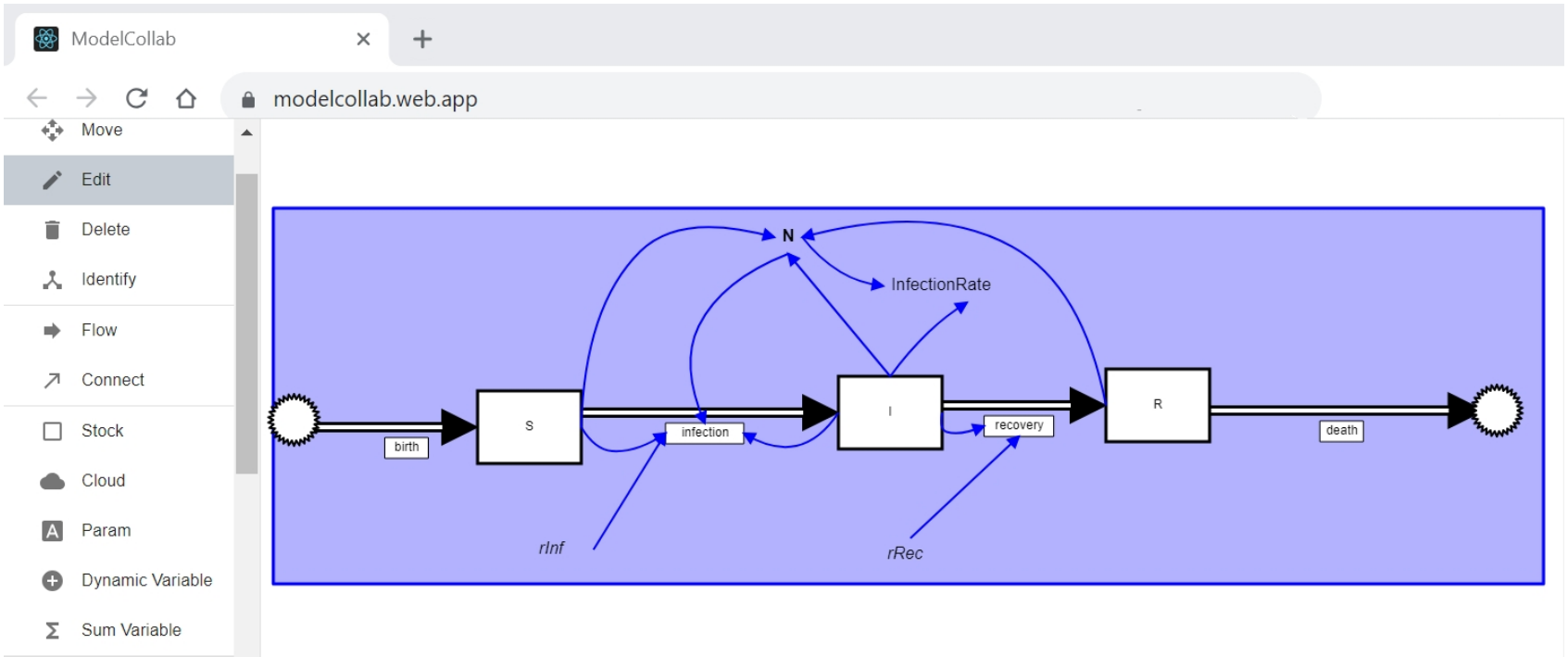
SOFTWARE FOR COMPOSITIONAL MODELING IN EPIDEMIOLOGY



John Baez

ACT 2023

2023 August 2

What should applied category theorists do?

In my opinion:

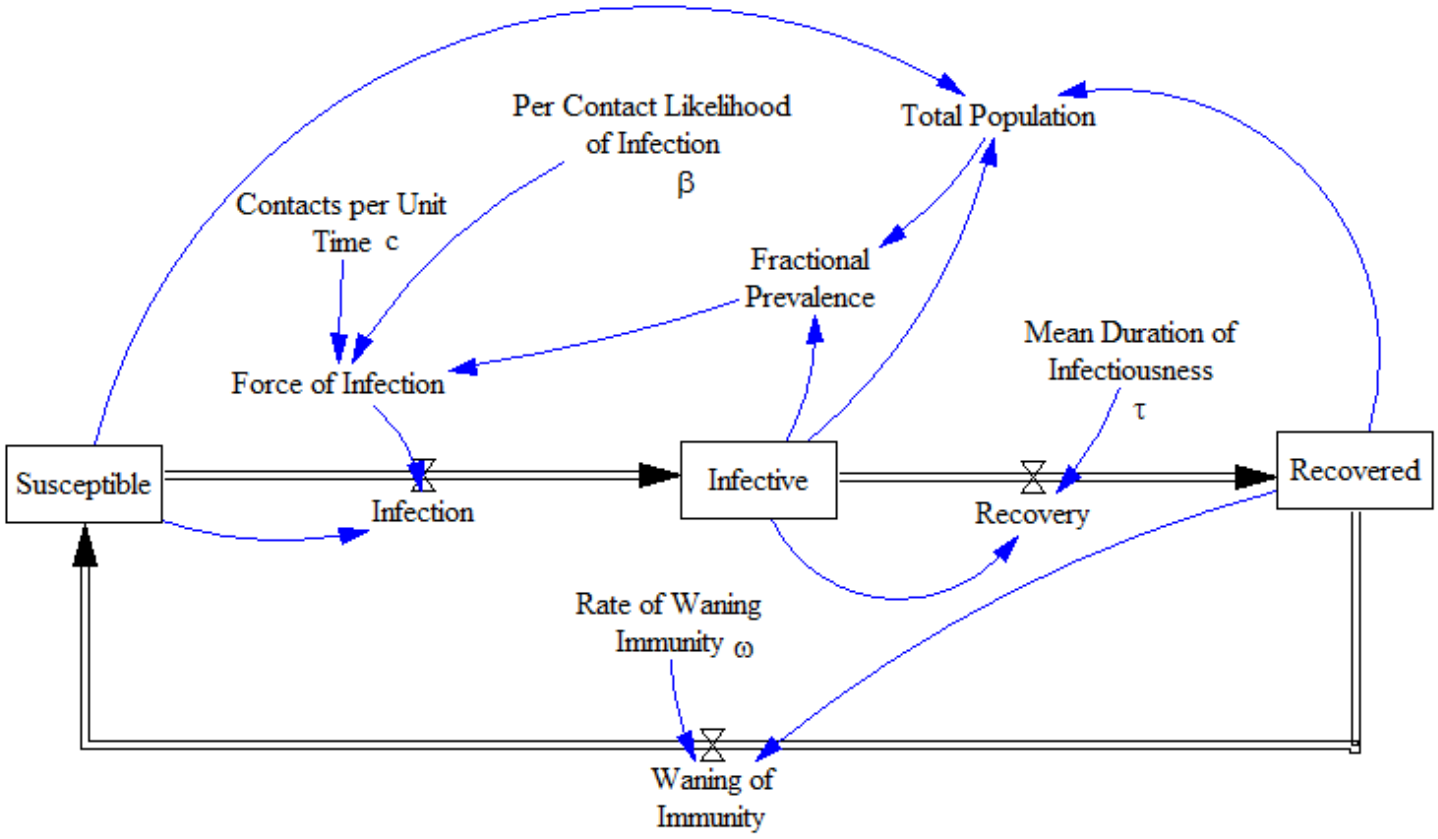
- We should develop beautiful mathematics,
- use it to solve practical problems, and
- help the world.

