NIST Special Publication 1249

Workshop on Applied Category Theory:
Bridging Theory and Practice

Spencer Breiner
Blake Pollard
Eswaran Subrahmanian

This publication is available free of charge from:
https://doi.org/10.6028/NIST.SP.1249

National Institute of
Standards and Technology
U.S. Department of Commerce

NIST Special Publication 1249

Workshop on Applied Category Theory:
Bridging Theory and Practice

Spencer Breiner
Blake Pollard
Software and Systems Division
Information Technology Laboratory

Eswaran Subrahmanian
Engineering and Public Policy
Carnegie Mellon University
Pittsburgh, PA, USA

This publication is available free of charge from:
https://doi.org/10.6028/NIST.SP.1249

February 2020

U.S. Department of Commerce
Wilbur L. Ross, Jr., Secretary

National Institute of Standards and Technology
Walter Copan, NIST Director and Undersecretary of Commerce for Standards and Technology

National Institute of Standards and Technology Special Publication 1249
Natl. Inst. Stand. Technol. Spec. Publ. 1249, 38 pages (February 2020)
CODEN: NSPUE2

This publication is available free of charge from:
https://doi.org/10.6028/NIST.SP.1249

Abstract

This report presents the summary of a workshop held at NIST on March 15-16, 2018 on the topic of applied category theory (ACT). The meeting had two main goals: (i) mapping the current ACT landscape and (ii) developing a roadmap for transitioning the field to concrete applications. The report is broken into six sections detailing different aspects relevant to the development of ACT: community development, domain-specific applications, pedagogy, tool support, marketing and funding. Each section contains a discussion of the current state of the field, identifies major goals and challenges in that area, and considers potential strategies and tactics to address them, along with a prospective timeline for future developments.

Key words

Applied Category Theory; Interoperability; System Science.

Table of Contents

List of Tables

Workshop Participants

John Baez	(UC Riverside)
Rad 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	(Army Research Office/ Bennett Aerospace)
Viktor Winschel	(Oicos)
Ryan Wisnesky	(Conexus AI)
Paul Witherell	(NIST)

1. Introduction

The modern era is characterized by an increasing dependence on complex, interlinked systems. This is true in nearly every sphere, from transportation and communication to medicine and manufacturing. As these systems' capabilities expand, our understanding of them shrinks: design is more complex, problems are harder to diagnose, implementation is more expensive. The speed and interaction enabled by digital technology means that human intuition is no longer sufficient to manage such complexity, but we currently lack the principled foundations needed to formalize these problems, much less their solutions.

One promising avenue for addressing this concern is the emerging field of *applied category theory* (ACT), based on a mathematical language for defining and studying compositional systems, where complex entities are built up from smaller, simpler pieces. The earliest applications of category theory (CT) were in pure math, where it provided a common language for comparing algebraic and geometric structures [1]. Later, in the 1970's and 80's, it became clear that the same mathematical structures devised to link algebra and geometry could be used to analyze physical, logical and computational systems as well. Today, CT is well-established across a broad range of pure mathematics, theoretical physics and computer science.

ACT is an attempt to use this same collection of mathematical tools to organize a much wider and (currently) less mathematicized range of activities by reducing them all to a relatively small dictionary of common categorical abstractions. ACT aspires to provide a global perspective in which more-or-less any problem can be formalized in terms of categorical structures. This claim is as yet unproven, but it is supported by substantial *prima facie* evidence:

- A wide (if piecemeal) range of existing applications in biology, chemistry, economics, data science, linguistics and more, in addition to the well-developed applications within math, physics and computer science.
- Deep relationships with formal logic, computation and information theory, other ostensibly global disciplines.
- A self-referential approach, in which categorical structures themselves form compositional systems subject to categorical analysis.

Realizing the goals outlined above will require significant investments of planning, time, effort and funding. In particular, most existing work remains theoretical, and putting those insights into practice will require substantial effort and stronger ties between mathematicians on one hand and domain and industry experts on the other.

1.1 Workshop Theme and Report

On March 15-16, 2018, the Information Technology Laboratory at the US National Institute of Standards and Technology (NIST) held a workshop to begin planning for this effort, entitled "Applied Category Theory: Bridging Theory and Practice" (**ACT-NIST**). The meeting had two main goals: (i) mapping the current ACT landscape and (ii) developing a roadmap for transitioning the field to concrete applications. The workshop itself was a mix of invited talks, parallel discussion sessions and free-form group discussions. Slides and videos for most of the talks as well as additional workshop materials (e.g., announcement, handouts) are available in the supplemental material.

Workshop participants identified several avenues of value when a problem or domain is translated into CT. One is in terms of conceptual clarity and intellectual 'hygiene'. CT forces a very organized and explicit identification of the components of a problem and how they fit together. Once framed in these terms, CT provides a toolbox of generic definitions and theorems which can be specialized to the problem at hand. Moreover, by expressing different problems in the same language, CT eases interdisciplinary work and generalization across domains. Furthermore, CT supports a unique style of formal diagrammatics in which mathematical structures can be defined and manipulated using intuitive, two-dimensional diagrams. This is a powerful tool both for organizing categorical models and explaining them to those outside the field.

This report summarizes the discussions from the 2018 workshop. Rather than attempt to reconstruct the discussions themselves, the content has been organized into six sections detailing different aspects of ACT development: community development, domain-specific applications, pedagogy, tool support, marketing and funding. Each section begins with current state of the field in that area (Current Landscape), followed by considerations moving forward (Roadmap) and closes with a short, prospective timeline for future developments.

2. Community Development

The development of ACT as an activity in its own right, separate from and generalizing over the particular areas of application, is a fairly recent development. As such, one of the initial challenges facing the field is the creation of a coherent community of researchers, as well as support infrastructure for both academic and industrial participants. This section describes the current state of community development process, and outline future steps for continued progress.

2.1 Current Landscape

Although communities interested in CT applications for physics and computer science are well-established, ACT has only emerged as a distinct field within the last decade. Over the past few years, interested researchers have convened several workshops and meetings (including the present meeting) both to share results and develop a more cohesive community. These include

- Categorical Methods at the Crossroads - Dagstuhl, DE - April 22 - May 2, 2014 [2]
- Computational Category Theory - NIST, MD, USA - Sept. 28-29, 2015 [3]

- Compositionality - Simons Institute, CA, USA - Dec. 5-9, 2016 [4]
- Special Session on ACT - AMS Western Sectional, CA, USA - Nov. 4-5, 2017 [5]
- ACT: Bridging Theory and Practice - NIST, MD, USA - March 15-16, 2018 [6]
- ACT: Towards an Integrative Science - Lorentz Center, NL - April 30 - May 4, 2018 [7]
- First Symposium on Compositional Structures - University of Birmingham, UK - Sept. 20-21, 2018 [8].

One area where ACT is still lacking is in academic infrastructure, such as journals and conferences, but this gap is currently being closed. At **ACT-NIST** , Joshua Tan announced a new journal *Compositionality* [9] devoted to applied categorical methods to launch in mid-to-late 2018. Additionally, Bob Coecke announced a new series of monographs devoted to domain-specific tutorials in categorical methods. The following month, in Leiden, Jamie Vicary announced a new series of symposia, the Symposium on Compositional Structures, with an inaugural meeting in September 2018.

A more substantial deficit, at present, is a lack of domain specialists and engineers from outside mathematics. Most of those involved with ACT are academics, but meeting the larger goals of the field will require more substantial buy-in from government and industry clients. With representatives from seven government organizations and six major corporations, **ACT-NIST** demonstrated clear interest from these groups. Nevertheless, much more work is needed before the applications and implementation side of ACT can match their theoretical counterparts. We will continue this line of discussion in Section 6.

The field of ACT is growing slowly but steadily and in many directions. As one participant remarked, there is a good case for optimism even if ”we just keep doing what we're doing.”

2.2 Roadmap

Looking ahead, discussion around community focused on two main concerns: coherence and constitution.

One clear takeaway from the discussion is the need for a central repository of information on ACT. This should include, at a minimum;

- A directory of ACT researchers, including interests, geographic locale and contact information.
- A hierarchical map of domain applications, with literature surveys, summaries of open problems, and links to experts in those areas.
- A directory of CT-based tools, with information on use cases and contributors.

- A blog or other content stream for communicating both results and ACT-relevant announcements (e.g., call for papers, funding opportunities, etc.)

One option raised for such a nexus is the Applied Category Theory website created shortly after the Sept. 2014 NIST meeting mentioned above. The site currently hosts a neglected Wordpress blog, and would require a redesign to meet the criteria above. Another alternative might be the creation of a blog linked to the upcoming *Compositionality* journal.

