Check for updates

NIST Interagency Report
NIST IR 8554

Workshop on Compositional Structures in Systems Engineering and Design

Spencer Breiner
Joe Moeller
Priyaa Varshinee Srinivasan
Eswaran Subrahmanian

This publication is available free of charge from:
https://doi.org/10.6028/NIST.IR.8554

NATIONAL INSTITUTE OF STANDARDS AND TECHNOLOGY
U.S. DEPARTMENT OF COMMERCE

NIST Interagency Report
NIST IR 8554

Workshop on Compositional Structures in Systems Engineering and Design

Spencer Breiner
Joe Moeller
Priyaa Varshinee Srinivasan
Eswaran Subrahmanian

This publication is available free of charge from:
https://doi.org/10.6028/NIST.IR.8554

December 2024

U.S. Department of Commerce
Gina M. Raimondo, Secretary

National Institute of Standards and Technology
Laurie E. Locascio, NIST Director and Under Secretary of Commerce for Standards and Technology

Certain commercial entities, equipment, or materials may be identified in this document in order to describe an experimental procedure or concept adequately. Such identification is not intended to imply recommendation or endorsement by the National Institute of Standards and Technology, nor is it intended to imply that the entities, materials, or equipment are necessarily the best available for the purpose.

NIST Technical Series Policies

Copyright, Fair Use, and Licensing Statements
NIST Technical Series Publication Identifier Syntax

Publication History

Approved by the NIST Editorial Review Board on 2024-12-19

How to cite this NIST Technical Series Publication:

Spencer Breiner, Joe Moeller, Priyaa Varshinee Srinivasan, Eswaran Subrahmanian (2024) Workshop on Compositional Structures in Systems Engineering and Design. (National Institute of Standards and Technology, Gaithersburg, MD), NIST IR 8554. https://doi.org/10.6028/NIST.IR.8554

NIST Author ORCID iDs

Spencer Breiner: 0000-0001-5400-1349
Joe Moeller: 0000-0002-5306-3502
Priyaa Varshinee Srinivasan: 0000-0001-6983-073X
Eswaran Subrahmanian: 0000-0002-4639-627X

Contact Information

spencer.breiner@nist.gov

Abstract

The focus of this workshop is to foster the development of applications of category theory to systems engineering and design. We are interested in bringing together experts from engineering and applied category theory, and those involved in software tools development. There were be several talks/demos showcasing existing approaches as well as ample time for discussion.

Keywords

Category theory; compositionality; system design; systems engineering.

Table of Contents

List of Figures

Acknowledgments

We would like to thank Janet Madison, Gayaletchimy Suppaya, Pauline Truong, and Tim Czakoczi for their assistance in putting this workshop together and making sure it ran smoothly.

Workshop Participants

Angeline Aguinaldo; UMD

Jon Aytac; Sandia National Labs

Georgios Bakirtzis; UT Austin

Chandler Becker; NIST

Spencer Breiner; NIST

Kristopher Brown; Topos Institute

Francisco Javier Calvo Amodio; OSU

Robert Cardona; SUNY Albany

Jacob Collard; NIST

Jared Culbertson; AFRL

Justin Curry; SUNY Albany

Matthew Daniels; NIST

Peter Denno; NIST

Zinoviy Diskin; McMaster University

James Fairbanks; University of Florida

Fabrizio Romano Genovese; 20squares

Robert Green; SUNY Albany

Tyler Hanks; University of Florida

Julian Hedges; University of Strathclyde

Brian Heller; SUNY Albany

Alan Hylton; NASA

Philip Johnson-Freyd; Sandia National Labs

Albert Jones; NIST

Matthew Klawonn; AFRL

Mieczyslaw Kokar; VIStology, Inc.