The first part is not very hard after you know category theory.

The second part seems to require an interdisciplinary team.

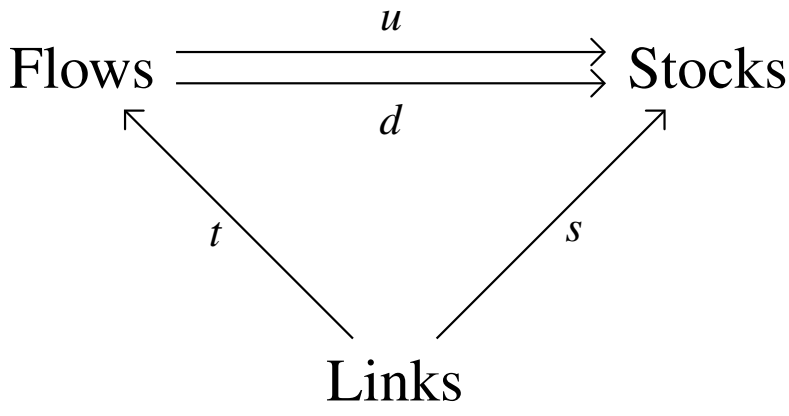
The hardest part is the third part.

In "System Dynamics", dynamical systems are modeled using "stock-flow diagrams":

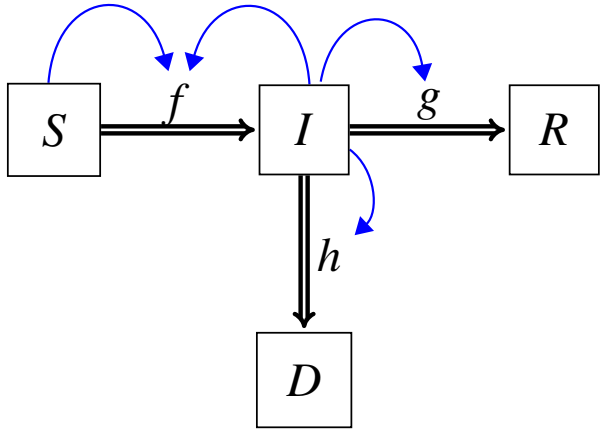


These diagrams are now widely used in economics, population biology, epidemiology, etc.

A **stock-flow diagram** consists of finite sets and functions like this:



together with, for each $f \in \text{Flows}$, a **flow function** $\phi_f : \mathbb{R}^{L(f)} \to \mathbb{R}$ where $L(f)$ is the set of links whose target is f.

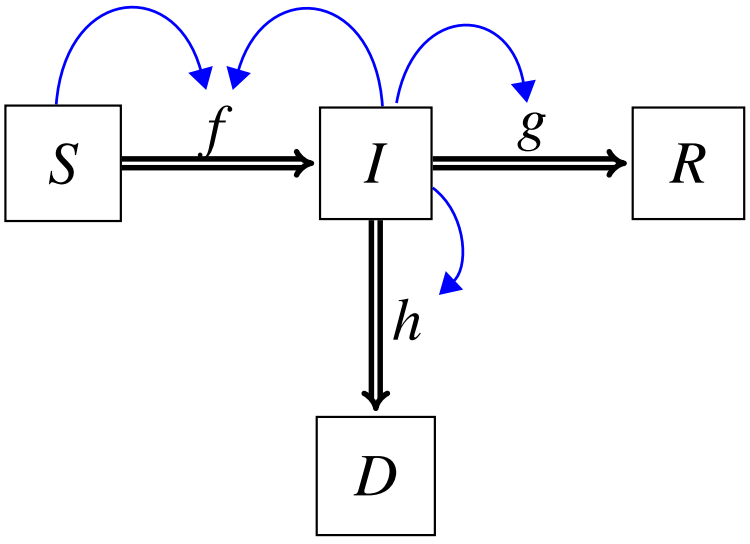


$$\phi_f : \mathbb{R}^{\{S,I\}} \to \mathbb{R}$$

$$\phi_g : \mathbb{R}^{\{I\}} \to \mathbb{R}$$

$$\phi_h : \mathbb{R}^{\{I\}} \to \mathbb{R}$$

Each stock-flow diagram gives a system of ordinary differential equations with one variable for each stock, saying how the stocks change with time:



ϕ_f, ϕ_g, ϕ_h

$$\frac{dS}{dt} = -\phi_f(S, I)$$

$$\frac{dI}{dt} = \phi_f(S, I) - \phi_g(I) - \phi_h(I)$$

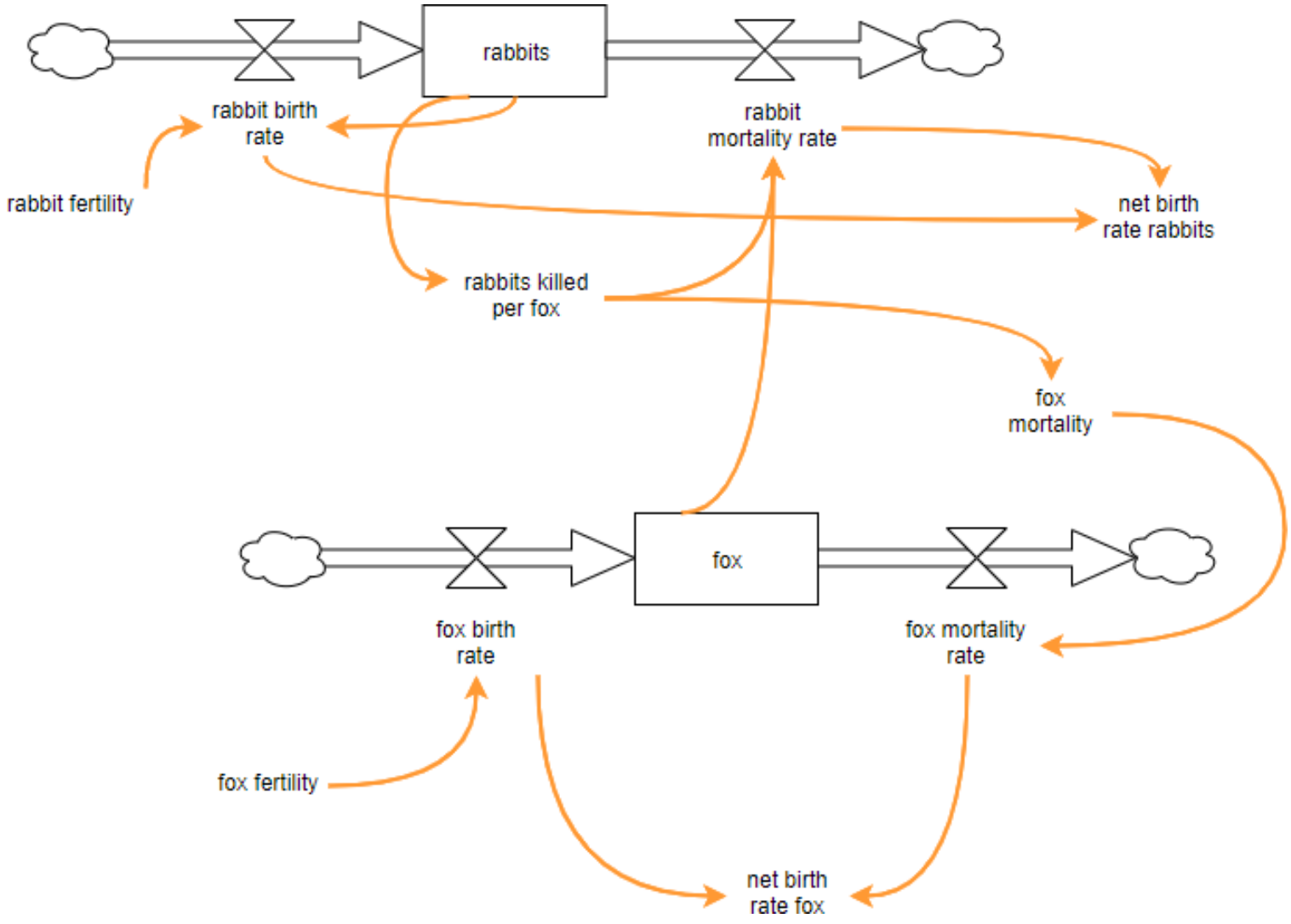
$$\frac{dR}{dt} = \phi_g(I)$$

$$\frac{dD}{dt} = \phi_h(I)$$

There is a category StockFlow, a category of systems of differential equations Dynam, and a functor

$$V\colon \mathsf{StockFlow} \to \mathsf{Dynam}$$

Modelers like stock-flow diagrams with extra features:



Our software handles these, but I won't say much about them!

Instead of stock-flow diagrams, why not just use differential equations?

- Diagrams make it easier to *compose* models.
- We can do more with stock-flow diagrams than just convert them into differential equations. If we *separate syntax from semantics*, we gain flexibility.
- For most people, diagrams are *easier to understand* than differential equations!

  In "community based modeling", diagrams let community members work with experts to help build models.

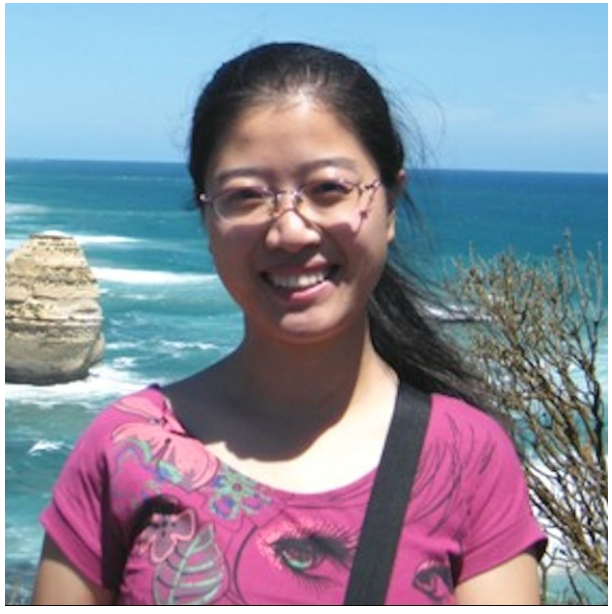
Sociologists would say the power of stock-flow diagrams — and other diagrams — is that they're "boundary objects":

> *A **boundary object** is any object that is part of multiple social worlds and facilitates communication between them; it has a different identity in each social world that it inhabits.*

Our goal is to make diagrams even more powerful by:

- formalizing them using category theory
- creating software to work with them.