Two related issues addressing such a portal are the need for a content stream to encourage regular visits to the site, and the need to maintain these directories as information goes out of date. One suggestion to address these issues is the use of (volunteer?) editors to solicit content and maintain updated information.

Further strategies for building community coherence concern additional support infrastructure for those in the field, such as

- Additional journals and/or monograph series.
- An annual conference.
- More (regional) workshops.
- One or more research institutes for ACT.

Academics, in particular, emphasized that these infrastructure components are important for providing the credibility, acknowledgement and funding that early-career researchers need in order to invest time in a new area like ACT.

A second set of community development strategies concerned the size and composition of the ACT community. First was a recognition that the community needs to grow; a lack of human resources is one of the binding constraints on the field today. In particular, a survey of participants prior to the meeting indicated a need for more interlocutors, those able to provide domain-specific interpretations for CT's generic abstractions.

More generally, and as noted above, the ACT community today consists primarily of academic mathematicians. Further development is contingent on firmer grounding in the domain knowledge from the areas ACT would like to target. Thus, one central challenge is to bring researchers from outside domains into the ACT community in order to build stronger ties between mathematicians and domain scientists and engineers. Due to its core importance, we will touch on this issue in a number of places, including Sections 3, 4, 6, and 7.

An additional desire expressed by several participants is greater contact with the functional programming (FP) community, with several avenues of potential value for ACT. First, the FP community has extensive experience with both the benefits and the hazards of translating CT from theory to concrete implementation. Second, the successes of FP can act as a guide for developing value and driving adoption of formal methods in practical contexts. Third, one of the major constraints on ACT today is a lack of tool support, and some in the FP community might be willing to help build such tools.

Finally, several participants noted the rather steep social imbalance at the **ACT-NIST** workshop, with only a handful of women and minorities in attendance. Many felt that this lack of diversity is best addressed now, while the community is small.

2.3 Timeline

1-2 yr.
Est. annual conf/flagship journal Diversify community Connect with functional programming
3-5 yr.
Centralized Repository/Website Special issues Regional meetings
6-10 yr.
Establish ACT research institute(s)

3. Domain-Specific Applications & Use Cases

The range of potential applications for CT encompasses the hard and soft sciences, information processing and computation, as well as more hands-on areas such as design and engineering. However, outside of the most mathematically anchored sciences (physics, computer science) these applications are mostly piecemeal and spotty. Consequently, there remains much work to be done in order to justify the value of this approach to domain practitioners. In particular, to justify commercial investment in CT methods, the ACT community must develop concrete use cases which clearly articulate CT's value in comparison to existing best practices.

3.1 Current Landscape

The discussion of current applications focused on the challenge of developing "category theory for X". To do so requires a translation in both directions: first reframing important concepts from X (terminology, laws, proofs) into categorical terms, and then reframing the essential elements of CT in terms of X in order to communicate those insights back to the domain.

Participants identified two common failure modes for ACT:

- A mathematician becomes interested in X, identifies CT structures in a standard textbook presentation, speaks or publishes the results in a domain venue, and garners little interest from domain specialists.

- A domain specialist becomes interested in CT, identifies CT structures in a complex domain phenomenon, presents the results in sketchy or erroneous detail, and garners little interest from the mathematical community.

To some degree, mathematicians lack insight into the practical value of their discoveries and how to communicate them to practitioners; domain specialists lack the intuition needed to correctly identify CT structures and the rigor to justify those observations.

For this reason, the most successful research in ACT is likely to be collaborative. Working collaboratively avoids both pitfalls as mathematicians keep things accurate and precise while domain specialists keep them interesting and comprehensible. In this respect, the ACT community can provide a valuable service by matching interested parties on both sides. Today this is mostly limited to word of mouth and conference encounters, but more organized materials for the field could reduce friction substantially.

One significant obstacle in building up "CT for X" is the difficulty of demonstrating the value of CT using small, singular examples. Communicating ACT research often requires introducing large chunks of elementary CT to preface an application. The generality of this background material is invisible in light of a single usage, and makes even easy applications seem overly complicated and top-heavy. As one participant put it, "CT is not a free lunch, it is a full meal!" Unfortunately, modern academia, where most proofs of concept are generated, is ill-suited to a more holistic approach.

Additionally, ACT faces an "innovators' dilemma" in each new field, with early-stage work recovering existing concepts and methods rather than pushing the field forward. Such work may appear trivial from the domain perspective, even if it is natural and necessary for the mathematician. Ultimately, this foundational work should be recognized as an investment in ACT more than the domain of interest. From this perspective, the ACT community provides a venue and launchpad for mathematically interesting applications that are not yet competitive in their target domains.

3.2 Roadmap

As indicated above, the simplest strategy to advance "CT for X", at least in the early stages, is to pair a mathematician with a specialist in X. The specialist explains various elements of the domain, while the mathematician sketches out a formal scaffolding to organize those concepts. Over time, the two build up a categorical map for (some fragment of) the domain.

This strategy has low upfront costs, but requires substantial time investment. Some participants estimated one year of regular effort to establish a common understanding in both directions. However, progress accelerates over time as the shared conceptual language grows.

In general, it would be helpful to establish a clearer understanding of what sorts of problems CT is (and is not) good for solving. This could help collaborators better target their early investigations. To this end, participants suggested several broad classes of problems that CT may be relevant for solving:

- Generalization - Transferring existing methods from one context to another.

- Translation - Establishing transformations between distinct but related models.
- Modularization - Organizing complex representations into common design patterns.
- Specification/Documentation - Providing precise, formal representations for domain structures.

Broadly speaking, participants felt that CT is good for creating and linking complex representations; its relevance to a particular problem can be estimated by the degree to which existing representations and transformations are unsatisfactory. Sometimes this may lead to faster or more capable algorithms, but often the primary benefit is cognitive, reducing overhead through abstraction. Several participants with direct experience applying these ideas in commercial contexts emphasized that, as these methods are fleshed out, it is important to avoid grandiose claims. Instead, look for small but definitive advances which indicate future directions.

Once a body of initial results has been established, the next step is to establish an active "CT for X" sub-community. Individual talks at domain conferences may stimulate some interest, but until CT approaches reach parity in those fields, most felt that these are unlikely to generate sustained interest. This approach also suffers from the poor overhead/application balance discussed above.

Others suggested more targeted approaches including:

1. "CT for X" workshops - Workshops provide relatively low-cost opportunities for networking as well as sympathetic feedback to improve messaging and presentation for target audiences.
2. Conference exchange - Establish parallel sessions at ACT and domain conferences; grouping presentations improves overall overhead/application balance (in either direction).
3. Conference tutorials - Many domain conferences offer longer tutorials (e.g., 3-5 hours) on special topics of interest; this provides the opportunity (and challenge) to present a more coherent statement of how CT representations can benefit practitioners of X.

As several participants emphasized repeatedly, in all these cases it is critical to speak to domain specialists *in their own language*. For example, engineers are more likely to be convinced by empirical case studies and simulations than by formal proof. Presentations to domain specialists should typically start from a concrete, domain-relevant example and use this to reverse-engineer any relevant theoretical constructs. This is directly opposed to the mathematician's usual presentation of definition-theorem-proof.

There was lack of consensus on how much of the mathematics should be exposed to practitioners. On one hand, maintaining focus on key ideas (e.g., composition) may require suppressing other mathematically important details (e.g., identities). At the same time, it can be dangerous to hide too much of the mathematical details, lest practitioners see a string diagram or a commutative diagram and write it off as "just another graph".

Participants also suggested a number of community-level steps that could be taken to improve the visibility and coherence of ACT across all domains. One immediate proposal is to organize a bibliography cataloging the ACT literature, organized by application area, so that incoming domain specialists can easily find work that is directly relevant to their interests.

For ease of creation and maintenance, this bibliography could be structured as a living document and repository. Much of the effort (e.g., source identification) could be crowd-sourced, and may need to be, given the wide range of publication venues for relevant work. Other useful information that could also be assembled includes:

- Summaries of domain-specific literature (e.g., "Overview of CT in Biology").
- Informal peer review and quality assessment.
- Rankings or flags for domain sophisitication, mathematical accessibility and/or serious errors.
- Challenge problems and conjectures.

Though there are many potential applications for CT, some felt that it would be strategic to (collectively) target a few application areas early on. This would help to realize concrete advances, thereby gaining legitimacy and additional resources. The discussion revealed two different strategies for selecting application domains.