Pascal Le Masson; Mines ParisTech

Frederick Leve; AFOSR

Darrell Lewis-Sandy; LinkedIn

Benjamin Long; NIST

Jesus Lopez; Chevron

Owen Lynch; Topos Institute

Joe Moeller; NIST

Juan Esteban Montero; Chevron

Luke Morris; University of Florida

Tristan Nguyen; AFOSR

Evan Patterson; Topos Institute

David Perner; UAH

Blake Pollard; Crossnokaye

Jose David Porter Medina; OSU

Daniel Rosiak; CWRU

David Rousseau; Ctr for Systems Philosophy

Joshua Shinavier; LinkedIn

Pawel Sobocinski; TalTech

David Spivak; Topos Institute

Ashok Srinivasan; NSF

Priyaa Varshinee Srinivasan; NIST

Ram D. Sriram; NIST

Michael Stone

Eswaran Subrahmanian; NIST/CMU

Donald Thompson; LinkedIn

Marquia Williams; SUNY Albany

Derek Wise; Lockheed Martin

Ryan Wisnesky; Conexus AI

Fig. 1. Group Photo

1. Introduction

The ”Compositional Structures for Systems Engineering and Design” workshop was hosted by NIST at the National Center for Cyber Security and Excellence (NCCoE) from Nov 3 - 4, 2022.

The overarching **theme** of the workshop is to explore different aspects of using methods and tools from Category Theory to Systems Engineering and Design - from scientific to social. We broadly categorize such aspects under *Theory, Applications, Programming, and Ecosystem*, collectively referred to by the acronym TAPE. Aligning to the theme of the workshop, the participants were drawn from multidisciplinary backgrounds - engineers, scientists, graduate students, business leaders from academia, industry and government research labs.

The **objectives** of the workshop were to:

- Facilitate and initiate communication among different players in the cross-disciplinary field of ‘Compositional Systems Engineering’ – application of Category Theory to Systems Engineering.
- Enable knowledge dissemination from Applied Category Theory to Systems Engineering and vice versa.
- Enable discussion on the existing tools, their use cases as well as the development of new tools to model and analyze systems.
- Initiate conversations on the development of a consortium of industry, academia, and research labs to foster research and innovation in Compositional Systems Engineering.

The **program** of the workshop were centered around these objectives. The program included tutorial talks (1 hour), invited talks (45 minutes), short talks (20 minutes), software demo sessions (30 minutes), structured and unstructured discussion times. The program also accounted for the theme of the workshop by having a good mix of speakers from applied category theory and systems engineering. The workshop also had a few speakers present talks on the social aspects concerning interaction of category theorists and engineers. See Figure 2 for the time distribution for different types of sessions. See Appendix A for the schedule of the workshop.

Speakers for each of the sessions are listed below. Slides for many of the talks can be found in the supplementary materials for the report.

Tutorials: David Spivak (Topos Institute), Fabrizio Romano (20 Squares)

Invited talks: Pawel Sobocinski (TalTech), Alan Hylton (NASA), Georgios Bakirtzis (UT, Austin), Jules Hedges (U. of Strathclyde), Zinovi Diskin (McMaster University), Derek Wise (Lockheed Martin)

Type of session	Duration in mins	# of sessions	Total duration	percentage	
Tutorials	60	2	120	12.9%	
Invited	45	6	270	29.0%	
Short talks	20	5	100	10.8%	
Lunch	60	2	120	12.9%	
Break 1	20	2	40	4.3%	
Break 2	20	2	40	4.3%	
Discussion	60	2	120	12.9%	
Presentation	30	1	30	3.2%	
Introduction	15	1	15	1.6%	
Demo	30	2	60	6.5%	
				98.4%	

Fig. 2. Distribution of time across various sessions

Short talks: Esteban Montero (Chevron), Angeline Aguinaldo (UMD, AFRL), Pascal Le Masson (Mines Paris PSL University), David Perner (UAH), Donald Thompson (Microsoft), Priyaa Varshinee Srinivasan (NIST), Daniel Rosiak (CWRU)

Demo: Ryan Wisnesky (Conexus), Evan Patterson (Topos Institute)

The workshop was organized by Joseph Moeller (NRC postdoc) and Priyaa Varshinee Srinivasan (Intl. Associate) from the Information Technology Lab with support and advice from Spencer Breiner and Eswaran Subramanian. Dr. Ram Sriram, the Division Chief provided travel support for student attendees of the workshop.

