NIST ACT Workshop Summary

Spencer Breiner[1], Blake S. Pollard[1], and Eswaran Subrahmanian[1,2]

[1]US National Institute of Standards and Technology, Gaithersburg, MD

[2]Carnegie Mellon University, Pittsburgh, PA

The following is a summary report of a 2018 workshop Applied Category Theory: Bridging Theory & Practice, held at the US National Institute of Standards and Technology. A more substantial report will be published later this year, and a preliminary version will be circulated at the ACT2019 conference.

1 Introduction

As systems in industry, transportation, communication, health care and national security become more and more complex, their design, implementation, maintenance and diagnosis become more expensive and prone to failures. At present we lack the principled foundations needed to even express these problems, much less provide solutions, predictions or guarantees about behavior. One promising avenue for addressing these problems is the growing field of Applied Category Theory based on a mathematical language for studying compositional systems.

To explore the prospects and challenges for pursuing this approach in practice, the Information Technology Lab of the US National Institute of Standards and Technology (NIST) organized a workshop on March 15–16, 2018 with participants from across academia, industry and government (see appendix). The following is a summary of the meeting; a more detailed report will be circulated at the ACT2019 meeting and published later this year.

2 Community Development

Current Landscape: Outside of pure math, CT applications are most developed in physics and computing, most notably quantum theory and functional programming. However, there has been a recent flurry interest in broader applications, with a number of workshops on the topic held across Europe and the US. Much of this interest has been driven by a robust online community [2, 3, 24] as well as research centers at Oxford, MIT and UC Riverside.

Since the 2018 NIST workshop, community infrastructure has improved dramatically, with a new journal (Compositionality [1]), a new annual meeting (ACT2019 [9]) and an umbrella organization for smaller local meetings (SYCO [28]). So far, community development, both intellectual and infrastructural, has been driven primarily by mathematicians, though there is scattered interest from domain experts and funding agencies in a number of areas.

Challenges: The ACT community today is small and also skewed in several respects. There is a (relative) abundance of mathematicians and a deficit of domain specialists, which can lead to interesting theory but sometimes trivial or unconvincing applications. The community is also predominantly male and mostly centered in Europe and North America. At a more individualized level it is difficult to identify other researchers interested in similar applications or approaches.

Next Steps: Generally speaking, all the elements of this report–domain applications, pedagogy, tool support–are needed to grow and diversify the ACT community. Here, we consider the potential value in better centralization. First, the field would benefit from a central portal with information on papers, researchers, job opportunities, etc. This could be used both to disseminate information and to crowd-source information collection and curation from the community.

Centralization of research efforts is also valuable, especially for interdisciplinary interaction. Long-term, focused efforts will be required for CT-based approaches to meet or exceed parity with existing methods. This will also be necessary to move away from toy models, towards the more substantial proof-of-concept implementations that will be needed to sell real-world adoption.

3 Domain-Specific Applications

Current Landscape: Traditional applications in physics and computing are well-documented [4, 7], and existing communities in categorical quantum mechanics and functional programming are well-established. Outside of those areas, the applications are somewhat scattered, but cover an impressively broad range of sciences and engineering including biology [23], chemistry [5], economics [17], natural language processing [11], materials science [18], cyber-physical systems [21] and machine learning [16] to point out a few. Discovery is difficult, with publication across a wide range of math and domain venues.

Furthermore, though most publications detail incremental research, there are very few domain-specific introductions to CT which could provide entry points for this work. Compounding this problem, the quality and sophistication of CT applications varies widely. Many papers present interesting ideas marred by technical flaws, while others may be valid but too abstruse for domain specialists to follow.

Challenges: Developing a new domain application for CT involves challenges in either direction. First, domain concepts, methods *and examples* must be translated into categorical terms, often involving substantial generalization. Next, we must convince practitioners that this new viewpoint is valuable. Although better tool support should eventually make this easier, for now these arguments often rely on subjective judgements about understanding and intutition. In many cases, it is the breadth and uniformity of a CT approach which is appealing, and this may not be visible from a singular application.

Next Steps: One short term step is the creation of a literature review, organized by topic and level of sophistication, so that those from other fields can easily identify prior work which is both of interest and appropriately pitched.

When developing "CT for X", the early stages of concept translation and example building may be of little interest to those in the field; in this respect, ACT venues can act as a launch ramp as these fields come up to speed. Once initial results establish a CT perspective on the field, workshops and conference tutorials are a good way to broadcast those ideas, and generate interest and critique from the target audience.

In discussion, participants identified several specific fields which may be ripe for categorical applications, in particular systems engineering and data science. Notably, both are generic sciences, concerned with general methods which can be applied across a wide range of specific examples.