Some felt that the best return on investment can be found in formalizing the foundations of fundamental and established sciences like biology and chemistry, a strategy that has already been effective in physics and computer science. The standard curricula in these fields provide a ready-made guide to the main ideas that need to be elaborated. Because many other fields (e.g., medicine, material science) depend on these fundamental sciences, the results of this work are likely to find broad application.

Others felt that the areas ripest for CT formalization are those that currently lack established and agreed-upon foundations altogether. On one hand, this indicates that the organizational element of a categorical formulation would be especially valuable. At the same time, the lack of a dominant paradigm in these fields may encourage greater openness to new approaches. Two areas–data science and systems engineering–were suggested as particularly attractive candidates for the latter approach. Like CT, both aim at universality. Moreover, both are intimately concerned with composition (of data transformations, of system components).

3.3 Timeline

1-2 yr.
Seed funding for CT/domain collaborations "CT for X" workshops, special sessions Develop ACT Bibliography & Literature Review
3-5 yr.
Conference exchanges Conference tutorials Establish challenge problems and prototype solutions
6-10 yr.
Establish domain-specific ACT methodologies, tools and languages

4. Pedagogy & Exposition

As noted in the discussion of community development (Section 2), human capital is one of the binding constraints on progress in ACT. There are simply too few knowledgeable researchers to support the ambitions of the field, especially outside of academic mathematics and computer science. This leads to a deficit in the domain expertise and intuition needed to build convincing use cases and examples. Already, good pedagogy and exposition have been important drivers in the development of ACT. Further steps are needed to realize the field's potential.

4.1 Current Landscape

For many of the participants at the **ACT-NIST** workshop, improved exposition has been and will continue to be the main driver of ACT adoption. This is, in part, a historical perspective. Most mathematical methods are developed to handle relatively simple examples and later generalized to more complicated situations. The history of CT has proceeded largely in reverse: the earliest applications were exceedingly abstract and, over time, mathematicians have found that the same perspective remains useful when applied to progressively simpler and more intuitive problems.

This history highlights abstraction as a central challenge in communicating categorical ideas to a wider audience. Unification through abstraction is undoubtedly one of CT's principle virtues; by a careful analysis of common structural features, the same generic construction can be specialized to a wide range of specific examples. Pedagogically, though, this can be an obstacle as it forces newcomers to juggle application and abstraction at the same time, contributing to a notoriously steep learning curve for CT.

However, a strong expositional landscape has lowered these barriers to entry in recent years. Progress can be tracked on two main fronts: traditional academic publishing and informal, mostly web-based media.

Among traditional publications, participants pointed to several strong introductions to CT from scratch, assuming neither deep background knowledge nor mathematical maturity. Lawvere & Schanuel's *Conceptual Mathematics* [10] carefully builds from elementary properties of sets and functions up to the core elements of topos theory. Spivak's *Category Theory for the Sciences* [11] is pitched at roughly the same level, and uses examples from outside mathematics to illustrate many points. These supplement several other recent textbooks that focus on more traditional applications in mathematics [12–14]. In addition to these, many academics provide unpublished or early versions of CT texts free online; see [15] for an extensive list.

More recently, Fong & Spivak have released *An Invitation to Applied Category Theory* [16], an introduction focused specifically on ACT. It is based on a course that the authors taught at MIT. The text uses categorical representations to analyze a variety of concrete examples ranging from resource dependence and cooperative design to electrical circuits and behavioral verification. Along similar lines, Bradley has written a short, illustrated monograph *What is Applied Category Theory?* [17] which gives a leisurely introduction to some of the big ideas in the field.

In addition to resources directed towards learning CT, some noted the recent use of categorical methods as a foundational tool in the pedagogy of *other* subjects. Whereas the texts above expose CT for its own sake (though they may use examples from other disciplines), the aim of Coecke & Kissinger's *Picturing Quantum Processes* [18] is to develop the basic elements of quantum theory. It simply uses categorical representations (specifically string diagrams) towards that end.

In his talk at **ACT-NIST** , Coecke discussed the pedagogical value of this approach. He also announced an upcoming book series which will focus on this sort of domain-specific introduction to CT. Along similar lines, Pavlovic's presentation concerned his development of a similar course in computer science at the University of Hawaii, and its utility for simplifying and unifying certain concepts in cryptography. Slides and videos of the talks are provided with the supplementary material 7.3.[1]

Outside of academic publishing, participants remarked on the growth of online, mostly informal resources for learning about CT. Of these, the most important channels identified by participants were blogs, wikis and online videos.

Blogging on CT has a long history, beginning with Baez's *This Week's Finds in Mathematical Physics*, a weekly blog published from 1993 to 2010 [19]. Baez, a physicist by training, regularly discussed categorical ideas and methods using examples from physics. In 2006, Baez, Corfield and Schreiber founded the *n-Category Café*, a group blog specifically focused on categories (and their higher-dimensional analogues) and the roles that that they play in "math, physics and philosophy" [20]. The Café's posts, along with extensive community commentary, provide informal and relatively accessible discussions of many CT ideas and applications. Other active blogs with a significant focus on CT include *Azimuth* [21], *Annoying Precision* [22], *Graphical Linear Algebra* [23], *Theoretical Atlas* [24] and *Math3ma* [25].

[1]Due to an unfortunate technical error, only the first half of Pavlovic's talk was recorded.

Most of those listed above provide a mathematical perspective on CT, but participants also identified a number of blogs which approach the same material from the perspective of programming and computer science. Milewski's *Programming Cafe* [26] uses Haskell to develop many categorical ideas from the perspective of programmers–"engineers rather than scientists"–and even collects these together into an introductory text [27]. Notably, Milewski has supplemented his presentation with a series of online videos [28]. More generally, one can easily find introductions to CT tailored to a variety of programming languages including Typescript [29], Scala [30–32], F# [33], C# [34] and many others.

Alongside blogs, wikis play several important roles in disseminating categorical ideas. The *nLab* [35], a spin-off of the *n-Category Café*, is an online encyclopedia in the style of Wikipedia but focused exclusively on CT methods. The site also provides a private sandbox for individual contributors. Though it is an invaluable resource for experts, the community's jargon can be an obstacle for outsiders (see, e.g., the *mLab* parody [36]). Another important wiki for the CT community is MathOverflow [37], a question & answer site devoted to research-level mathematics, with many CT-specific questions and responses.

Finally, and unsurprisingly in today's media environment, online video provides another avenue for those interested in learning about CT. In addition to Milewski's videos mentioned above, the Catsters [38] have recorded video introductions for many ideas in basic category theory. Additionally, video recording of academic lectures and talks is much more common than in the past, with many released free online.

It is not web-based, but Cheng's popular nonfiction book *How to Bake* π [39] provides another informal exposition of CT. Published as *Cakes, Custard and Category Theory* in the UK, the book uses analogies with cooking as a jumping-off point to introduce some big ideas from CT and modern mathematics. Specifically, Cheng argues that, in contrast to its popular conception as a difficult field, mathematics is actually a study of "what is easy", addressing problems that can be standardized and treated in a uniform fashion. She goes on to argue that CT is "the mathematics of mathematics", making superficially difficult mathematical ideas easier through expression in a common, abstract language. This was recommended for those interested in understanding *why* CT is valuable, especially those without a technical background.

4.2 Roadmap

A major challenge in ACT pedagogy is to balance the inherent genericity and abstraction of categorical methods with concrete examples and domain intuition. This is especially true as the field tries to target those from outside the traditional domains of math, physics and computer science. As workshop participants emphasized repeatedly, ACT can only succeed by speaking to domain specialists in their own languages.

One proposed target, then, is a collection of domain-specific introductions to CT in a wide variety of areas. In fact, Coecke reported to **ACT-NIST** that he is developing a monograph series for just this purpose. The volumes will necessarily be somewhat independent, as they will target different domains, but there may still be some ways to leverage the unity of a categorical approach.

Another suggestion explored at the meeting is a repository of open documents, which other researchers can modify and reuse (e.g., Creative Commons). In addition to written exposition, such a repository could contain course materials such as slides and homework problems, as well as smaller, labor-intensive elements such as diagrams, visualizations and video content.

More ambitiously, such a repository could support a (non-linear) CT curriculum tied to a database of such materials, allowing users to build tailored, domain-specific introductions from available materials. This would also play a role in community development, with users (especially from new fields) working to develop new examples over time. In addition, some felt that it could be useful to target established platforms for online learning in order to encourage greater penetration and to handle concerns that are out of scope for ACT (e.g., accreditation).