Some epidemiologists use stock-flow diagrams to model the spread of disease. This includes my collaborators Nathaniel Osgood and Xiaoyan Li, who are computer scientists specializing in public health.

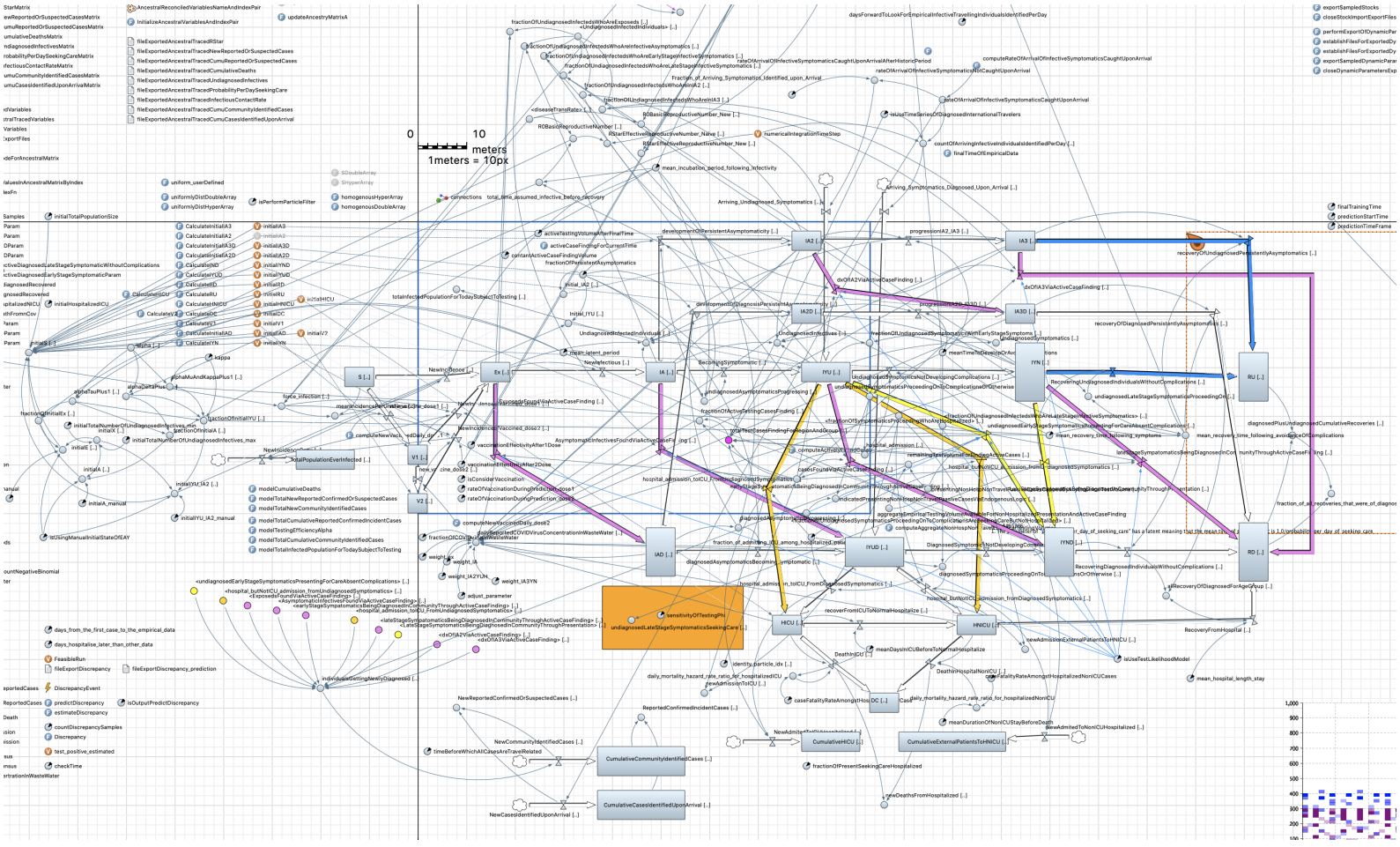


Most stock-flow modeling is done using software called AnyLogic. It's powerful, but it has some big problems:

- It has no support for *composing* models: taking several smaller models and putting them together to form a larger model.
- It has no support for *separating syntax from semantics*: interpreting the same model in multiple ways.
- It has no support for *stratifying* models: taking a model and splitting one stock into several stocks (e.g. age groups).
- It has no support for *collaboratively* building models.
- It is not *free* and not *open-source!*

Our work aims to fix all these problems.

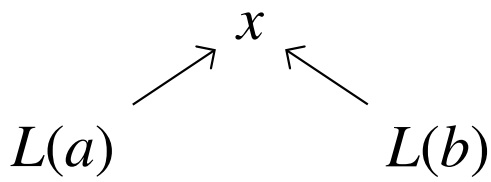
The ability to *compose* models is crucial because realistic models are complicated and built out of many smaller parts. Here is Osgood and Li's COVID model used by the government of Canada:



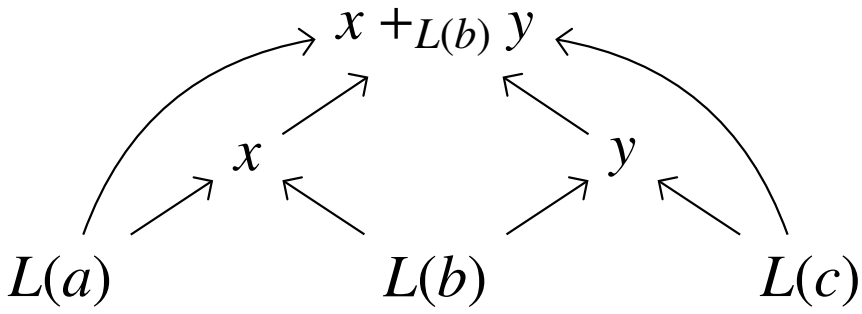
To compose stock-flow models we can use structured cospans. Given any functor

$$L \colon \mathsf{A} \to \mathsf{X}$$

a **structured cospan** is a diagram in X of this form:

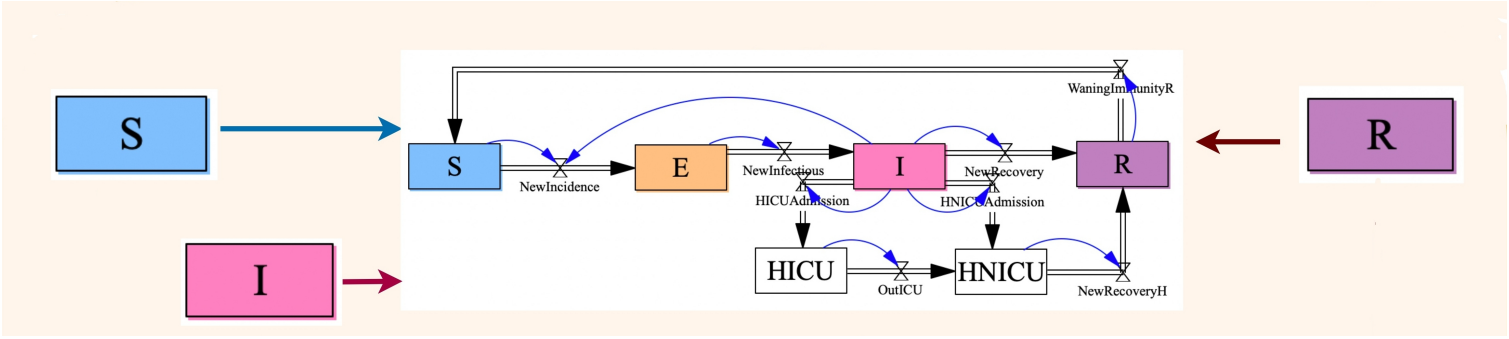


If X has pushouts, we can compose structured cospans by taking pushouts:



We get a (double) category with objects of A as objects and structured cospans as morphisms.

An open stock-flow diagram looks like this:

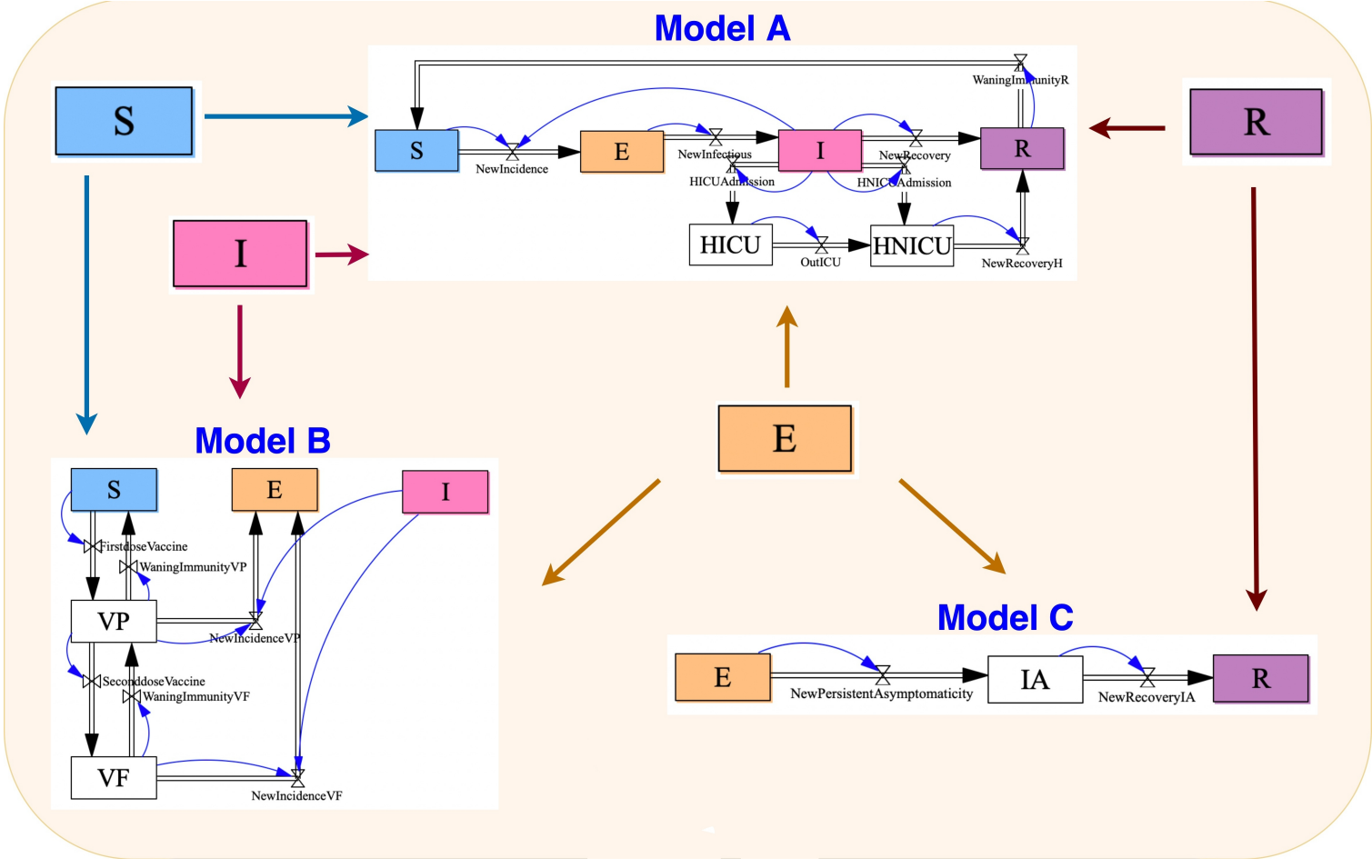


It's a structured cospan built using the functor $L\colon \mathsf{A} \to \mathsf{X}$ with

- $\mathsf{A} = \mathsf{FinSet}$,
- $\mathsf{X} = \mathsf{StockFlow}$,
- L sends any finite set to the stock-flow diagram with that set of stocks, and no links or flows.

