OPERADS FOR DESIGNING SYSTEMS OF SYSTEMS

JOHN C. BAEZ[1] AND JOHN D. FOLEY[2]

"System of systems" engineering seeks to analyze, design and deploy collections of systems that together can flexibly address an array of complex tasks. For example, a smart phone is a system of systems: it includes a touch screen, camera, CPU, internet connection, etc. and supports a wide array of applications perhaps not foreseen, but nonetheless enabled, by its original design. To unlock this flexibility, the Defense Advanced Research Projects Agency launched the Complex Adaptive System Composition and Design Environment (CASCADE) program to develop mathematical foundations for composing and designing complex adaptive systems and demonstrate these foundations in concrete application domains.

We initially anticipated applying existing mathematics for our CASCADE work: specifically, operads and their algebras. Analogous to how a group abstracts the symmetries of an object, an operad abstracts operations that compose many objects into a single object. Each "type" in an operad defines a kind of thing it is able to compose; each "operation" defines a specific way to compose a number of things of various types into one. Just as a group can act on a set, an operad can act on something called an "operad algebra". When an operad acts on an algebra of this sort, each type is mapped to an actual set of elements of that type, while each operation is mapped to an actual function. This gives a flexible, general method for building systems of systems from components: operads give the syntax, while their algebras capture their intended meaning, or semantics, in some concrete situation. Though this general approach to systems has been applied before [3, 5, 6, 7], one of our very first use cases required new mathematics.

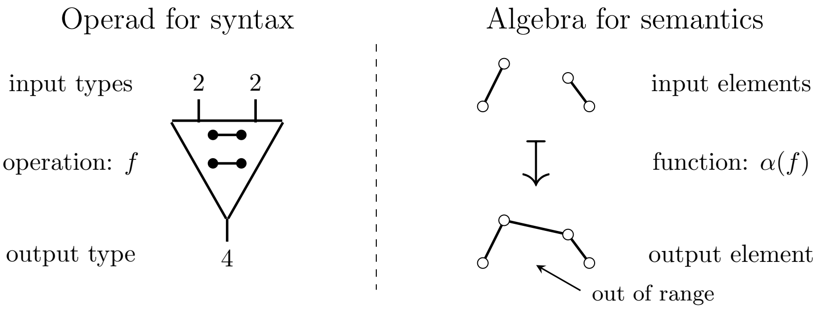

FIGURE 1. The operation f in the operad O attempts to add two edges connecting two graphs; in a certain algebra α of this operad, the concrete function $\alpha(f)$ adds edges only between vertices that are close enough.

This early application concerned different types of aircraft employing secure point-to-point communication. Since an operad was not available "off the shelf" to design networks with point-to-point links, we constructed one directly. Operations in this operad O are graphs, such as f in Fig. 1; types are the number of vertices of a network. Each operation gives a formal blueprint for combining a collection of input networks into a single output network by adding edges.

To model system composition, this operad acts on algebras whose elements describe specific aircraft at specific locations. One of the first interesting mathematical challenges was to model *range-limited* communication as an algebra of this operad. When an operation f acts on this algebra, each vertex of the graph f represents an aircraft. Each edge of f represents a potential communication link. In the algebra, these are instantiated as actual communication links only if the two aircraft involved obey a condition, e.g. being within a certain distance of each other.

Based on this example, we developed a general theory [1] which lets us efficiently construct "network operads" and algebras suitable for a wide range of applications. The operations in a

[1]DEPARTMENT OF MATHEMATICS, UNIVERSITY OF CALIFORNIA, RIVERSIDE CA, 92521, USA
[2]METRON, INC., 1818 LIBRARY ST., SUITE 600, RESTON, VA 20190, USA
E-mail addresses: baez@math.ucr.edu, foley@metsci.com.

network operad describe ways to combine several networks to form a larger one. A network operad acting on an algebra provides a starting point to automatically generate and evaluate candidate system-of-system designs. It lets us explore formally correct blueprints (operations in that operad) that combine basic systems (elements of the algebra of that operad) into a system of systems.

