Axiomatic Category Theory For Knowledge Representation

Henson Graves
September 2015

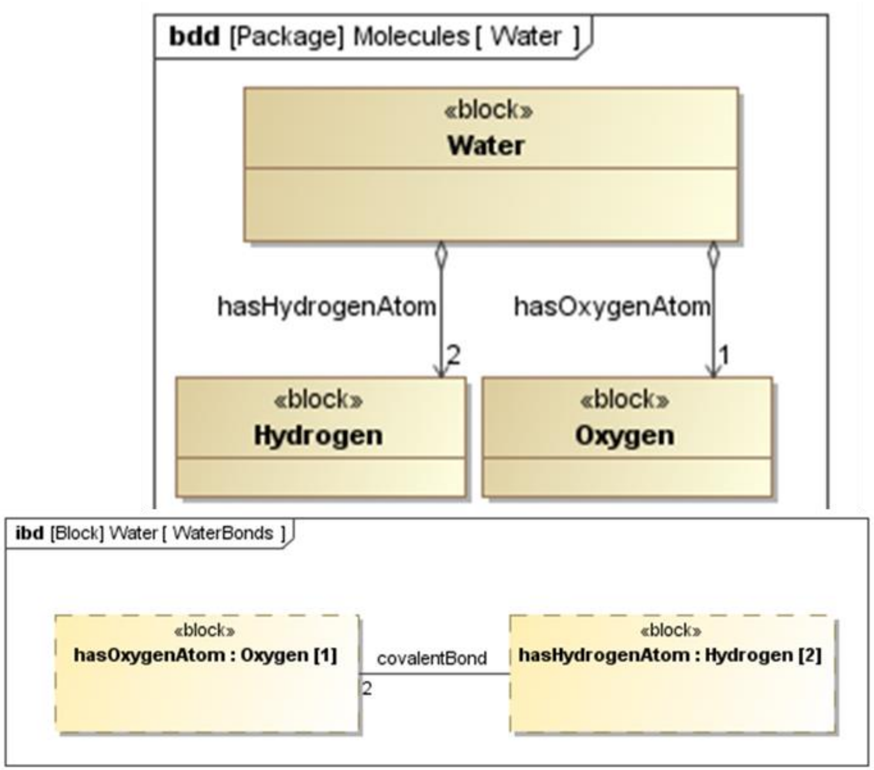
- **Engineers already use category theory, they just don’t learn it from mathematicians**
- **Engineering tools can be retrofitted with axiomatic category theory to solve currently unmanageable development problems**
- **Outline some techniques to convert axiomatic categories for use with formal reasoning – specifically for a constructive topos theory**

Note: Engineers and logicians use the term “model” differently. An engineer’s model is an axiom set, a logician’s model is an interpretation.

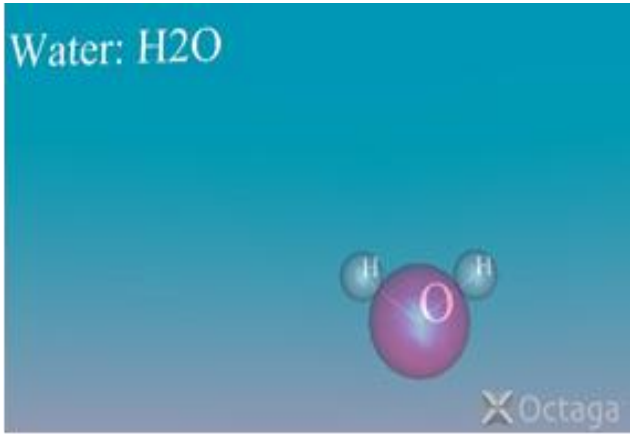
Engineers already use category theory (they just don't learn it from mathematicians)

...they use it to build descriptive models for design and analysis

Engineering Model



Realization
(Simulation)