4 Pedagogy and Exposition

Current Landscape: The learning curve for CT is notoriously steep, but the barrier to entry is trending down. There are several good introductions to CT which assume no background knowledge [15, 20, 25]. Along somewhat different lines, the recent textbook [10] uses categorical structures called string diagrams to develop the basic theory quantum mechanics. There is also a wealth of online content (most notably the n-Category Café [2]) which provides accessible and informal introductions to a wide variety of topics. There is also a large class of resources available for understanding CT in terms of functional programming. CT is even the topic of a popular non-fiction book: Cheng's *How to Bake* π [8].

Challenges: The main challenge in ACT pedagogy is to balance the inherent genericity and abstraction of categorical methods with concrete examples and ideas from the field. This is especially true in less technical areas, where researchers may not be comfortable with the technical style of rigorous mathematics.

Pedagogically speaking, the breadth of potential applications is another challenge in itself. A wide variety of domain-specific introductions will be necessary to drive interest and adoption in existing fields, but this is needed in order to bring in the deep domain expertise which is sometimes lacking in ACT.

Next Steps: In addition to introductory texts, it would be useful to have a centralized and openly-licensed repository of examples, diagrams, course documents and other labor-intensive resources that could be used and improved by others in the community. This could also help to develop domain-specific pedagogy more efficiently, assembling it somewhat modularly from existing libraries of explanations and examples.

The potential for CT in the development of other subjects should not be understated, analogous to learning one's native language as a child. Authors should also consider alternative forms of presentation; for example, engineers might prefer a cookbook full of prototyped examples to formal definitions and proofs.

5 Tool support

Current Landscape:

There are a number of extant software projects which involve CT in some way; we will not attempt to survey them here, but this will be documented in the full report. Many projects are proof assistants [6, 19], developed by category theorists to help check and structure proof about categorical structures. Others use categorical structures to target some some particular domains, such as databases [22, 26] and software specification [27]. Most of these tools are are relatively small-scale prototype applications, with neither industrial horsepower nor intuitive user interactions, although some startups are working to address these issues in order to commercialize CT-based technologies.

Challenges: CT's diagrammatic notation is already among its most valuable features, but the true utility of these representations will not be revealed until we can manipulate them computationally. Today we may describe a string-diagram proof in terms of "sliding boxes past wires", but this is much less powerful than constructing a formal encoding of a proof by actually sliding boxes around a screen, an intuitive process that opens the door to a much broader audience. However, several participants warned from personal experience that the design issues involved in such a project are extremely subtle.

Additionally, the complex systems that we hope to model with CT are simply too complicated to keep track of without automated guidance. One system model might involve dozens of diagrams, tables and other "local" descriptions which are all inter-related: processes have requirements which refer to state variables which arise from components which are documented in diagrams and so on. For anything larger than a toy model, changes in one place must propagate automatically (or in a guided way) in order to maintain consistency.

Finally, the essential uniformity of categorical methods suggests that many elements of CT software should be reusable. However, existing projects tend to start from scratch and usually fail to interoperate. Better mechanisms for sharing and integrating CT-based code could lead to substantial efficiency gains in implementation.

Next Steps: One early goal should be better engagement with the functional programming community. It is a prominent success story for the field, as well as a potential partner in the development of more robust and easier-to-use tools. Similarly, there may be some affinity with the computer algebra and graph rewriting communities, which both employ CT at some levels.

Realizing the intuitive promise of a "computational category theory" will require a deep interaction between mathematicians, computer scientists, user interface experts and domain specialists. For individuals and small groups, prototype implementations remain valuable tools for exploring the space of potential software solutions. The sort of broad-based system described above is a longer-term goal, and will require substantial buy-in from both the public and the private sector. Justifying such an investment will require much better requirement and design specifications. A good start would be the specification of a categorical modeling language, to store and document CT-based models and (possibly) act as an exchange format between existing categorical data structures.

6 Funding

Current Landscape: Both interest in and funding for ACT seem to be growing rapidly, though from a low base. Much of this tied applications in quantum computing and functional programming, but also less developed areas like systems theory, information theory and natural language processing. Typical of early-stage research, most funding to date has come from government sources, both civilian and military. However, ACT's visibility in industry was elevated in recent years by several DARPA programs [12–14], and both start-up and incumbent businesses are exploring potential applications.

Challenges: The promotion of ACT within funding agencies and industry will be difficult given its reputation, even within mathematics, for abstract non-sense". Encouraging investment in this area will require a clear and concise explanation of the *pragmatic* benefits that CT brings to the table, and these are often not the elements of interest to a mathematician.

Next Steps: Obtaining the funding needed to develop ACT to its full potential will require, essentially, a marketing campaign. This should target decision-makers in terms of costs and capabilities rather than definitions and theorems; for those at the top, what CT can do matters more than how it works. There are many possible answers–understanding, precision, reuse, interoperability–which must be organized into a coherent statement of value.