Along with more targeted "on-ramp" introductions, these approaches would help to ease new-comers into the field, and show them right away how CT abstractions are relevant to their interests. As the curriculum for a particular domain is honed and improved, it may eventually transition to a textbook *a la* Coecke & Kissinger [18], using CT as a tool to teach the subject outright.

However, workshop participants also emphasized the need for alternative forms of exposition that are less familiar to mathematicians. Several noted that engineers are particularly fond of cookbooks, texts that implement a variety of concrete examples that readers can use as prototype solutions to their own problems. Similarly, taxonomies like Selinger's survey of graphical languages [40] can be extremely helpful for identifying exactly the right concept that a user might need for the purpose at hand. Over all, it is important to remember that mathematicians' style of definition-theorem-proof is rarely well-suited to other communities.

Another important consideration raised at **ACT-NIST** is the integration of computerized tools for manipulating categorical structures. The string diagrams pervasive in ACT are essentially dynamical structures; authors describe proofs in terms of "bending wires", "sliding boxes" and "yanking equations". Today learners (mostly, see section 5) engage these dynamics with their minds by looking at pictures on paper, but this will change as the available tools improve. In that case, many axioms and rules can be built in, preventing errors and allowing for more unstructured exploration and play. Furthermore, all but the most contrived examples in applied contexts are simply too big to describe, edit and maintain on paper; encoding larger and more complex examples in pedagogical works will help to demonstrate the legitimacy of these methods.

4.3 Timeline

1-2 yr.
"CT for X" monographs How-to & Cookbook Presentations
3-5 yr.
"CT for X" textbooks CC-licensed repository of pedagogical material
6-10 yr.
Tool-integrated curricula Capstone/project-based courses Modular curriculum assembly

5. Tool Support

For participants from industry, the most immediate barrier to ACT today is the gap between abstract theory and concrete computation. This includes the theoretical challenge of identifying efficient algorithms for calculating with categories, as well as a host of design and user interface issues involved in eliciting, manipulating and sharing them. Achieving the field's goals will require better methods for authoring, storing and editing categorical structures, and for using these structures to derive insights and implementations at the level of application. That said, categorical tools have the potential for major benefits in areas like formal verification and model-driven engineering.

5.1 Current Landscape

A list of CT-based tools solicited from **ACT-NIST** participants, augmented with projects from [41], is presented in Table 1.[2] Existing projects generally fall into one of three categories: proof assistance, specification and data manipulation. Most of these projects are academic, representing relatively small prototype projects. As a result, more work is needed to justify both the scalability and the usability of CT approaches.

Proof assistants, which help to check and manage the details of large proofs, are a natural target for mathematicians because they are directly useful *within* mathematics. Notably, Fields medalist Vladimir Voevodsky's interest in foundations stemmed from dissatisfaction with mathematicians' inability to verify purported proofs [42]. Based on ideas from (higher) category theory, these considerations led to homotopy type theory (HoTT, [43]), a new field of research connecting the foundations of geometry and computing. As one component in this development, the HoTT project has developed extensive libraries verifying the core elements of the theory inside the Coq type theory engine [44, 45]. Type theory has also been used to implement a range of other categorical constructions, in both Coq [46–49] and Idris [50].

[2] Representatives of unlisted projects are encouraged to contact the report's editors.

Coq is based on type theory, which is closely related to CT, but other proof assistants rely on categorical structures more directly. For example, Globular [51] and its successor, Homotopy.io [52], provide graphical representations of string diagrams (and higher-dimensional structures) that can be manipulated to construct and validate equational proofs between them. Opetopic [53] uses a different class of underlying structures (opetopes) towards a similar end. Quantomatic [54] and PyZX [55] use a more specific categorical representation called the ZX-calculus to formalize proofs about quantum protocols.

In categorical specification, CT structures are used to describe and (often) implement a variety of semantic artifacts. The earliest applications in this area were to software specification, especially the work of Goguen & Burstall [56], which would later lead to the Computational Category Theory library implemented in ML [57]. More recent examples of CT-based software specification include Specware [58] and CAP [59].

In software specification, the programming language acts as a semantic context for CT models, but other choices are also available. Cateno [60] and Catlab [61] target numerical linear algebra, providing high-level environments to express complex calculations in computer algebra. Semantic contexts can be scientific as well as computational; Quipper [62] and Matriarch [63] use CT to assemble, respectively, quantum circuits and hierarchical proteins. There are also several projects in development focused on CT specifications for concurrent systems, based on connections with Petri nets (Statebox [64]) and the π-calculus (Rholang [65]).

Another class of categorical software targets data storage and manipulation. One approach, pioneered in EASIK[3] [66], uses categories and functors to represent database schemas, states and queries. The Categorical Query Language [67] builds on this idea, using functorial relationships between schemas to implement data migrations and merges. From a rather different perspective, PySheaf provides a rigorous approach to multi-modal data fusion based on an abstract representation of the overlapping information between different data sources.

Finally, any discussion of categorical software should include a mention of functional programming (FP). This is the most significant existing application of CT by a wide margin, though we will only touch on the topic here. Compositional techniques (notably monads) lie at the heart of FP, and CT has had a strong influence on programming language design from the Categorical Abstract Machine [68] in the 1980's to Haskell, Scala and F# today [69–71]. More generally, functional and compositional techniques also form the basis for important applications outside traditional FP languages, such as the symbolic methods used in machine learning [72] and resource management in digital currencies [73].

Though FP and ACT are different communities, with different goals and scope, both would benefit from deeper interactions. For FP, ACT can provide connections to fields outside computing as well as new approaches for specifying and analyzing functional programs. For ACT, FP provides a larger, better established community with concrete successes to emulate (and failures to avoid), as well as a deep pool of potential recruits and collaborators to help develop the tools and methodology that the field requires. The grow-

[3]Entity-Attribute Sketch Implementation Kit

ing recognition of FP as a best practice for the design of complex, concurrent systems provides a perfect environment to test and validate ACT in the real world. Similar considerations also hold for other, smaller communities that embed elements of CT, including computer algebra [74] and graph re-writing [75].

Table 1. ACT software projects

Project	Description	Active?	Contact	Language	License
CAP	Categories, algorithms, and programming	Y	Sebastian Gutsche	GAP	GPL2
Cateno	Computational category theory	N	Jason Morton	Julia	?
Catlab	ACT computer algebra library	Y	Evan Patterson	Julia	2-clause BSD
CCT	Computational category theory	N	David Rydeheard	Standard ML	?
CQL	Categorical database management	Y	Ryan Wisnesky	Java	AGPL 3
CT in Coq 1	Representation and manipulation of CT terms	Y	John Wiegley	Coq	3-clause BSD
CT in Coq 2	Representation and manipulation of CT terms	Y	Amin Timany	Coq	?
CT in Coq (HoTT fork)	Representation and manipulation of CT terms	Y	Jason Gross	Coq	2-clause BSD
EASIK	Categorical database management	N	Robert Rosebrugh	Java	FreeBSD
Globular	(Higher) String diagram proof assistant	N	Jamie Vicary	Javascript	WTFPL
Homotopy.io	Proof assistant for higher categories	Y	Jamie Vicary	Javascript	CC-BY-ANC 3.0
HoTT	Homotopy Type Theory	Y	Bas Spitters	Coq, Agda	2-clause BSD, MIT

Table 1. (continued)

Project	Description	Active?	Contact	Language	License
idris-ct	Verified CT library	Y	Statebox	Idris	AGPL3
Matriarch	Hierarchical protein modeling	N	David Spivak	Python	CC-A-4.0
Opetopic	Proof assistant for higher categories	Y	Eric Finster	Scala	?
PySheaf	Sheaf library for data integration	Y	Michael Robinson	Python	?
PyZX	ZX-calculus for quantum calculations	Y	Aleks Kissinger	Python	GPL3
Quantomatic	ZX-calculus for quantum calculations	Y	Aleks Kissinger	Standard ML	GPL
Rholang	Concurrent programming	Y	Pyrofex	Scala	MIT
Specware	Categorical software specification	Y	Kestrel Inst.	Common Lisp	2-clause BSD
TikZit	Graphical editor for TikZ	Y	Aleks Kissinger	C++	GPL3
Typedefs	Programming language agnostic type definitions	Y	Jelle Herold	Idris	AGPL3
UniMath	Univalent Mathematics	Y	Daniel Grayson	Coq	Attribution

5.2 Roadmap

The **ACT-NIST** workshop identified several affordances that make computerized tools a prerequisite to the ultimate success of ACT. First and foremost, for many of the domains that ACT would like to target (e.g., systems engineering, data science) even small-scale problems are too big, with too many details, to manage with paper and pencil. Prototype implementations, at a minimum, are needed just to evaluate CT's utility in these areas. At the same time, better tools will multiply the productivity of ACT researchers across fields, especially by simplifying or automating the creation of useful representations (e.g., configuration files) from abstract structures. This includes the creation of both general infrastructure (databases, GUIs) and bespoke models (Bayes nets, dynamical systems).