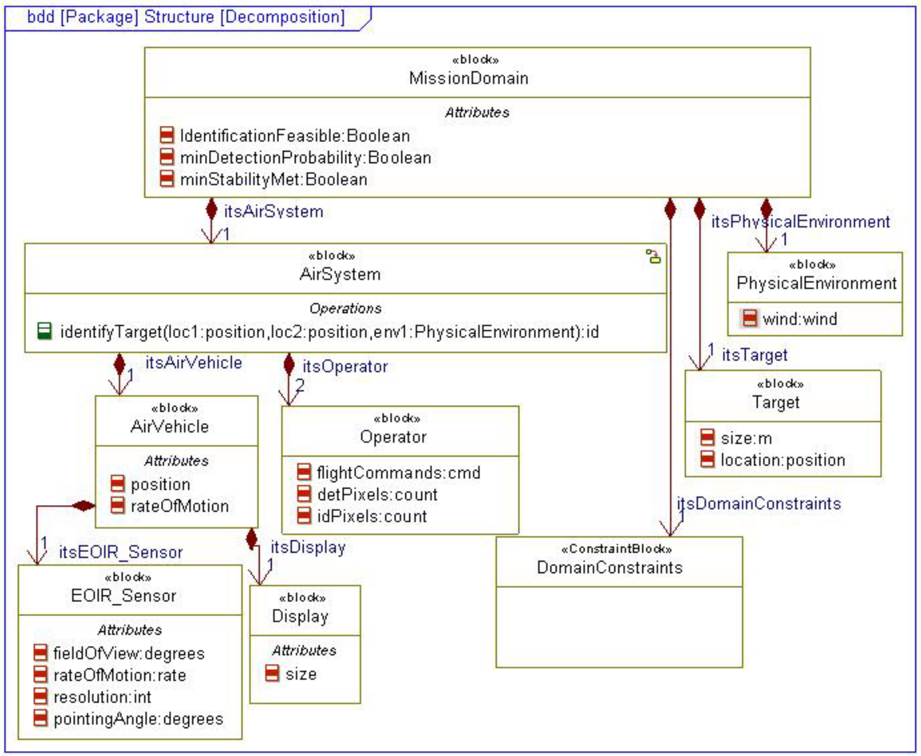
- Model was developed in a SysML authoring tool
- Simulation was generated from model

Graves, Henson "Structural Modeling in Biomedical and Product Engineering "NCOSE International Symposium, 2013

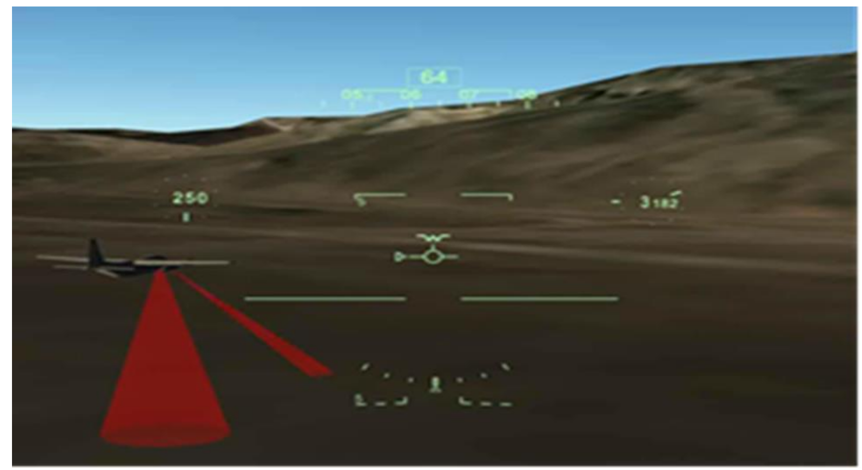
Engineers Build Models & Use Them For Design And Analysis

...this includes biomedical products as well as cyber physical ones

Engineering Model (axiom set) of Aircraft & Operating Environment



Operating Environment Simulation



- Diagram includes physics models and code to produce simulations
- Simulation is an interpretation of model axiom set

Graves, Henson, and Yvonne Bijan. "Using formal methods with SysML in aerospace design and engineering." *Annals of Mathematics and Artificial Intelligence* 63.1 (2011): 53-102

Engineering is Concerned To Build Models & Derive Conclusions From Them

... yet the engineering tools & methodology are failing

- **These models are too large and complex for manual analysis**
  - miss checking that power requirements cause battery overheating
  - integration of component models with conflicting assumptions
  - simulations are not consistent with the model to be simulated
- **They don't contain or link to information needed for analysis**
- **Automated reasoning can catch many of these problems**
  - check design consistency
  - Specify validity for simulations
  - enable semantic integration of models
- **Interactive proof construction can be used to verify design consequences**
  - E.g., verify that design modifications can avoid obstacles

- ***Engineering languages have good primitives, graphical syntax,***
- ***Require a suitable logic to embed engineering models***
- ***even with a suitable embedding, we are not home free***

Axiomatic Method Potentially Solves Many Engineering and Science Problems

... There are significant impediments to application of axiomatic method

- **Basis for knowledge sharing and reuse**
- **Enables the divorce of syntax from meaning**
- **Criteria for logically correct reasoning**
- **Application questions translate to logical questions**

- **Finding a sufficiently expressive language**
- **Embedding language within a logic**
- **Ensuring physical correctness of results**
- **Difficulty writing axiom sets**

how can the axiomatic method be made to work, and scale for science and engineering?