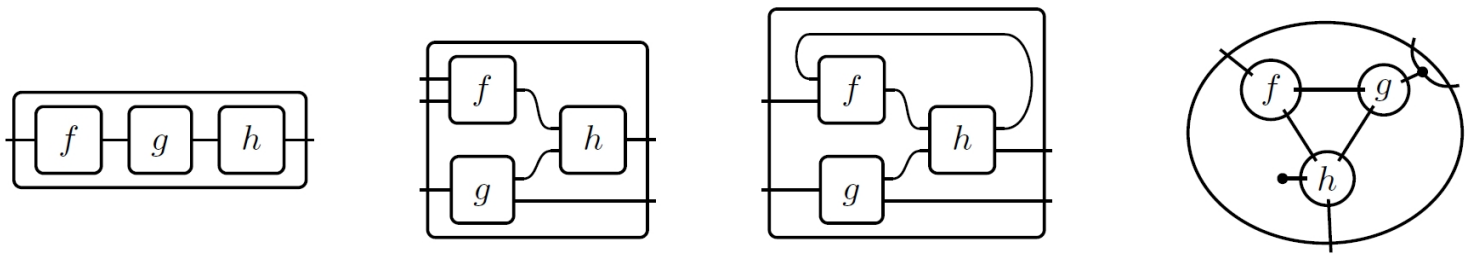
So, open stock-flow diagrams are morphisms in a category, say $\mathsf{Open}(\mathsf{StockFlow})$.

However, besides composing structured cospans end-to-end, we can also compose them in more complex ways, like this:



The reason is that if A and X are categories with finite colimits and $L\colon \mathsf{A} \to \mathsf{X}$ is a left adjoint, the corresponding structured cospan category is a "hypergraph category".

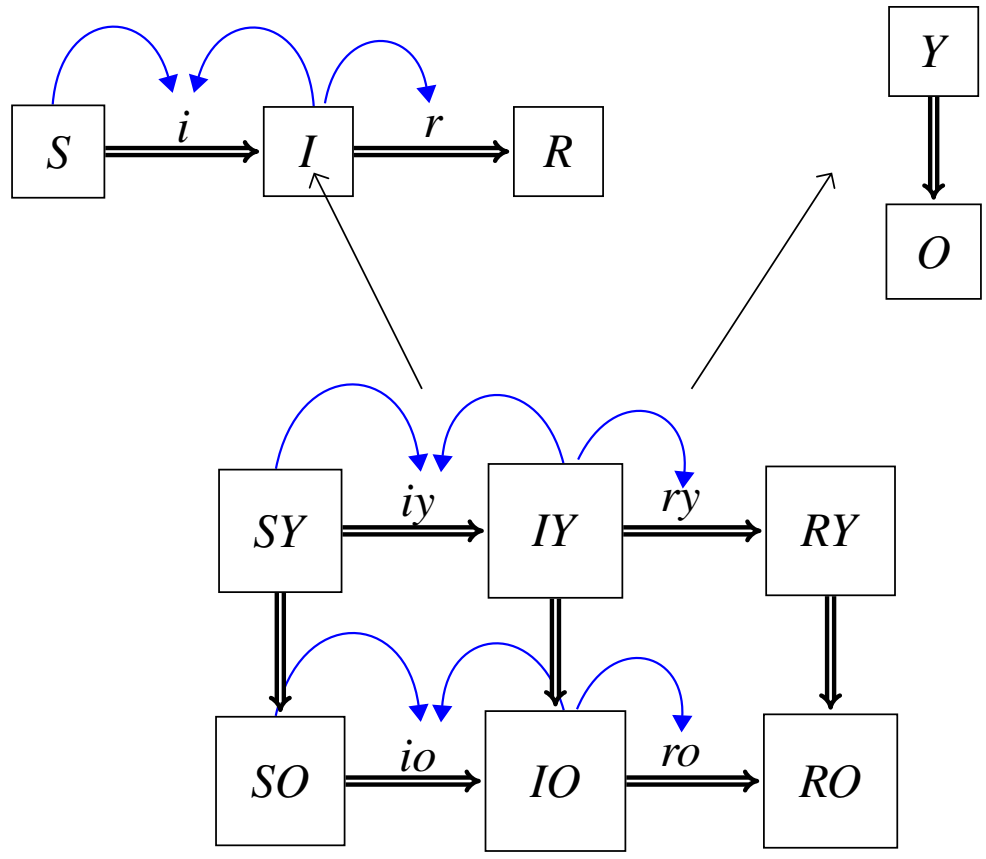
In a hypergraph category, we can compose operations in patterns described by the "operad of undirected wiring diagrams":



category | monoidal category | traced monoidal category | hypergraph category

- Brendan Fong and David Spivak, Hypergraph categories.
- JB and Kenny Courser, Structured cospans, Thm. 3.12.

To “stratify” stock-flow models — refine them by breaking a single stock into several stocks — we must use pullbacks in the category StockFlow, e.g. products:



Together with Evan Patterson and Sophie Libkind we created software for working with stock-flow diagrams.


- John Baez, Xiaoyan Li, Sophie Libkind, Nathaniel D. Osgood and Evan Patterson, Compositional modeling with stock and flow diagrams.

We used AlgebraicJulia: a framework for high-performance scientific computing using categories. This was developed by James Fairbanks, Evan, Sophie, and many others.