Additionally, in the long run, computerized interaction will provide a powerful medium for learning about CT itself. Proofs using string diagrams are inherently dynamic–sliding boxes along wires–and computers can demonstrate this in a way that paper cannot. By maintaining "the rules of the game" computerized tools allow users to explore the space of possible constructions without introducing accidental errors, thereby reducing the field's reliance on "mathematical maturity". In principle, these tools could be used off-the-shelf without any knowledge of CT at all, but would reward deeper insight with more capabilities and greater flexibility. This would broaden exposure to the field and provide both an on-ramp and incentive to learn more.

Workshop participants identified a range of specific goals and requirements for future CT-based software. These clustered around issues of representation, transformation, computation and user interface, which we consider in turn.

To manipulate categorical structures on a computer, one must first design concrete data structures to represent them. Today each tool typically implements its own representations, but many participants urged the adoption of a more standardized approach. They argued that libraries of categorical data structures would ease developer workload and increase interoperability between categorical tools. Others cautioned that efficiency often requires a close match between algorithm and representation, and that the design of such general-purpose representations is extremely subtle. At a minimum, such an attempt should be prepared to fail and intend to iterate quickly. One proposal, very much in the categorical spirit, is to standardize the *description* of CT representations (sometimes called "doctrines"), rather than the representations themselves.

Regardless, a framework for managing transformations among representations will be required, since this is the only way to maintain backward compatibility with existing tools and data. Parsers to and from existing formats are a relatively straightforward means to expand the reach and value of ACT, especially in an industrial context. These would allow users to pull existing models and data into CT, and to integrate any results without needing to modify underlying infrastructure. The targets for such translations are extremely broad including formal modeling languages (OWL [76]), semi-formal modeling languages (UML/SysML [77, 78]), programming languages (Java, Haskell [69, 79]), database languages (SQL, RDF [80, 81]) and a host of domain-specific tools (e.g., in power systems, MatPower and GridLabD [82]). The same approach could also be used for model evolution and updating, by providing compositional transformations between successive iterations of a project.

Once models and data have been represented in CT, the next step is to calculate with them. This is itself an area for research. It is well-known that most of CT is constructive, meaning that mathematicians' definitions and proofs can (provably) be translated into data structures and algorithms. However, those with direct experience report that these translation usually need to be reworked with an eye towards efficiency. This suggests a substantial research program linking CT to complexity theory by mapping categorical constructions to existing algorithms.

Many at the workshop felt that the trickiest element in designing usable CT software, and certainly the farthest from mathematicians' expertise, is the development of graphical interfaces and interactions to present and elicit CT models. The importance of diagrams in CT is plain, and managing them in an intuitive way is a key requirement to expand the accessibility of CT to a broader audience.

One participant suggested that a "minimum viable product" in this area should target string diagrams: (i) creating and manipulating them graphically, (ii) reasoning about them equationally and (iii) mapping them to a variety of semantics like functions, relations, probabilities and matrices. However, other representations (particularly commutative diagrams) will also be needed, and there is work to be done sorting out the relationships between them.

One central question raised at **ACT-NIST** concerns the degree to which CT should be exposed to a tool's users. On one hand, even rudimentary knowledge of CT radically limits the base of potential users; for most, this is simply too esoteric a requirement. On the other hand, hiding CT from the user requires a mapping between domain concepts and categorical structures that is already a research project in itself.

To resolve this tension, some participants suggested a layered model for CT software, with mathematical structures designed by "CT gurus" at the top, and user-facing tools at the bottom that (can) insulate end-users from CT representations. In between would be a collection of "tools for the tool-smiths", providing an environment for researchers to build new tools by establishing CT-to-domain mappings. This would allow researchers to focus on high-level theoretical considerations rather than the low-level details of a practical implementation. To maximize developer efficiency, this layer should include libraries to implement both domain-specific analyses (e.g., Bayes nets) and generic infrastructure like databases and web servers.

A related perspective raised at the meeting suggests that category theory is better regarded as a platform than a product line, providing a common space for the interaction of many different types of data, models and programs. This relies on the fact that a categorical application is typically built up iteratively by defining and composing progressively more complicated structures (categories, functors, etc.). Whatever their initial purpose, the pieces that make up the construction can be reused for other goals.

This suggests that a CT-based platform ought to exhibit classical network effects and increasing returns to scale. However, this is something of a double-edged sword, as significant upfront effort will be required to develop initial applications without the benefit of a preexisting catalog of structures. Furthermore, this approach requires research on two separate fronts, addressing both platform design and application development simultaneously. Nonetheless, this represents a high-risk, high-reward strategy which could enable substantial efficiencies in the creation of complex models across a wide range of target domains.

5.3 Timeline

1-2 yr.
Identify tool(s) requirements & map to CT constructions Develop prototypes and existence-proof tools for end users Develop libraries for CT constructions Initial focus on documentation and representation
3-5 yr.
CT platform design Tool validation through domain-focused use cases
6-10 yr.
CT platform implementation DSL-based end-user applications Tools for tool developement ("toolsmithing")

6. Marketing & Technology Transfer

Applying CT techniques in commerce and government will require buy-in from a wide range of stakeholders, who will demand better explanations and better demonstrations of ACT's utility. At the **ACT-NIST** workshop, commercial participants and funders urged academics to emphasize *value* in the form of reduced costs and new capabilities. Moreover, the community must recognize that the introduction of these techniques will involve years of validation and vetting, including explicit metrics for success. In this section we consider the transition from interesting idea to realized value.

6.1 Current Landscape

For workshop participants coming from government and industry, most academic work spends too much energy demonstrating why the mathematics works and too little on the corollaries and context that explain what it is good for. More-than-toy examples and quantitative benchmarking are rare, and several participants expressed a general confusion as to what sorts of problems CT is useful for dealing with.

CT is a rich topic, and it can be difficult to choose which elements to highlight and which to suppress. Workshop participants suggested a range of potential selling points, listed in Table 2. We will not attempt to justify these here, though each of the points deserves further elaboration. However, all agreed that the case for these benefits would be much stronger with better tools (Section 5) for linking categorical concepts to related areas in functional programming and data science.

One useful perspective distinguishes 'hard' and 'soft' benefits. Hard benefits correspond to measurable improvements or new capabilities, like the speed-up of an algorithm or a new method for consistency-checking. Soft benefits, on the other hand, flow from improved cognitive or conceptual models, and might include fewer mistakes or easier mainte-

Table 2. Potential virtues of category theory.

Precision	Formalization in CT helps to anchor slippery terminology and root out hidden assumptions.
Unification	CT identifies, analyzes and implements *a priori* unrelated constructions and analyses through the use of universal design patterns (e.g., monads).
Diagrammatics	CT has a collection of diagrammatic languages which provide precise, formal representations similar to the diagrams that engineers and scientists already draw.
Inter-disciplinarity	CT provides a *lingua franca* to align conceptual models and formal relationships between different domains in science, engineering and design.
Multiplicity	The “relative point of view” in CT provides a methodology for aligning models and data across multiple viewpoints (e.g., multi-scale, multi-physics, etc.).
Expressivity	Categories can interpret a wide range of existing representations including symbolic logic, programming languages, graphs and matrices/tensors.
Constructivity	Most of CT is constructive, meaning that definitions and theorems can be translated into algorithms and code.
Inter-operability	CT’s emphasis on transformation and isomorphism allows for bottom-up integration without relying on centralized planning or control.
Evolution	CT models can be updated and evolved through the use of mappings that connect earlier and later versions of a representation.

nance. It is clear from Table 2 that CT's perceived advantages skew towards soft benefits. Although these are no less valuable, they are harder to quantify, and this is an obstacle to justifying CT adoption. Consequently, it may be useful to focus more explicitly on the down-stream benefits that flow from these conceptual shifts.