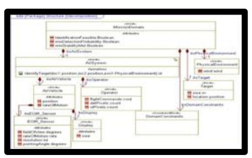
Reasoning is Correct When Results Are True In All Interpretations

... The logic paradigm is also the same in science and engineering

Meta Logic

Meta logic representation of language

Theory
generated by axiom set



The theory of an axiom set is the set of logical consequences of the axioms

An Interpretation specifies how the theory is to be corresponded with the world

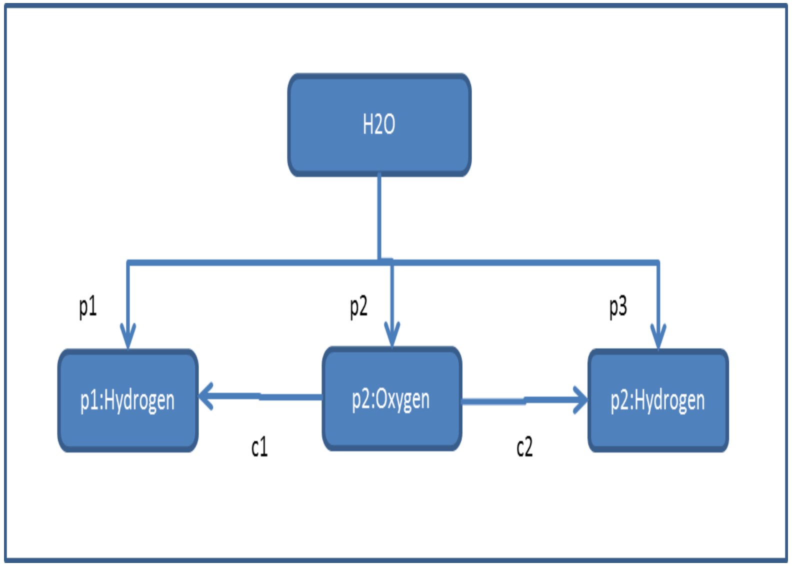
Interpretation that satisfies the axioms

...this is representation theory for mathematicians and model theory for logicians

The H2O Diagram is an Axiom Set for an Axiomatic Category

...the axiom set generates a topos

graphical syntax



Linear syntax

TypeSymbols

H2O, Hydrogen, Oxygen

PartMaps

p1:H2O → Im(p1)

p2:H2O → Im(p2)

p3:H2O → Im(p3)

PartTypes

H2O, Im(p1), Im(p2), Im(p3)

ConnectionMaps

c1: Im(p2) → Im(p1)

c2: Im(p2) → Im(p3)

Equations

c1(p2)= p1

c2(p2) = p3

...characterization of H2O requires a lot more axioms, but they can be generated from the diagram

Axiomatic Category Theory Can Be Made Computer Friendly For Reasoning

... but it takes work

Category Axioms

$f{:}Y \rightarrow X \equiv \mathrm{Domain}(f) = X, \mathrm{Range}(f) = Y$

$id_X : X \rightarrow X$

$f{:}X \rightarrow Y,\ g{:}Y \rightarrow Z \Rightarrow g(f){:}X \rightarrow Z$

$h{:}W \rightarrow X,\ f{:}X \rightarrow Y,\ g{:}Y \rightarrow Z \Rightarrow$

$(g(f))(h) = g(f(h))$

$id_X (f) = f$

Product Axioms

One – type constant, called the terminal

$f{:}Z \rightarrow Y,\ g{:}Y \rightarrow Z \Rightarrow <f,g>{:}Z \rightarrow (Y,X)$

$!_{X:}X \rightarrow$ One

$!(f) = !$

$pi1_{X,Y}{:}(X,Y) \rightarrow X,\ pi2_{X,Y}{:}(X,Y) \rightarrow Y$

$f(<pi1,pi2>) = f$

$h(<f,g>) = <h(f),h(g)>$

...