2. Compositional Systems Engineering and Design

2.1. Theory

Monoidal categories are the most widely used mathematical structures in applied category theory, providing a framework for modeling systems that can combine processes both sequentially and in parallel. String diagrams, a graphical language similar to flowcharts, are used to visualize processes in monoidal categories, and have proven to be an effective tool for performing calculations. Monoidal categories and string diagrams have been applied in fields such as quantum computing and natural language processing. Genovese's tutorial discussed diagrammatic methods and interpreting monoidal categories as systems and process theories.

In network and system modeling, category theory has been used to analyze complex systems by representing their components in a monoidal category. Open systems, which have external ports that can interact with the outside world and other open systems, are of particular interest. Structured cospans provide a formal framework for constructing monoidal categories of open systems, where the composite of two open systems results in a larger open system derived from the coupling of common ports.

Category theory has also found applications in database theory by providing a unified framework for describing and manipulating data structures. Ologs are a categorical language for describing the semantics of a database schema and can be used to reason about relationships between different parts of the schema. A database fitting a schema is represented as a functor into the category of sets, also known as a presheaf or a C-set. Lenses provide a modular and composable way of viewing and manipulating data in a database.

Sheaves offer a way of studying the relationship between local and global properties of a system and have found applications in physics and computer science. Sheaves associate local data with a space and glue together this data to form a global object. They can be used to model complex systems where the behavior of the whole system is influenced by the behavior of its individual components.

2.2. Application of Category theory to specific problems

2.2.1. Sheaves and their applications

As mentioned in the theory section, sheaves are a very important tool in category theorists' quivers. Daniel Rosiak's presentation on sheaves and their role in addressing the problem of the composition of local structures into a valid global structure as a generic method applicable to a wide range of applications. A problem that is at the heart of systems engineering problems where the goal is the composition of the structure and behavior of local components into a logically valid global systems behavior.

The workshop included two applications of this technology. Alan Hylton from NASA discussed the use of sheaves as a basis for solar-system-wide networking, helping to manage the dynamic network topologies required for communication between orbiting bodies. Second, Pascal Le-Masson presented on the use of sheaves to provide space for creativity within formal models, with an example from culinary design.

Daniel Rosiak: Sheaves and their applications

The lecture's primary focus is on the role of sheaves in the process of creating a valid composition of the local and global behaviors of a given system. A common problem in systems engineering is the composition of the structure and behavior of local components into logically valid global systems behavior. The lecture "Some Propaganda for Sheaves: how to think about them, what you can use them for, and where you can find them" offered a brief glimpse into the power of sheaf theory as a framework for modeling a variety of applied problems involving local-to-global synthesis.

The talk was structured around three "headlines" about sheaves, starting by unpacking the claim that sheaves are the general calculus for local-global inference, i.e., an information structure used to track and aggregate local data (together with their constraints, codifying some consistency or consensus relationships between the data) into global solutions.

The second headline was that sheaves are to category theory what continuous functions are to the study of functions. A function is a device that assigns a datum to each element of a set; continuous functions are then assignments that interact well with limiting behaviors, such as the convergence of a sequence of points. The notion of a sheaf can be seen as a far-reaching retelling of this story at a much higher level of abstraction: a presheaf is a sheaf if it preserves certain limits (specifically, sending colimits to limits), letting the limits in question come from open covers (which similarly "approximate" an object).

The third headline considered the utility of doing things in greater generality, with sites instead of topological spaces. There are plenty of contexts where some "topological intuition" is floating around, but where the situation cannot be described by either a literal space or even locale—yet such a situation may give rise to a Grothendieck topos (or sheaves on a site).

The talk covers the general pattern typically found in applied situations where sheaves often prove useful: (1) some broadly topological intuition is present in the domain category to which the data is functorially attached, where this underlying data assignment is something that involves "locality"; (2) it is useful to deduce observations about an entire system of entities from observations about its pieces, while just tracking information locally, and where perhaps it is not feasible to make any such global observations directly.

Finally, the talk gave a glimpse into a few of the more recent exciting applications of sheaves: for instance, in generalizing parts of the usual approach in graph neural networks and lifting the combinatorial graph Laplacian and diffusion dynamics to a greater variety of situations (e.g., applications involving opinion dynamics on social networks).