In many sectors labor inputs are a primary driver of cost and expertise is a binding constraint, so the value of improved productivity is high. Sometimes these improvements can emerge directly from better understanding (e.g., fewer mistakes), but more visible examples arise from a better alignment between human understanding and formal representations. As a concrete example, one participant pointed to the use of types in a functional programming language to improve programmers' efficiency by catching errors, cutting down "boilerplate" code and structuring code manipulations (refactoring).

There are currently a number of research groups working to explore CT's potential in the commercial sphere. These are split between CT-focused startups and exploratory projects at more established firms. Examples from both groups are collected in Table 3; see the linked websites, descriptions and contacts for additional information.

Though they are less established than the academic centers of CT, all agreed that commercial research groups are of immense value for ACT because they are simply nearer to applications than most university environments. They offer an opportunity to establish small-scale collaboration between domain specialists and mathematicians, as discussed in Section 3. In particular, several of the groups listed offer summer internships for mathematics students who are interested in learning more about industry.

6.2 Roadmap

With some notable exceptions, most of today's ACT research terminates in papers and proofs of concept. Discussion in this area focused on the challenge of moving an idea from that point to practical application in the day-to-day business of engineers, designers and analysts.

The move from research insight to realized value takes time, and must progress through a series of incremental expansions. Many of the domains that ACT hopes to target are dominated by large enterprises, and participants from those institutions advised that any application in this context will likely begin with an exploratory project. These would include internal teams, industry-university collaborations, contracted work and consulting. One significant friction that was identified for commercializing CT is the difficulty of targeting exploratory proposals to interested parties within enterprise. Better indexing of ACT-aligned and interested researchers, especially in industry, would help a great deal.

Regardless of the details, several participants stressed the importance of *managing expectations*. It is much easier to anticipate the theoretical possibilities of an application than to realize them in practice, and it can be tempting to promise too much in the early stages of development. It is better to focus on small, incremental advances that indicate directions for future growth.

They also emphasized the need for exploratory projects to identify (or develop) specific

Table 3. Commercial ACT research and development

	Company	Description	CT Contact(s)
Startups	Cambridge Quantum Computing	Quantum Computing	Ross Duncan
	Conexus	Categorical data management	Ryan Wisnesky
	OICOS	Economic and social systems modeling	Viktor Winschel
	Pyrofex	Distributed computing and cryptocurrency	Mike Stay
	Protocol Labs	Distributed computing and cryptocurrency	David Dalrymple
	Statebox	Compositional diagrammatic programming	Jelle Herold

	Company	Project(s)	CT Contact(s)
Established	Airbus	Light-weight meta-modeling	Dominique Ernadote
	BAE Systems	Software analysis	Steve Huntsman
	Dassault Systèmes	Multi-scale systems (unpublished)	Philippe Belmans
	Honeywell Aerospace	Systems of systems, temporal logic	Alberto Speranzon
	Kestrel Institute[a]	Software specification and analysis	Douglas Smith
	Metron	Network design and tasking	Christopher Boner
	Rolls Royce	Surface texture modeling	Qunfen Qi[b]
	Siemens	Cooperative Robotics (unpublished)	Arquimedes Canedo

[a]Kestrel is a non-profit entity.
[b]Dr. Qi led this research, but is not affiliated with Rolls Royce.

metrics of performance in order to demonstrate *measurable* improvements in dimensions of interest. In particular, formal proof is often unconvincing for engineers, who tend to prefer simulation and empirical validation. Some felt that this challenge may require the ACT community to forge deeper ties with social-science communities like education and management in order to benefit from their experience evaluating 'soft' benefits like (some of) those for CT.

Several participants cautioned against orthodoxy and an insistence on mathematical purity, arguing that earlier efforts to apply categorical ideas had foundered in part due to inflexibility. In contrast to a foundational approach, practical CT applications will need to sit atop an unruly stack of proprietary technologies. Businesses and their employees are deeply invested in existing tooling, and ACT will need to work with these products in order to achieve success.

The discussion also noted another potential point of friction between the academic and commercial communities in ACT: the role of intellectual property. Typically, abstract ideas like mathematical structures are not eligible for patent protection, but the applications designed with them are. Many mathematicians argued for an open-source approach to product development, pointing to the recent success of (open) Jupyter notebooks over (proprietary) Mathematica; indeed, Kestrel Institute has recently open sourced their Specware package to spur broader usage. However, not all companies will be amenable to this approach, as it may undercut competitive advantage or require additional development prior to release.

6.3 Timeline

1-2 yr.
Exploratory projects with industry partners Clarify ACT goals and capabilities Directory and documentation of successful projects Identify/develop early metrics of success
3-5 yr.
Live testing in production situations Establish benchmarks and methodology for CT applications Scale up use cases and demonstrations
6-10 yr.
Deployment at scale CT semantics for standards, contracts

7. Funding

Realizing the vision described in this report will require substantial investment from both public and commercial entities. As with most early-stage research, the bulk of ACT work to date has been supported by government investment, representing a wide spectrum of organizations and interests. To develop further, the field will need to cultivate sources of internal funding for research and development within industry.

7.1 Current Landscape

Pragmatically speaking, there are a wide range of potential funding sources to support early-stage ACT research. Much of ACT work today is funded by the mathematics branches of government funding agencies[4] like the National Science Foundation (NSF). If CT is viewed as *pure* math, these are among the only sources available, but workshop participants identified a much wider range of potential resources opened up by the move toward applications.

Other branches of scientific funding agencies provide one group of targets. For example, the NSF's Systems Science program funded a recent post-doctoral position at NIST studying applications of CT in power systems. One participant suggested that the mathematical rigor of a new "CT for *X*" project (Section 3) may be more compelling for domain practitioners than the theoretical insights are for mathematicians. Not every new application will lead to a new theorem (though they often do), and for practical applications this is usually the wrong metric for success. Moreover, the necessary mathematical advances are often difficult to anticipate before establishing a firm grounding in the domain.

It is not only other branches of academic funding agencies that are opened by the move to applications, but entirely new institutions as well. An (incomplete[5]) list of ACT-aligned researchers in a variety of government agencies is included in Table 4. NIST has been a leader in this area, hosting several community workshops [3, 6] as well as funding several internal research projects and a series of small business investment grants for the development of categorical software. Others include the Pacific Northwest National Lab (PNNL), which has conducted internal research on data integration using CT, and the National Aeronautics and Space Administration (NASA)), which has funded outside research in categorical applications for critical safety analysis.

In addition to civilian applications, military agencies are also investing in ACT. Significant interest has centered around quantum foundations, a topic of concern for researchers and funders at the Air Force Office of Scientific Research (AFOSR), the Army Research Lab (ARL) and the Naval Research Lab (NRL). Other areas of application which have benefitted from military funding include formal verification (AFOSR) and the analysis of nonlinear dynamical systems (Air Force Research Lab). Notably, AFOSR has funded two 5-year Multi-University Research Initiatives (MURIs) involving elements of CT, focused on homotopy type theory (2014, [83]) and semantics and formal reasoning for quantum programming (2015, [84]).

Most significantly, recent interest from DARPA has provided a significant visibility boost for the field. Recent programs including SIMPLEX, CASCADE and FunDesign

[4]This document will focus only on US agencies, but the remarks apply more broadly.

[5]Other ACT-interested government researchers, in the US or elsewhere, are encouraged to contact the report's authors.

Table 4. Government ACT research and development

Organization	Project(s)	CT Contact(s)
Air Force Office of Scientific Research	Quantum foundations & Formal verification	Tristan Nguyen
Air Force Research Lab	Nonlinear Dynamical Systems	Jared Culbertson
Army Research Lab	Quantum foundations	Joseph Myers Radhakrishnan Balu
Defense Advanced Research Projects Agency	Complex adaptive systems & Conceptual design	John Paschkewitz, Jan Vandenbrande
National Aeronautics and Space Administration	Critical safety analysis	Alwyn Goodloe
National Institute for Standards and Technology	Interoperability & Systems Modeling	Spencer Breiner, Blake Pollard
Naval Research Lab	Quantum foundations	Keye Martin
Pacific Nortwest National Lab	Heterogeneous data integration	Emilie Purvine, Cliff Joslyn

[85–87] have involved elements of category theory, both in the call for proposals and the eventual funding awards. In contrast to the work mentioned above, most of which was either internal or university-focused, these calls were open to industry participants, encouraging a number of commercial firms to invest in categorical and related expertise (e.g., algebraic topology).

As the list of workshop participants indicates (p. iii), a number of businesses ranging from start-ups to multinationals have begun to explore potential applications for CT. However, some have found it difficult to build the business case needed to convince their own superiors that this is an area worthy of investment; see Section 6 for more on this topic.