Composing structured cospans using undirected wiring diagrams had already been implemented in AlgebraicJulia:

- Sophie Libkind, Andrew Baas, Evan Patterson and James Fairbanks, Operadic modeling of dynamical systems: mathematics and computation.

Using AlgebraicJulia we created a software package called **StockFlow**, now available on GitHub. This lets you:

- build stock-flow diagrams
- draw them
- make them "open"
- compose them using the operad of undirected wiring diagrams
- apply various functors to interpret stock-flow diagrams, e.g. as systems of differential equations but also other things
- numerically solve the resulting differential equations
- stratify stock-flow diagrams using pullbacks

What are the "various functors" that StockFlow can use to interpret stock-flow diagrams?

Besides

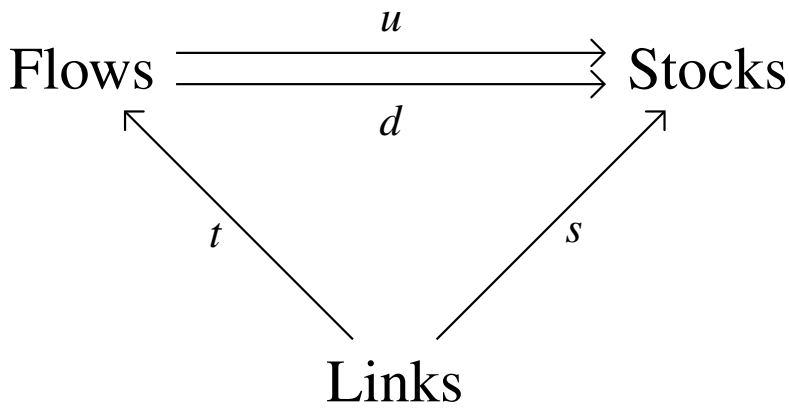
$$V\colon \mathsf{StockFlow} \to \mathsf{Dynam}$$

another is

$$A\colon \mathsf{StockFlow} \to \mathsf{SystemStructure}$$

This converts stock-flow diagrams to another style of diagram.

A **system structure diagram** is a diagram of this shape in the category of finite sets:

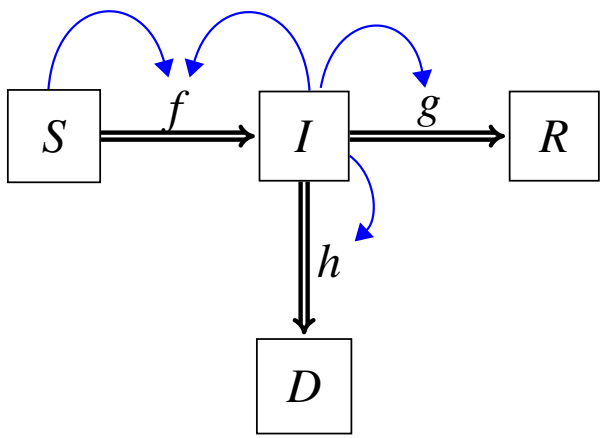


So, SystemStructure = $\mathsf{FinSet}^{\mathsf{D}}$ for this diagram D.

A stock-flow diagram is a system structure diagram equipped with flow functions. The functor

$$A : \mathsf{StockFlow} \to \mathsf{SystemStructure}$$

simply forgets these flow functions:



$$\phi_f : \mathbb{R}^{\{S,I\}} \to \mathbb{R}$$

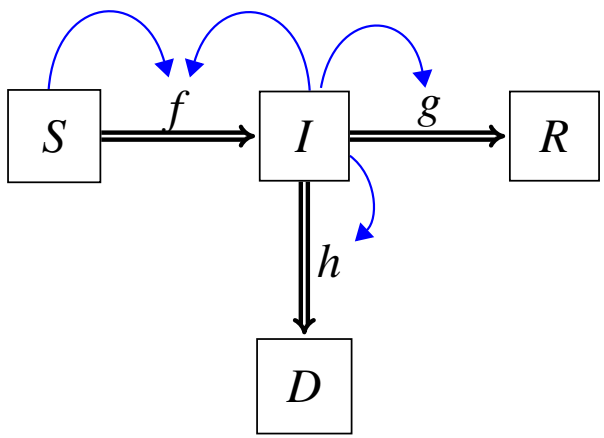
$$\phi_g : \mathbb{R}^{\{I\}} \to \mathbb{R}$$

$$\phi_h : \mathbb{R}^{\{I\}} \to \mathbb{R}$$

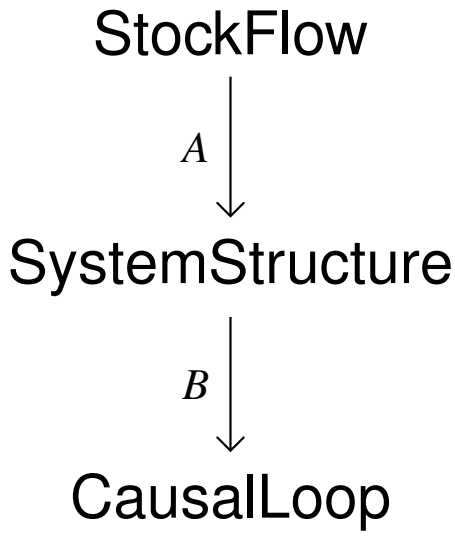
A stock-flow diagram is a system structure diagram equipped with flow functions. The functor

$$A : \mathsf{StockFlow} \to \mathsf{SystemStructure}$$

simply forgets these flow functions:



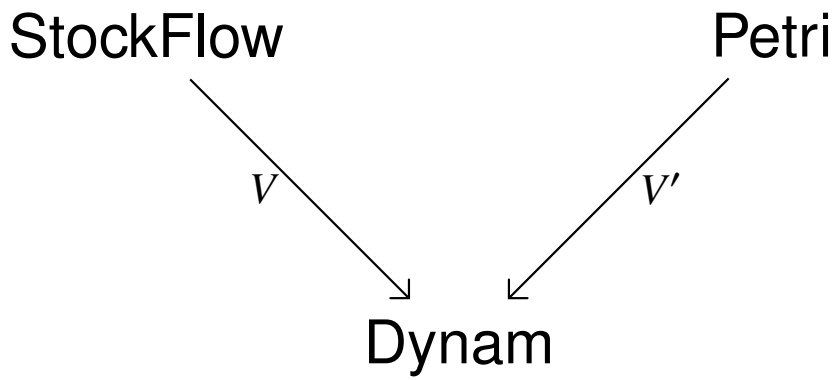
System Dynamics also uses a third style of diagram, “causal loop diagrams". We have implemented a hierarchy of forgetful functors in our software:



People design systems “from the bottom up”, starting simple and adding detail to go *up* the hierarchy!