Questions and Answers

What kind of logic

first order language with two sorts, Map and Type

What kind of model theory

composition only defined when preconditions are met => non-standard model theory

model = functor

Is reasoning computer friendly

- Cartesian closed category axioms are all Horn Rules
- subobject classification for a topos uses existential quantification

What about limit constructions

see next page

Computational Toposes Replace Existential Quantification With Constructors

...see the difference between the standard mathematical formalism and a computational one for subobject classification

Subobject classification

$m:S \to X$, monic(m)

$\exists\,!\ c:X \to \Omega$ such that

the diagram commutes

$$\begin{tikzcd} S \arrow[r, "!"] \arrow[d, "m"] & One \arrow[d, "true"] \\ X \arrow[r, "c"] & \Omega \end{tikzcd}$$

and the diagram is a pullback,

i.e., c(m) = true,

$\forall\ f:T \to X, \exists\,!.\ g:T \to S,$

g = m(f)

Subtypes

$p:X \to \Omega\ , \Rightarrow\ X\{p\}$ is a type

$incl_p:X\{p\} \to X,$

$p(incl_p) = true$, $monic(incl_p)$

$f:Y \to X \Rightarrow fac_{f,p} :Y \to X\{p\}$

$p(f) = true \Rightarrow\ p = incl_p(fac_{f,p})$

Characteristic maps

$f:Y \to X \Rightarrow\ char_f :X \to \Omega$

$char_f\,(f) = true,$

$char_{incl\_p} = p$

$p(f) = true \Rightarrow p(incl_{char\_f}) = true$

Inverse

$f:Y \to X \Rightarrow f^{-1}:X \to Y$

$monic(f), X = X\{char_f\} \Rightarrow$

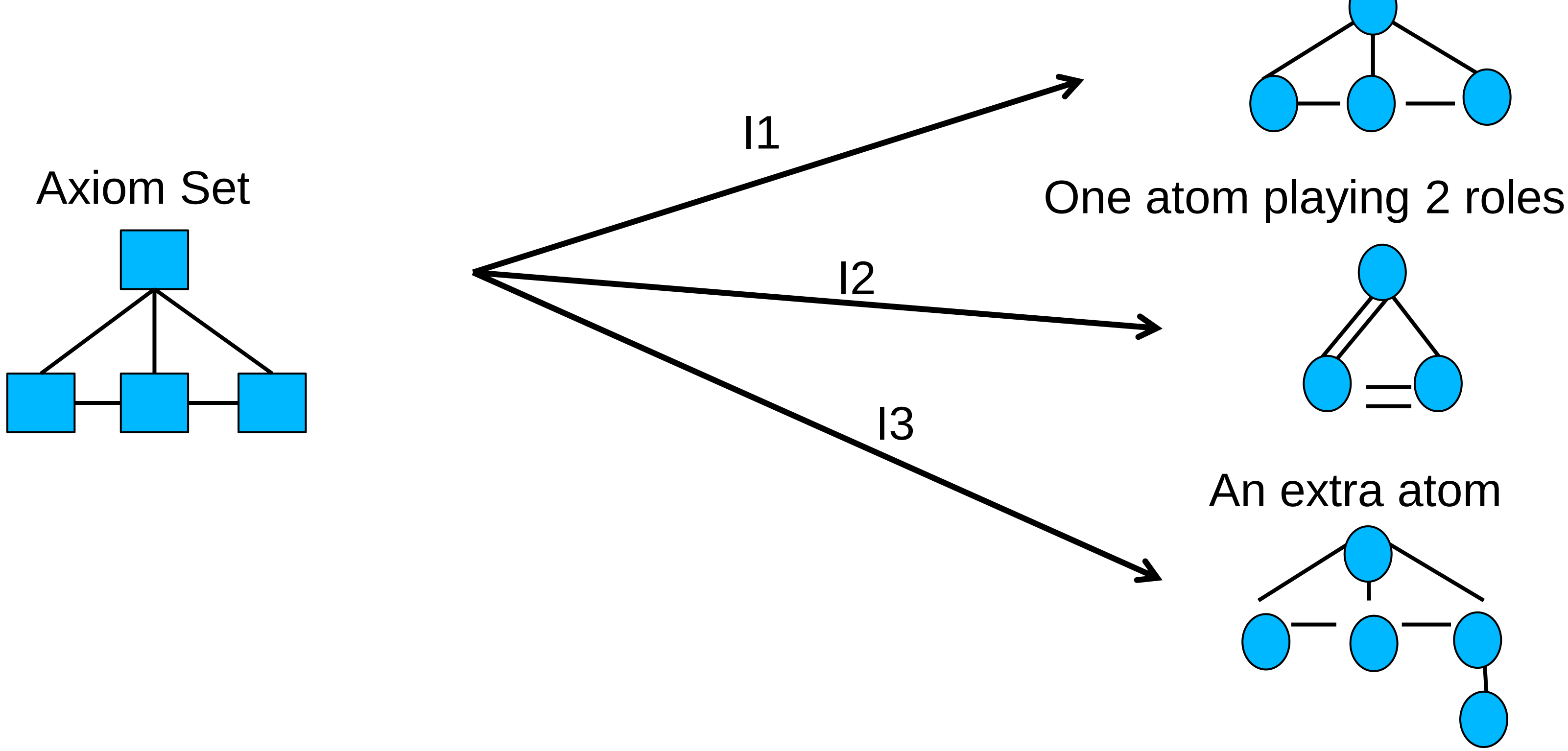
$f(f^{-1}) = id$

$f^{-1}(f) = id$

...the constructors, incl, fac, char all produce well-typed maps, as a function of their arguments, but you cannot use them until their preconditions are satisfied