7.2 Roadmap

Looking forward, all agreed that the community would benefit from better and more open strategies for identifying new funding opportunities, as well as pipelines for growing from one to the next. Some types of information that might reduce friction and improve discovery include:

- CT contacts within governmental and corporate organizations
- Domains of interest
- Funding types (e.g., seed grant, SBIR, MURI, etc.)
- Application requirements
- Metrics of success

As in Section 2, a central online nexus for ACT would make it easier to broadcast new funding and employment opportunities, but there is a bootstrap problem of generating enough content to draw readers. In addition to upcoming opportunities, others thought it would also be helpful to collect information about successful funding applications and their outcomes. This would help to map how various CT projects succeed or fail over time.

To realize the longer-term goals raised at the meeting, some feel that a digital nexus may not be enough; a network of established research groups and centers of excellence focused on ACT are also needed. Face-to-face interaction provides much deeper communication than any remote channel and this is critically important for exchanging complex ideas, especially across linguistic and community boundaries. Short-term visits form the bonds of the community while chairs and appointments provide space for long-term investment.

For those within the funding community, a first step is to strengthen ties between agencies, both within the US and across borders. Whereas the value of most CT projects today is considered in isolation, this would allow for alignment between projects to maximize coverage and reuse. This would be especially important for the more ambitious platform-style model of CT tool support discussed in Section 5, which world require both systems-level engineering to design and domain-specific deep dives to validate.

Along similar lines, several participants argued that funding from industry could be both increased and coordinated by the creation of an industry consortium to develop ACT. This would help to spread development costs, sidestep intellectual property issues and speed the spread of best practices as ACT gets off the ground. Under the proposed system, each participating company would contribute an annual fee that would go to fund ACT research. In return, they would obtain access to the intellectual and computational outputs of all the consortium's research. In addition, the consortium could provide a nexus for defining and developing new ACT projects to be seeded from annual fees before transitioning to a contract model for larger deployment.

7.3 Timeline

1-2 yr.
Exploratory grants (NSF, NASA) Small business grants (NIST) Course-development grants (NSF)
3-5 yr.
Multi-University Research Initiatives (AFOSR, ARL) Industry-University Collaborations (NSF) Focused research programs (DARPA)
6-10 yr.
Industrial consortium (proposed)

References

[1] Eilenberg S, MacLane S (1945) General theory of natural equivalences. *Transactions of the American Mathematical Society* 58(2):231–294.

[2] Abramsky S, Baez JC, Gadducci F, Winschel V (2014) Categorical Methods at the Crossroads (Dagstuhl Perspectives Workshop 14182). *Dagstuhl Reports* 4(4):49–63. https://doi.org/10.4230/DagRep.4.4.49. URL http://drops.dagstuhl.de/opus/volltexte/2014/4618

[3] Breiner S, Subrahmanian E (eds) (2015) *Computational Category Theory*, (National Institute of Standards and Technology). Slides available at http://www.appliedcategorytheory.org/type/link/.

[4] Abramsky S, Hardy L, Mislove M (eds) (2016) *Compositionality*, (Simons Institute). Slides and video available at https://simons.berkeley.edu/workshops/schedule/3440.

[5] Baez J (ed) (2017) *Applied Category Theory*, (AMS Special Session). Slides and video available at http://math.ucr.edu/home/baez/ACT2017/.

[6] Breiner S, Pollard B, Subrahmanian E (eds) (2018) *Applied Category Theory: Bridging Theory and Practice*, (National Institute of Standards and Technology). Slides and video available at http://www.appliedcategorytheory.org/nist-workshop-slides/.

[7] Coecke B, Fong B, Kissinger A, Lewis M, Tan J (eds) (2018) *Applied Category Theory: Towards an Integrative Science*, (Lorentz Center). Slides and video available at https://johncarlosbaez.wordpress.com/2018/04/30/applied-category-theory-2018-live-stream/.

[8] Vicary J, et al (eds) (2018) *Symposium on Compositional Structures*, (University of Birmingham). Website: http://www.cs.bham.ac.uk/~vicaryjo/syco/1/.

[9] Baez J, Coecke B, Hess K, Lack S, de Paiva V (eds) (2018) *Compositionality*, (TBD). Website: http://www.compositionality-journal.org/.

[10] Lawvere FW, Schanuel SH (2009) *Conceptual mathematics: a first introduction to categories* (Cambridge University Press), .

[11] Spivak DI (2014) *Category theory for the sciences* (MIT Press), 1st Ed.

[12] Awodey S (2010) *Category theory* (Oxford University Press), .

[13] Riehl E (2017) *Category theory in context* (Courier Dover Publications), .

[14] Leinster T (2014) *Basic category theory*. Vol. 143 (Cambridge University Press), .

[15] Smith P (2018) Category Theory: Lecture Notes and Online Books, https://www.logicmatters.net/categories/. Accessed: 2019-06-26.

[16] Fong B, Spivak DI (2019) *An Invitation to Applied Category Theory: Seven Sketches in Compositionality* (Cambridge University Press), .

[17] Bradley TD (2018) What is Applied Category Theory? *arXiv preprint arXiv:180905923* .

[18] Coecke B, Kissinger A (2017) *Picturing quantum processes* (Cambridge University Press), .

[19] Baez J (1993-2010) This Week's Finds in Mathematical Physics, http://math.ucr.edu/home/baez/TWF.html. Accessed: 2019-06-26.

[20] Baez J, Corfield D, Schreiber U, et al. (2006-present) The *n*-Category Café, https://golem.ph.utexas.edu/category/. Accessed: 2019-06-26.

[21] Baez J, et al. (2010-present) The Azimuth Blog, https://johncarlosbaez.wordpress.com/. Accessed: 2019-06-26.

[22] Yuan Q (2009-2018) Annoying Precision, https://qchu.wordpress.com/. Accessed: 2019-06-26.

[23] Sobocinski P (2015-present) Graphical Linear Algebra, https://graphicallinearalgebra.net/. Accessed: 2019-06-26.

[24] Morton J (2007-present) Theoretical Atlas, https://theoreticalatlas.wordpress.com/. Accessed: 2019-06-26.

[25] Bradley TD (2015-present) Math3ma, https://www.math3ma.com. Accessed: 2019-06-26.

[26] Milewski B (2008-present) Bartosz Milewski's Programming Cafe, https://bartoszmilewski.com/. Accessed: 2019-06-26.

[27] Milewski B (2017) Category Theory for Programmers, https://bartoszmilewski.com/2014/10/28/category-theory-for-programmers-the-preface/. Accessed: 2019-06-26.

[28] Milewski B (2016) Category Theory, https://www.youtube.com/playlist?list=PLbgaMIhjbmEnaH_LTkxLI7FMa2HsnawM_. Accessed: 2019-06-26.

[29] Maggiore G (2017) Category Theory for Typescript Programmers, https://blog.usejournal.com/category-theory-for-typescript-programmers-a2372f771b20. Accessed: 2019-06-26.

[30] Yokota E (2012) learning Scalaz–Category theory, http://eed3si9n.com/learning-scalaz/. Accessed: 2019-06-26.

[31] Seeberger H (2010) Introduction to Category Theory in Scala, https://hseeberger.wordpress.com/2010/11/25/introduction-to-category-theory-in-scala/. Accessed: 2019-06-26.

[32] Louc S (2017) Demistifying the Monad in Scala, Part 2: A Category Theory Approach, https://medium.com/@sinisalouc/demistifying-the-monad-in-scala-part-2-a-category-theory-approach-2f0a6d370eff. Accessed: 2019-06-26.

[33] Tsarpalis E (2017) Programming in the Point-Free Style, https://eiriktsarpalis.wordpress.com/2017/04/02/programming-in-the-point-free-style/. Accessed: 2019-06-26.

[34] Yan D (2017) Category Theory via C#, https://weblogs.asp.net/dixin/Tags/Category%20Theory. Accessed: 2019-06-26.

[35] *n*Lab contributors (2009-present) The *n*Lab, https://ncatlab.org/nlab/show/HomePage. Accessed: 2019-06-26.

[36] Meadors C (2018) The *m*Lab, https://cemulate.github.io/the-mlab/. Accessed: 2019-06-26.

[37] MathOverflow Contributors (2009-present) MathOverflow, https://mathoverflow.net/questions/tagged/ct.category-theory. Accessed: 2019-06-26.

[38] Cheng E, Willerton S (2007-2014) The Catsters, https://www.youtube.com/user/TheCatsters/featured. Accessed: 2019-06-26.