Alan Hylton: Mathematics of the solar system internet

The focus of the talk was work on the use of sheaves in modeling a potential future solar-system internet (SSI). The talk began by showing why the current model of the internet relies on assumptions of (mostly) fixed networks which does not apply to networks in space because of high latencies and high variance in latencies, disruption, mobility, density, limited group stations, and limited relays. The problems require a dynamically composable network based on the current state of the networks to interconnect them to create a path between the two vertices. In this high-variance communications context, there is a need to keep the network alive by composing a mobile collection of local networks to create a global path. The work here provides a mathematical formulation time-varying networks using sheaves, allowing for the calculation of a dynamically-varying distance between nodes. The sheaves are constructed on the edges of the local networks (e.g., lunar), and the global network is composed by dynamic stitching of these subnetworks based on changing positions and occlusions.

The talk identified future areas of applied research, including the use of hypergraphs for modeling broadcast, higher-dimensional network interactions, service dependencies among cloud services, ranking of cloud service providers, and domain-name analysis.

On a more theoretical side, future work includes the study of algebraic topology in clock synchronization, sheaves on hypergraphs and graphs as schemes, time-varying systems – moduli space of graphs, generalized sheaf pullback for non-sequential networks, time synchronization using sheaf Laplacians, consistency, sheaf cohomology and network coding as sheaves,

Pascal Le Masson: Sheaves as a design technique for creatively preserving complex systems – principles, illustrations, and first applications.

In this presentation, a preliminary sheaf-based characterization of the Concept-Knowledge design theory that describes the design as a set of moves from the concept space and knowledge space until a satisfactory solution concept can be justified by the accumulated knowledge. In the case of designs that requires the preservation of certain facets of the artifact while adding new facets to the artifact. In this case, the speaker used the example of creating a new culinary dish, as well as industrial applications such as an aircraft fuselage. The primary focus of the talk was the use of presheaves to describe existing knowledge structures, and sheaves to express joint constraints on existing knowledge generated by design requirements. Global sections of the sheaf characterize designs with the desired properties. Future work here would be to characterize a design problem and work out the detailed mathematics of a design to connect the theoretical to a detailed instance using the theory of sheaves.

2.2.2. Algebraic databases, Multi-model updates, robotic planning, and lenses

Algebraic databases provide the mathematical foundation for data migration, integration through composition across different data models, data integration, and with type consistency checking and data integrity. Lenses are powerful operators on categories that allow for extracting and updating specific views of a system model. Lenses can be seen as views/updates in traditional relational databases. They operate in any system that exhibits dynamic behavior and state changes in its behavior. Examples discussed included robotics planning, model management as updates, and BX programming language for systems

Ryan Wisnesky: Algebraic Databases

The presentation by Ryan Wisnesky was on Algebraic databases and starts with its naïve definition of the schema as a category and a functor from schema C to sets and a query from Schema C to schema D is a profunctor $C \to D$. Further, the limitations of this definition as having infinitely many arrows will require the use of finite representations of categories. The other suggestion is to equip schemas with additional structures that should be preserved to overcome the problem of infinitely many choices of functors that are constructed up to natural isomorphisms. The talk provided a review of how these problems were addressed for algebraic data integration.

From here, the talk continued to applications, where these techniques, specifically fast algorithms for left Kan extensions, were used to support consensus-free spreadsheet integration. This process converts and merges a two or more spreadsheets (often produced by different engineers) into a composite algebraic database, retaining the original spreadsheets as different views into the composed model. The resulting bridge between views can be used to shared data asynchronously different engineers, and to flag attempted updates by one engineer that violate constraints specified by another.

2.2.3. Knowledge Elicitation and Representation for knowledge bases

An important contribution of category theory is in the formalization of knowledge in the domains of interest and building formal knowledge bases to support modeling, tracking, and reasoning over databases. The first talk was on Category theory as accounting addresses the problem of elicitation of knowledge from domain experts while the second talk is on category theory to represent higher-level knowledge to construct knowledge bases.