On a smaller scale, a variety of partnerships may be effective for opening up new funding sources. Formal methods are highly regarded, making collaborations between mathematicians and domain specialists a relatively easy sell in many contexts. Industry-university collaborations tend to be funded at higher rates than other grant programs, and can help to build interest for CT in the commercial setting.

This sort of seed project provides the justification and proof-of-concept needed a broader marketing campaign.

7 Adoption

Current Landscape: Today, adoption of CT techniques and ideas in real-world applications is limited almost exclusively to functional programming. However, as the modern world becomes more and more connected, the importance of interaction and composite systems will only grow. The prevalence of cost over-runs and design failures in in major projects and products shows that there is deep need for a more principled and systematic approach to the modeling of complex systems, broadly construed.

Challenges: Many participants from outside mathematics expressed some confusion about what sorts of practical problems CT is useful for solving. Particularly tricky, given that CT is often viewed as a foundational approach, will be the need to work within the bounds of existing infrastructure and workflows. Significant effort must be devoted towards backward-compatibility.

It is also necessary to consider earlier attempts to apply CT, and why they failed to spark broader interest in the field. Several participants emphasized sociological characteristics that held back earlier movements, especially a fairly dogmatic attitude towards mathematical and categorical purity, and a resistance to compromise and pragmatism.

Next Steps: The first step in driving adoption of CT will be the identification of concrete use cases, and the added value that a CT-based approach provides. In many cases, this is more an issue of methodology–how does CT apply to the problem at hand?–than it is of developing new mathematics. Prototypes and detailed examples are critical, as they provide a starting point for others to refashion for their own purposes.

Proponents for a specific application need to establish a road map running from theoretical development through production. What are the milestones and metrics by which the application will be judged? How long will it take to translate from toy implementation to prototype to production system? What are the major risks involved in development, and what sorts of expertise are needed to overcome them? Answers to questions like these are necessary for ACT to move from idle curiosity to legitimate business decision.

Strategically speaking, some suggested that ACT may benefit from targeting "toolsmiths" rather than end users. These are researchers who (want to) create tools for others to use. By creating those tools within a categorical framework, it may be possible to simplify or automate the creation of features like databases, servers and user interfaces which are critical for getting a working implementation off the ground but orthogonal to the intellectual substance of the project. As several generations of software development (desktop, mobile, cloud) have shown, one of the surest paths to market share is by providing concrete benefits for developers.

References

[1] John Baez, Bob Coecke, Kathryn Hess, Steve Lack, and Valeria de Paiva, editors. *Compositionality*. TBD, 2018. Website: http://www.compositionality-journal.org/.

[2] John Baez, David Corfield, Urs Schreiber, et al. The n-Category Café. https://golem.ph.utexas.edu/category/, 2006-present.

[3] John Baez et al. The azimuth blog. 2010-present. https://johncarlosbaez.wordpress.com/.

[4] John C Baez and Aaron Lauda. A prehistory of n-categorical physics. *arXiv preprint arXiv:0908.2469*, 2009.

[5] John C Baez and Blake S Pollard. A compositional framework for reaction networks. *Reviews in Mathematical Physics*, 29(09):1750028, 2017.

[6] Krzysztof Bar, Aleks Kissinger, and Jamie Vicary. Globular: an online proof assistant for higher-dimensional rewriting. *arXiv preprint arXiv:1612.01093*, 2016.

[7] Michael Barr and Charles Wells. *Category theory for computing science*, volume 49. Prentice Hall New York, 1990.

[8] Eugenia Cheng. *How to Bake* π. Basic Books, 1 edition, 2015.

[9] Bob Coecke, editor. *Applied Category Theory 2019*. Oxford, July 2018.

[10] Bob Coecke and Aleks Kissinger. *Picturing quantum processes*. Cambridge University Press, 2017.

[11] Bob Coecke, Mehrnoosh Sadrzadeh, and Stephen Clark. Mathematical foundations for a compositional distributional model of meaning. *arXiv preprint arXiv:1003.4394*, 2010.

[12] Defense Sciences Office DARPA. DARPA-SN-17-71, Fundamental Design (FUN DESIGN). https://www.fbo.gov/utils/view?id=5bf675fd74a5142798a918587efbca64, 2017. Program Manager: Jan Vandenbrande.

[13] Defense Sciences Office Defense Advanced Research Projects Agency. DARPA-BAA-14-59, Simplifying Complexity in Scientific Discovery (SIMPLEX). https://www.fbo.gov/utils/view?id=1c8310fb3e69401d0d75d294fc5f93b3, 2014. Program Manager: John Paschkewitz.