[39] Cheng E (2015) *How to Bake π* (Basic Books), 1st Ed.

[40] Selinger P (2010) A survey of graphical languages for monoidal categories. *New structures for physics*, ed Coecke B (Springer), , , pp 289–355.

[41] Tan J (2018) Comparison of category theory software, http://www.joshuatan.com/a-comparison-of-category-theory-software/. Accessed: 2019-06-26.

[42] Voevodsky V The Origins and Motivations of Univalent Foundations, https://www.ias.edu/ideas/2014/voevodsky-origins. Accessed: 2019-06-26.

[43] Program UF (2013) *Homotopy type theory: Univalent foundations of mathematics* (Univalent Foundations), .

[44] Voevodsky V, Ahrens B, Grayson D, et al. (2017) UniMath: Univalent Mathematics. *URL: https://github com/UniMat* .

[45] Bauer A, et al. (2017) The HoTT library: a formalization of homotopy type theory in Coq. *Proceedings of the 6th ACM SIGPLAN Conference on Certified Programs and Proofs* (ACM), pp 164–172.

[46] Wiegley, John (2018–2019) Category Theory in Coq, https://github.com/jwiegley/category-theory.

[47] Timany A, Jacobs B (2016) Category theory in Coq 8.5. *1st International Conference on Formal Structures for Computation and Deduction*, *LIPIcs. Leibniz Int. Proc. Inform.*, Vol. 52 (Schloss Dagstuhl. Leibniz-Zent. Inform., Wadern), , , pp Art. No. 30, 18.

[48] Carvalho A, Mateus P (1998) *Category theory in COQ*. Ph.D. thesis. Diploma Thesis. Institute Superior Técnico, Universidade Técnica de Lisboa, .

[49] Gross J, Chlipala A, Spivak DI (2014) Experience implementing a performant category-theory library in Coq. *International Conference on Interactive Theorem Proving* (Springer), pp 275–291.

[50] Perone, Marco and others (2019) idris-ct, https://github.com/statebox/idris-ct.

[51] Bar K, Kissinger A, Vicary J (2018) Globular: an online proof assistant for higher-dimensional rewriting. *Log Methods Comput Sci* 14(1):Paper No. 8, 16.

[52] Reutter D, Vicary J (2019) High-level methods for homotopy construction in asso-

ciative *n*-categories. *arXiv preprint arXiv:190203831* To appear: Logic in Computer Science 2019.
[53] Finster, Eric (2015-2019) Opetopic, address.
[54] Kissinger A, Zamdzhiev V (2015) Quantomatic: A proof assistant for diagrammatic reasoning. *International Conference on Automated Deduction* (Springer), pp 326–336.
[55] Kissinger A, van de Wetering J (2019) Pyzx: Large scale automated diagrammatic reasoning. *arXiv preprint arXiv:190404735* To appear: Quantum Physics and Logic 2019.
[56] Burstall RM, Goguen JA (1980) The semantics of Clear, a specification language. *Abstract Software Specifications* (Springer), pp 292–332.
[57] Rydeheard DE, Burstall RM (1988) *Computational category theory*. Vol. 152 (Prentice Hall Englewood Cliffs), .
[58] Srinivas YV, Jüllig R (1995) Specware: Formal support for composing software. *International Conference on Mathematics of Program Construction* (Springer), pp 399–422.
[59] Gutsche, Sebastian and Posur, Sebastian and Barakat, Mohamed and Skartsæterhagen Øystein (2013–2019) Categories, Algorithms and Programming, https://homalg-project.github.io/CAP_project/.
[60] Morton, Jason (2015–2019) Cateno, https://github.com/jasonmorton/Cateno.
[61] Patterson, Evan (2017–2019) Catlab, https://github.com/epatters/Catlab.jl.
[62] Green AS, Lumsdaine PL, Ross NJ, Selinger P, Valiron B (2013) Quipper: a scalable quantum programming language. *ACM SIGPLAN Notices*, Vol. 48.6 (ACM) Vol. 48.6, pp 333–342.
[63] Giesa T, Jagadeesan R, Spivak DI, Buehler MJ (2015) Matriarch: a python library for materials architecture. *ACS biomaterials science & engineering* 1(10):1009–1015.
[64] Team S, Genovese F (2019) The Mathematical Specification of the Statebox Language. *arXiv preprint arXiv:190607629* .
[65] Pyrofex (2019) Rholang, https://rholang.org/.
[66] Rosebrugh R, et al. (2009) EASIK: An EA-Sketch implementation kit, address.
[67] Spivak DI, Wisnesky R (2015) Relational foundations for functorial data migration. *Proceedings of the 15th Symposium on Database Programming Languages* (ACM), pp 21–28.
[68] Cousineau G, Curien PL, Mauny M (1985) The categorical abstract machine. *Conference on Functional Programming Languages and Computer Architecture* (Springer), pp 50–64.
[69] Marlow S, et al. (2010) Haskell 2010 language report, http://www.haskell.org/(May2011). Accessed: 2019-06-26.
[70] Odersky M, et al. (2018) Scala Language Specification, https://www.scala-lang.org/files/archive/spec/2.12/. Accessed: 2019-06-26.
[71] Syme, Don and others (2016) The F# Language Specification, https://fsharp.org/specs/language-spec/.

[72] Chollet, Francois and others (2015–2019) Keras, https://github.com/keras-team/keras.

[73] Libra Association (2019) An Introduction to Libra, https://libra.org/en-US/white-paper/#introduction.

[74] Ellis G (2008) Homological algebra programming. *Computational group theory and the theory of groups* 470:63–74.

[75] Lack S, Sobociński P (2004) Adhesive categories. *International Conference on Foundations of Software Science and Computation Structures* (Springer), pp 273–288.

[76] McGuinness DL, Van Harmelen F, et al. (2004) OWL web ontology language overview. *W3C recommendation* 10(10):2004.

[77] Object Management Group (2017) Unified Modeling Language version 2.5, https://www.omg.org/spec/UML/About-UML/.

[78] Object Management Group (2017) System Modeling Language Version 1.5, https://www.omg.org/spec/SysML/1.5/.

[79] Gosling J, Joy B, Steele G, Bracha G (2000) *The Java language specification* (Addison-Wesley Professional), .

[80] Joint Technical Committee 1 for Information Technology (2016) *The SQL Standard (ISO/IEC 9075:2016)* (International Standards Organization), .

[81] World Wide Web Consortium (2004) Resource Description Framework, https://www.w3.org/TR/2004/REC-rdf-concepts-20040210/.

[82] Pollard, Blake and Nolan, Jack and Breiner, Spencer and Subrahmanian, Eswaran (2019) Compositional Models for Power Systems. Under review.

[83] Awodey S, et al. (2014) Homotopy Type Theory:Unified Foundations of Mathematics and Computation, https://hottheory.files.wordpress.com/2014/04/proposalpublic2.pdf. Program Manager: Tristan Nguyen.

[84] Mislove M, et al. (2015) Semantics, Formal Reasoning, and Tool Support for Quantum Programming, http://cs.tulane.edu/quantum/. Program Manager: Tristan Nguyen.

[85] Defense Advanced Research Projects Agency DSO (2014) DARPA-BAA-14-59, Simplifying Complexity in Scientific Discovery (SIMPLEX), Broad Agency Announcement. Available at https://govtribe.com/opportunity/federal-contract-opportunity/simplifying-complexity-in-scientific-discovery-simplex-darpabaa1459. Program Manager: John Paschkewitz.

[86] Defense Advanced Research Projects Agency DSO (2015) DARPA-BAA-16-11, Complex Adaptive System Composition And Design Environment (CASCADE), Broad Agency Announcement. Available at https://govtribe.com/opportunity/federal-contract-opportunity/complex-adaptive-system-composition-and-design-environment-cascade-darpabaa1611-1. Program Manager: John Paschkewitz.

[87] Defense Advanced Research Projects Agency DSO (2017) DARPA-SN-17-71, Fundamental Design (FUN DESIGN), Special Notice. Available at https://govtribe.com/opportunity/federal-contract-opportunity/disruption-opportunity-special-notice-fundamental-design-fun-design-darpasn1771. Program Manager: Jan Vandenbrande.

Appendix A: Supplemental Materials

The supplemental materials can be found online at http://www.appliedcategorytheory.org/nist-workshop-slides/. These include slides and videos for most of the talks given at the workshop, though due to an unfortunate technical error Wisnesky's, Huntsman's and (part of) Pavlovic's talks were not recorded.