David Spivak: Category theory as accounting

Knowledge elicitation is critical to modeling and validity. David Spivak talked through the process of knowledge elicitation with an subject-matter expert in finance to illustrate how a categorical perspective provides the ability to be precise and formal. The formality allows for the accounting of all the elements and their relationships. The talk then illustrated the use of Ologs, Operads, and string diagrams through examples in the accounting of systems and system characteristics.

Derek Wise: Higher Knowledge Representation

The goal of this project is to explore the use of category theory to represent higher levels of knowledge to encode them as higher knowledge representation structures. Traditionally the link between raw data, information, and formalized knowledge for reasoning is used in applications such as design, requirements checking, autonomous systems, information retrieval, and others. Information contained in raw data manifests itself in spreadsheets, books, and engineering drawings, but knowledge is more accessible when expressed in terms of formal structures such as ontologies and knowledge graphs. Category theory with its reflective nature allows for knowledge to be represented formally in categories but more importantly, allows for composing knowledge structures at different levels of abstraction and vertical and horizontal composition through the use of higher categories. The work presented the common use of graphs as knowledge representation and their representation as categorical structures as it captures the compositional nature of relationships included in labeled graphs. Category theory overcomes some of the limitations of graphs such as lack of compositional structure, overly flexible interpretations, and problems of surveyability for large-scale graphs.

CT allows for the representation of knowledge as category instances that are functorially mapped to a category of type. If two categories of instances are functorially mapped to the same category of types, queries can be formed to create the functorial map between the two categories of instances. The creation of the functorial maps from one category of instances

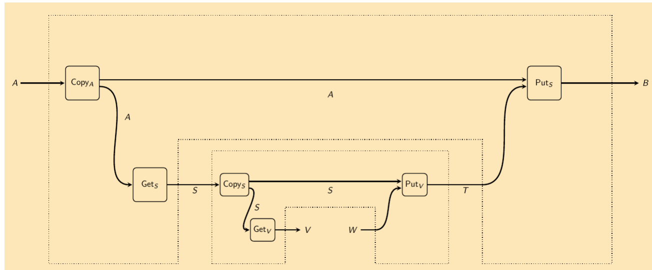

Fig. 3. Lens composition corresponds to nesting of "comb" processes (dashed lines).

to the other provides a map between the two functors (natural transformation). In the case of databases, the Grothendieck construction converts a relational model for a schema into the associated graph database, and vice versa.

One common use of types systems is to encode hierarchies between both conceptual entities (e.g., woman/person) and between relationships (e.g., daughter/child). In the world of ontologies, types hierarchies are constructed as an "is-a" relationship to be able to compute subsumption relationships between types and become useful for cross-domain interoperability. These hierarchies can be incorporated as internal categories of some sort as a map between an ordered set of objects and an ordered set of morphisms. Pre-order categories represent the orderings found in the type and relationship hierarchies.

Recent work suggests that double categories, compositional structures involving two separate axes of composition, usefully model this situation, with type and relation hierarchies as one- and two-dimensional structures, respectively. One axis in the double-categorical models only "is-a" relationships, while the other dimension captures more general relationships. These two dimensions generate a space that is "filled" by entailments between these complex relationships.

The categorical machinery presented facilitates the creation of a rigorous knowledge representation for knowledge bases thus allowing for collaboration among domain experts, ontologists, AI reasoning, system engineers, and applied category theorists.

2.2.4. Compositionality and modeling of systems

Fabrizio Genovese: Compositionality in systems: the 10x engineer secret sauce

The primary focus of the talk was how compositionality is the act of composing systems leading to the behavior of a complex system. Systems are objects that act and transform the input in a given environment. Processes act on systems that lead to their transformation. These are illustrated by examples from simple arithmetic and planetary dynamics. From this basis, the condition of composition requires that type of the input of the process should match the output of a process that is connected to it. Types of objects are very important in ensuring the composition of processes. With this basic set of primitives, the category of data and sets are defined, along with a mapping that relates a program as code (the intensional view) to its input-output behavior (the extensional view).