[14] Defense Sciences Office Defense Advanced Research Projects Agency. DARPA-BAA-16-11, Complex Adaptive System Composition And Design Environment (CASCADE). https://www.fbo.gov/utils/view?id=1c8310fb3e69401d0d75d294fc5f93b3, 2015. Program Manager: John Paschkewitz.

[15] Brendan Fong and David I Spivak. Seven Sketches in Compo-

sitionality: An Invitation to Applied Category Theory. *arXiv preprint arXiv:1803.05316*, 2018.

[16] Brendan Fong, David I Spivak, and Rémy Tuyéras. Backprop as functor: A compositional perspective on supervised learning. *arXiv preprint arXiv:1711.10455*, 2017.

[17] Neil Ghani, Jules Hedges, Viktor Winschel, and Philipp Zahn. Compositional game theory. In *Proceedings of the 33rd Annual ACM/IEEE Symposium on Logic in Computer Science*, pages 472–481. ACM, 2018.

[18] Tristan Giesa, Ravi Jagadeesan, David I Spivak, and Markus J Buehler. Matriarch: a python library for materials architecture. *ACS biomaterials science & engineering*, 1(10):1009–1015, 2015.

[19] Aleks Kissinger and Vladimir Zamdzhiev. Quantomatic: A proof assistant for diagrammatic reasoning. In *International Conference on Automated Deduction*, pages 326–336. Springer, 2015.

[20] F William Lawvere and Stephen H Schanuel. *Conceptual mathematics: a first introduction to categories*. Cambridge University Press, 2009.

[21] Eugene Lerman. A category of hybrid systems. *arXiv preprint arXiv:1612.01950*, 2016.

[22] R Rosebrugh, R Fletcher, V Ranieri, K Green, J Rhinelander, and A Wood. EASIK: An EA-Sketch implementation kit, 2009.

[23] Robert Rosen. *Life itself: a comprehensive inquiry into the nature, origin, and fabrication of life*. Columbia University Press, 1991.

[24] Pawel Sobocinski. Graphical linear algebra, 2015-present. https://graphicallinearalgebra.net/.

[25] David I Spivak. *Category theory for the sciences*. MIT Press, 1 edition, 2014.

[26] David I Spivak and Ryan Wisnesky. Relational foundations for functorial data migration. In *Proceedings of the 15th Symposium on Database Programming Languages*, pages 21–28. ACM, 2015.

[27] Yellamraju V Srinivas and Richard Jüllig. Specware: Formal support for composing software. In *International Conference on Mathematics of Program Construction*, pages 399–422. Springer, 1995.

[28] et al. Vicary, Jamie, editor. *Symposium on Compositional Structures*. University of Birmingham, September 2018. Website: http://www.cs.bham.ac.uk/~vicaryjo/syco/1/.

A NIST ACT Workshop Participants

John Baez (UC Riverside)
Radhakrishnan Balu (Army Research Lab)
Philippe Belmans (Dassault Systemes)
Spencer Breiner (NIST)
David Broniatowski (GWU)
Kristopher Brown (Stanford)
Arquimedes Canedo (Siemens)
Jules Chenou (Mercury Data Systems)
Bob Coecke (Oxford)
Jared Culbertson (Air Force Research Lab)
Peter Denno (NIST)
Zinovy Diskin (McMaster)
Stephane Dugowson (SupMéca)
John Foley (Metron)
Brendan Fong (MIT)
Mark Fuge (UMD College Park)
Fabrizio Romano Genovese (Statebox)
Alwyn Goodloe (NASA Langley)
Helle Hansen (TU Delft)
Perry Harabin (Strategic Analysis)
Jelle Herold (Statebox)
Marisa Hughes (Johns Hopkins/APL)
Steve Huntsman (BAE Systems)
Al Jones (NIST)
Cliff Joslyn (Pacific NW National Lab)
Vicky Lynn Karen (NIST)
Xiang Li (NIST)
Jason Lucas (Metron)
Oliver Lum (Strategic Analysis)
Charlie Manion (UMD College Park)
Carl Miller (NIST/UMD)
Tom Mifflin (Metron)
Ira Monarch (CMU)
Sarala Padi (NIST)
Evan Patterson (Stanford)
Dusko Pavlovic (U Hawaii)
Blake Pollard (NIST/CMU)
Emilie Purvine (Pacific NW National Lab)
William Regli (UMD College Park)
Michael Robinson (American U.)
Alison Schuetz (Booz Allen Hamilton)
Alberto Speranzon (Honeywell)
David Spivak (MIT)
Ram Sriram (NIST)
Eswaran Subrahmanian (NIST/CMU)
Joshua Tan (Oxford)
Remy Tuyeras (MIT)
Andrew Vlasic (Bennett Aerospace/Army Research Office)
Viktor Winschel (Oicos)
Ryan Wisnesky (Categorical Informatics)
Paul Witherell (NIST)