Forgetful functors let you check that you’re on the right track — or throw out some details and start again.

Functorial semantics also lets you build big models out of smaller pieces described using several different choices of syntax — as long as they all map to some common semantics (or syntax):



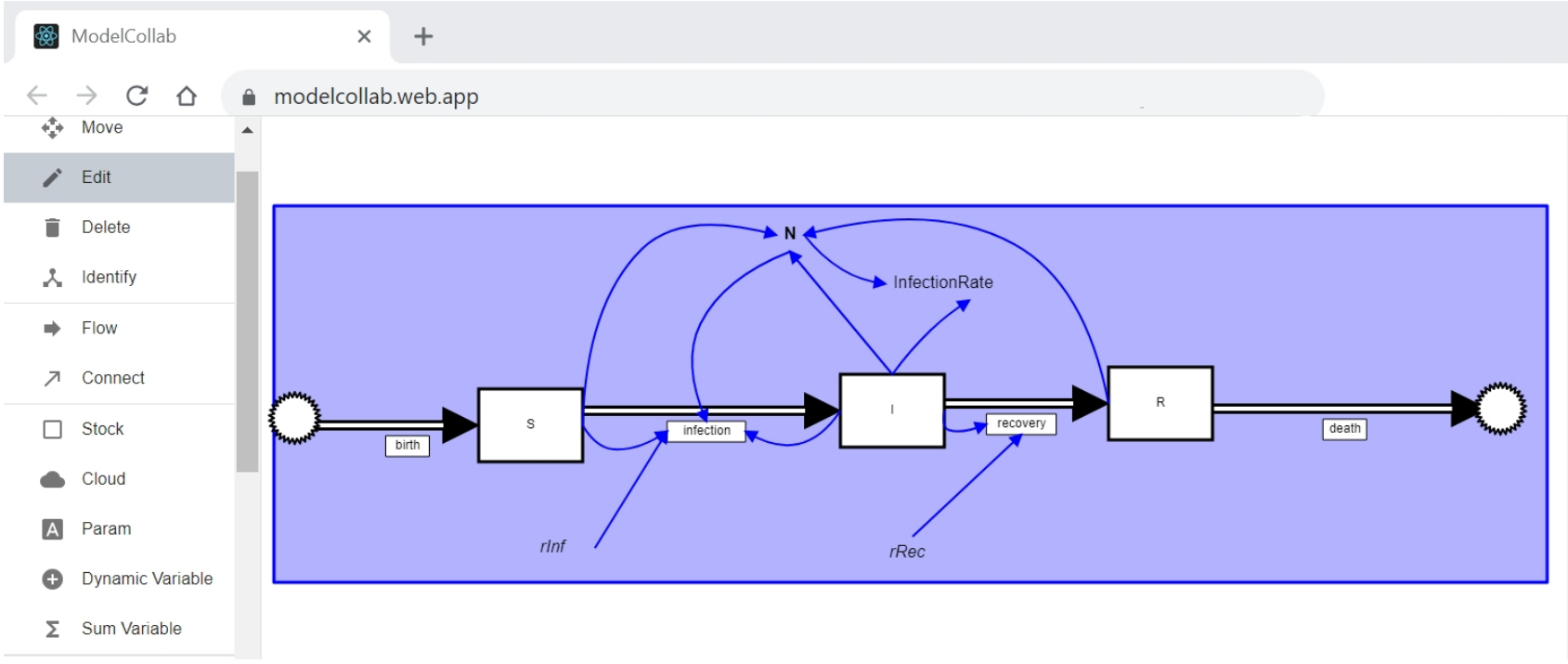
Indeed, V' has also been implemented in AlgebraicJulia:

- Andrew Baas, James Fairbanks, Micah Halter, Sophie Libkind and Evan Patterson, An algebraic framework for structured epidemic modeling.

With Eric Redekopp we made a graphical user interface for StockFlow, called **ModelCollab** — also available on GitHub.

- John C. Baez, Xiaoyan Li, Sophie Libkind, Nathaniel D. Osgood, Long Pham and Eric Redekopp, A categorical framework for modeling with stock and flow diagrams.

ModelCollab runs in your browser, so teams can collaborate to build stock-flow diagrams.



And the great thing about ModelCollab is that you *don't need to know anything about category theory or AlgebraicJulia to use it!*

The fancy stuff is black-boxed.

Hypothesis: to become widely successful, applied category theory should make itself invisible.

Three basic principles:

1. **Compositionality**: stock-flow diagrams are morphisms in the hypergraph category $\mathsf{Open}(\mathsf{StockFlow})$, so we can compose them using undirected wiring diagrams.
2. **Functorial semantics**: to interpret stock-flow diagrams we need to choose a hypergraph functor $F\colon \mathsf{Open}(\mathsf{StockFlow}) \to \mathsf{C}$, and there is more than one interesting choice.
3. **Stratification via pullbacks**: we can build more complicated stock-flow diagrams from simpler ones by taking pullbacks in $\mathsf{StockFlow}$.

StockFlow implements all three; ModelCollab just 1½ so far. But we're not done.

There is a lot of room for growth here!