A more sophisticated example include the composition of programs that read and write external data, which may be handled using formal structures known as lenses, abstracting the familiar get/put operations for databases. These support a more complex style of composition using "combs" (see Fig. 3) to represent bidirectional interaction between subsystems.

This model allows for incomplete specifications involving multiple levels of abstraction, with "open slots" for unspecified components (with compatible boundaries). The same ideas can be used to study the composition of strategic games, such as the iterated Prisoner's dilemma. The tutorial illustrated using simple examples to illustrate the use of bidirectional transformation that was generalized from the database rewrite/read operations for applying them to system design and composition.

Giorgios Bakirtzis: Categories, formal methods, learning and compositional problems in autonomy

This presentation began with a review of the need for effective modeling to support early-stage design decisions in systems engineering. The 80-20 rule suggests approximately 80% of the variance in important metrics like cost and delivery time is explained by the first 20% of design decisions. Ensuring coherence and correctness of these decisions is dependent on matching the correspondence of the models to the system being designed. The problem of correspondence comes from semantic gaps arising out of static and dynamic models. Semantic convergence of the models dictates the conformance to requirements that are high-level models of systems behavior. Category theoretical diagrams as in wiring diagrams represent system components as boxes and wires as information flows.

Given a wiring diagram model for a composed system, we can use monoidal functors to describe different semantics for the model (e.g., deterministic vs. probabilistic). One important semantics for system design is the algebra of contracts, which encode constraints that subsystems are expected to satisfy. From these, one can define the Markov decision process as space of states, a space of actions and a probabilistic function that sends state-action pairs to future states and rewards.

Using these categorical models, one could address safety in the case of robot operation for obstacle avoidance as the composition of obstacles. Finally, the development software as wiring diagrams provides facilities to describe a system as set of empty boxes and interfaces by constraining the behavior of the empty boxes. Using this approach wiring diagram composition simulates behavior allowing for control response to be computed compositionally. Vertical composition of contracts from abstract machine to abstract contract machine to specific contracted machine using the static contract to derive both the discrete and continuous behaviors. By identifying failures at a local level, this allows for learning at the level of individual tasks, while composition supports the modularization of these subtasks to allow for system-wide learning.

2.3. Programming

Evan Patterson: AlgebraicJulia: a compositional approach to technical computing

The talk was on AlgebraicJulia, which focuses on developing open-source software for applied category theory specifically for technical computing. The software architecture facilitates a compositional approach to scientific and technical computing. The software platform has been developed in the Julia programming language. Over the last 10 years, there has been a continual expansion of the application of category theoretic concepts, including methods for symmetric monoidal categories (SMC) and wiring diagrams, open systems via co-spans, functorial semantics in science and categorical data migration. The goal of the AlgebraicJulia project is to provide computational support to implement CT abstractions as highly general software and to instantiate these abstractions in specific scientific and engineering domains. The underlying objective is to provide for compositionality and interoperability between domains and applications, functorially.

Catlab.jl is at the core of AlgebraicJulia, supporting modeling applications in epidemiology (Petri nets, stock flow diagrams), dynamical systems, and computational physics (simplicial sets, discrete exterior calculus). Computational category theory is realized in several ways: in the design of programs and programming languages, in proving theorems about program behavior and correctness, and in the specification of complex models of systems outside computing. The focus of Catlab is primarily on computational modeling in last sense, focused on the scientific and technical domains.

Adopting a pragmatic approach, AlgebraicJulia provides the categorical mathematics for syntax while providing the syntax functorially. In AlgebraicJulia, Domain Specific Language gives a “front end” for syntax, via Julia macros with Semagrams as the graphical editor with combinatorial data. For example, in AlgebraicJulia undirected wiring diagrams provide the syntax for composing spans of data tables in a conjunctive query, while structure co-spans allow for the creation of open graphs, open Petri nets, and open free diagrams. Open dynamical systems, which compose via input/output flows, shared resource constraints. All of these algebras are implemented in AlgebraicJulia, with benefits arising from the shared abstraction. Data structures are provided by a certain class of profunctors called attributed C-sets, for a given data schema C. These support a powerful set of data transforms based on iterated limits (joins), colimits (disjoint unions) and graph rewriting (rules).