In the CASCADE project, we applied network operads to domains including maritime search and rescue. For example, one application problem was inspired by the 1979 Fastnet Race and the 1998 Sydney to Hobart Yacht Race, in which severe weather conditions resulted in many damaged vessels distributed over a large area. Both events were tragic, with 19 and 6 deaths, respectively. This sort of disaster remains beyond the scale of current search and rescue planning.

In the problem we dealt with, various larger assets—e.g. ships, airplanes, helicopters—could be based at ports and ferry smaller search and rescue units—e.g. small boats, quadcopters—to the search area. Network operads with directed graph operations were customized to describe allowable nestings of assets and ports. Algorithms explored designs within budget constraints to achieve a high capacity for search—known in the literature as "search effort"—that could be delivered in a timely manner. The most effective designs could ferry a large number of cheap search and rescue units quickly to the scene.

Such applications raised questions about whether limitations on the degree of vertices in a graph—e.g. limits on how many quadcopters a helicopter can carry—could be directly encoded into an operad. This is indeed possible, and the relevant theorems were proved by the first author's graduate student Joe Moeller [4].

Surprisingly, network operads—originally used to design systems of systems—can also be applied to "task" them: in other words, specify their behavior. An elegant example of this approach is given in [2] where "catalyst" agents enable behavioral options for a system. For a search and rescue application to recover downed pilots, operations were built up from primitive tasks that coordinate multiple agent types—e.g., airplanes together with helicopters—to a form a coordinated task plan. A direct translation of primitive tasks to decision variables for a constraint program—originally a mixed integer linear program and later reworked to leverage the scheduling toolkit of the CPLEX optimization software package—provided a flexible approach to automatically design task plans. Though this direct approach facilitated correct and transparent modeling for complex tasking problems, only small problems—relative to the demands of applications—were computationally tractable. So, more research is needed to develop efficient algorithms for searching in the set of operations of a network operad.

The operad formalism offers new ways to handle changing levels of abstraction in system-of-system design and tasking. Both design and tasking are aspects of a multi-stage process in which the structure and behavior of a network is specified in increasing detail, starting from a rough outline. The initial rough description of a network is typically very abstract. For example, we can describe a network of boats connected by communication channels without specifying the type of boats, their positions, or the type of channels. As we proceed further in the process of design and then tasking, we can fill in more details and move to a less abstract description. Each "level of abstraction" can be described by an algebra of a network operad. "Changing levels of abstraction" is then described by a homomorphism between operads, or a homomorphism between their algebras. We hope more researchers explore the promise of this methodology.

Acknowledgment. This work was supported by the Defense Advanced Research Projects Agency (DARPA) under Contract No. N66001-16-C-4048.

References

[1] J. C. Baez, J. Foley, J. Moeller and B. Pollard, Network models, *Theor. Appl. Categ.* **35** 20, (2020), 700–744. Available as arXiv:1711.00037.

[2] J. C. Baez, J. Foley and J. Moeller, Network models from Petri nets with catalysts, *Compositionality* **1** 4 (2019). Available as arXiv:1904.03550.

[3] S. Breiner, O. Marie-Rose, B. Pollard and E. Subrahmanian, Modeling hierarchical system with operads, *Electron. Proc. Theor. Comput. Sci.* **323** (2020), 72–83. Available as arXiv:2009.09848.

[4] J. Moeller, Noncommutative network models, *Math. Struct. Comp. Sci.* **30** (2020), 14–32. Available as arXiv:1804.07402.

[5] D. I. Spivak, The operad of wiring diagrams: formalizing a graphical language for databases, recursion, and plug-and-play circuits, 2013. Available as arXiv:1305.0297.

[6] D. I. Spivak, *Category Theory for the Sciences*, MIT Press, Cambridge Massachusetts, 2014.

[7] D. Yau, *Operads of Wiring Diagrams*, Springer, Berlin, 2018.