Engineering Models Often Have Unintended Interpretations

… they underspecify intended interpretations

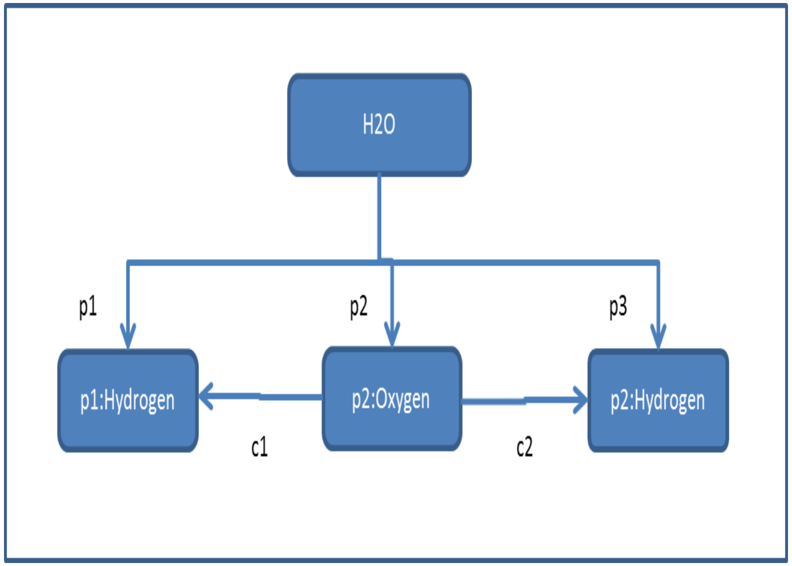


...result is that reasoning from the models leads to invalid conclusions, with regard to the intended interpretations

Magka, Despoina, Markus Krotzsch, and Ian Horrocks. “A rule-based ontological framework for the classification of molecules ”Journal of biomedical semantics, 2013.

How Can One Express The Assumptions Needed To Rule Out The Non-Intended Models For H2O

...including constructive topos axioms in the H2O axiom set together with specific axioms



Axioms

H2O, Hydrogen, Oxygen

p1:H2O → Im(p1)

p2:H2O → Im(p2)

p3:H2O → Im(p3)

H2O, Im(p1), Im(p2), Im(p3)

c1: Im(p2) → Im(p1)

c2: Im(p2) → Im(p3)

c1(p2)= p1, c2(p2) = p3

More Axioms

Im(pi) ⊥ Im(pj), i not = j

monic(p1), monic(p2), monic(p3),

Invertible(c1), Invertible(c2)

... constructive axioms and inference rules for a topos is a necessary for embedding engineering models, by not sufficient – we haven't ruled out extraneous connections

Axiom Sets With All of Whose Interpretations Are Isomorphic Using Templates

... a template is a meta level axiom schema

Unique Decomposition Template

MapConstant(f)

MapConstant(f),

PartMap(f)

ConnMap(f)

PartType(X)

Root(X) ≣ MapConst(f), Range(f) = X ⇒

f = nil (the empty map)

Root(Y), Root(X) ⇒ X = Y

last::PartPath → PartMap

firstPart:PartPath → PartPath

PartPath(f) ≣ f = firstPart(p).last(p)

Acyclic(f) ≣ Domain(last(p)) ≠ Range(firstpart(p))

Add

PartMap(p) ⇒ Monic(p)

ConnMap(f) ⇒ Invertible(p)

H2O has unique a decomposition as it satisfies the template

Root(H2O)

MapConstant(pi), MapConstant(ci),

PartMap(pi)

ConnMap(ci)

PartType(H2O), PartType(Im(p1),)
PartType(Im(p2)), PartType(Im(p3)),

PartPath(f), Root(Domain(f)) ⇒

Acyclic(f)

...model development tools can enforce conformity

What Does Constructive Topos Theory Contribute To Classical Engineering

...

- **A method to retrofit successful engineering modeling languages into a computational logic**
- **The ability to integrate sound inference for both automated reasoning and interactive verification**
- **Method to semantically integrate component models**
- **A formal interpretation semantics that can be used as part of tool validation**
    - Model theoretic semantics using sheaf theory for dynamic models

...many of same principles apply for cognitive science, other areas