The presentation illustrated structured scientific modeling by providing the ability to compose simple models by making explicit the compositional structures in the domain of modeling that are often lost in the design and implementation domain models in current practice. The power of this modeling was illustrated with examples from the epidemiological model of COVID-19 as a case study. In this case, examples of multi-scale models in computational physics using a rigorous model of the Tonti diagram of Maxwell’s equation. This is further extended by providing tools for a pipeline of differential equations through wiring diagrams, AC-sets as schemas, and directed wiring diagrams for computational graphs.

2.3.1. Lenses for model management

Lenses are a very useful category in modeling the dynamics of systems and the update of the state of a system. The set of talks below is on the application of delta Lenses in robotic path planning, system models, and as a programming approach.

Angeline Aguinaldo: Contextual affordances in context-aware autonomous systems

In this talk, context is defined as affordances provided for robot actions related to a given robot with the objects in the environment. A motivating example was provided with actions that can be taken by the robot and an initial scene graph of the objects in the environment to create an affordance task plan. In the next step, a query is initiated with respect to a change in the scenario for computing the new affordance task plan as Delta of the previous plan. Using this example as the basis the talk described the use of symmetric data lenses to create affordance relations.

The first step in this process is to define a language of scene graphs represented as copresheaves (C-sets) with C as a schema category and target sets as specific instances of each class. Arrows are natural transformations. Similarly, a planning domain language is described as a set of action schemas with parameters with preconditions and effects. Affordance relations are computed using functors from the planning domain to the scene and from the scene, the ability to complete the planning tasks is computed. Given this structure, change propagation is computed using symmetric data lenses to answer the following queries: a) What scene graph is afforded by the planning domain? Given a change in the scene what changes are offered by the planning domain, 3) what planning domain is afforded by the scene graph, and 4) Given a change in the afforded planning domains, what changes are necessary for the scene graph?

Future work in this area is to provide for compositional affordances and hierarchical affordances to address the lack of formalization of affordances requiring the composition of objects and actions.

Zinovy Diskin: Multiple Model Synchronization with Multiary Delta Lenses (Mx)

The motivation for this work comes from applications in model-driven engineering and the case of safety engineering for an automaker. The primary problem in the practice of model-driven engineering is the relationship between different models that arise from software design using UML and hybrid models of hardware and software that require a mapping between the software models and hardware models while maintaining the consistency of the models. The primary goal of the talk is to illustrate model synchronization using categories and delta lenses along with algebraic laws for lens composition. In other words, why use delta lenses better than state-based lenses to address view update problems and address concurrent updates?

The talk makes the case for Delta lenses instead of state-based lenses for change propagation. The problems with state-based deltas are in the efficiency of computation and maintaining semantic consistency. Given the three dimensions of model space as views, refinements, and change in time, there is a necessity for MX change over binary delta change propagation. The compositionality in the MX change is developed to a formal well-behaved lens where PPG operations are unital and compositional. Subsequently, several themes for MX-BX(binary updates) were explored including MX to Bx- spans and back. More general compositions include complex synchronization of multiple systems and (Co) star composition of Lenses. Change propagation with amendments and Concurrent updates. The mathematical formulation through delta lenses for multi-model software/hardware systems will aid in the software implementation of this work for change management across models.

2.4. Ecosystem for the propagation of Category theory and applications

Since the first workshop on applied category theory at NIST in 2017 with about 20 attendees, the community has grown because of multiple events that have taken place over the years.

A community of Applied Category theorists has successfully launched a series of workshops (ACT 2018-2023) as a bridge between mathematical disciplines and more applications outside mathematics. Consistently the community has grown into 100's of participants over the years.

Topos Institute for applied category theory was started by David Spivak and Brendan Fong in Berkley, California, in 2010. This institute is committed to promoting category theory and open-access tools for applications.

