How a vast digital twin of the Yangtze River could prevent flooding in China. *Nature*, 639(8054), 303–305. <https://doi.org/10.1038/d41586-025-00720-0>

Willems, J. C. (2007). The behavioral approach to open and interconnected systems. *IEEE Control Systems Magazine*, 27(6), 46–99. <https://doi.org/10.1109/MCS.2007.906923>

Zardini, G., Spivak, D. I., Censi, A., & Frazzoli, E. (2021). A compositional sheaf-theoretic framework for event-based systems (extended version). *Electronic Proceedings in Theoretical Computer Science*, 333, 139–153. <https://doi.org/10.4204/EPTCS.333.10>