In light of these developments, the second NIST workshop was targeted at "Category theory for Systems engineering." as the theme of the workshop. There were about 50 participants from Industry, government, and academia. There were many interactions between different groups that have led to possible collaboration possibilities. There was also an attempt to scope the possibility of a consortium for future research and development in the focused meeting of groups. Three major themes emerged out of these discussions:

1. To create mechanisms for developing a list of common problems across industries to create joint projects that can contribute to the case bases and tools for applications. Also
2. To create a funding base through which industries, academics, and government can participate to enhance the field of applications and tools while creating a collective open computational infrastructure.
3. Finally, an important avenue for category theory could be developed it into language for the interoperability of tools and databases encoding both the formal syntax and semantics of exchange. This effort could dramatically but requires effort to create a standards development mechanism. Overall, the need for coordinated efforts and propagation of CT and its applications and education was recognized and also seen

as an impediment to promoting CT. The tasks of any efforts to create a consortium will have to address education, Industry applications and tool development as their cornerstones.

Disclaimer: Certain commercial equipment, instruments, or materials, commercial or non-commercial, may be identified in this paper in order to adequately describe the event. Such identification does not imply recommendation or endorsement of any product or service by NIST, nor does it imply that the materials or equipment identified are necessarily the best available for the purpose. The opinions, recommendations, findings, and conclusions in this publication do not necessarily reflect the views or policies of NIST or the United States Government.

Appendix A. Workshop Schedule

Nov 3, Thursday			
Start	End	Duration	Speaker
9:00	9:10	0:10	NIST Introduction
9:10	10:10	1:00	*Compositionality and categorical thinking* Fabrizio Genovese
10:10	10:30	0:20	*The role of industry in progressing category theory* Esteban Montero
10:30	10:45	0:15	Break
10:45	11:30	0:45	*Multiple model synchronization with multiary delta lenses* Zinovy Diskin
11:30	11:50	0:20	*Sheafification as a critical design technique for creative preservation in complex systems – principles, illustrations and first applications* Pascal Le Masson
11:50	12:10	0:20	*A contextual affordance model for autonomous systems* Angeline Aguinaldo
12:10	12:30	0:20	*Category Theory as a possible theoretical foundation for Systems Engineering* David Perner
12:30	13:30	1:00	Lunch
13:30	14:30	1:00	*What are we tracking? How applied category theory puts thinking on rails* David Spivak
14:30	15:00	0:30	*Algebraic Databases - Lessons Learned* Ryan Wisnesky
15:00	15:15	0:15	Break
15:15	16:00	0:45	*Categories, formal methods, learning and compositional problems in autonomy* Georgios Bakirtzis
16:00	17:00	1:00	Discussion Session

Nov 4, Friday			
Start	End	Duration	Speaker
9:00	9:45	0:45	*Emergent effects and contextual behaviours in categorical systems theory* Jules Hedges
9:45	10:15	0:30	*AlgebraicJulia: a compositional approach to technical computing* Evan Patterson
10:15	10:35	0:20	*Compassionate Mathematics* Priyaa Srinivasan, Pawel Sobocinski
10:35	11:00	0:25	Break (Group photo)
11:00	11:45	0:45	*Electrical circuits with string diagrams* Pawel Sobocinski
11:45	12:30	0:45	*Higher Knowledge Representation* Derek Wise
12:30	13:30	1:00	Lunch
13:30	14:15	0:45	*Mathematics of the solar system internet* Alan Hylton
14:15	14:35	0:20	*Azure Knowledge Marketplace: Categories for Scale* Donald Thompson
14:35	14:55	0:20	*Monoidal fibrations with applications* Joe Moeller
14:55	15:15	0:20	*Sheaves: Where to Find them and How to Use them* Daniel Rosiak
15:15	15:30	0:15	Break
15:30	16:30	1:00	Discussion Session
16:30	17:00	0:30	Presentation

Appendix B. List of Symbols, Abbreviations, and Acronyms

AFRL Air Force Research Laboratory.

AFOSR Air Force Office of Scientific Research.

CMU Carnegie Mellon University.

CWRU Case Western Reserve University

NASA National Aeronautics and Space Administratio

NIST National Institute of Standards and Technology

NSF National Science Foundation

OSU Ohio State University

SUNY State University of New York

TalTech Tallinn University of Technology

UAH University of Alabama in Huntsville

UMD University of Maryland

UT University of Texas