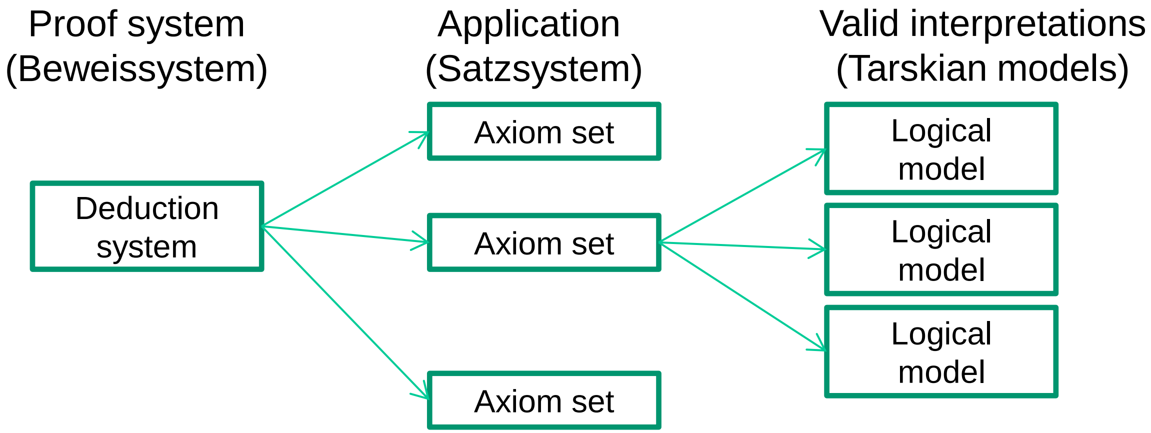
A Rule System For Engineering Modeling

Henson Graves

August, 2013

Foreword

- The rule-based paradigm discussed is intended for use in domains that are stable and where community collaboration is required; it is not for knowledge discovery or theory construction
- Graphically the paradigm is describe by the tree



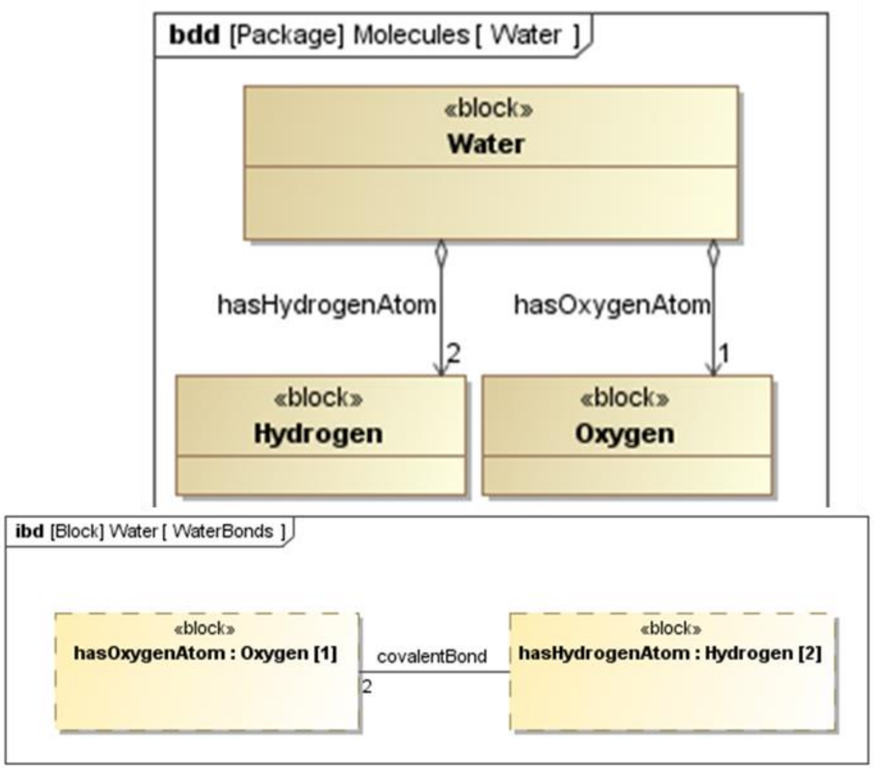
- The choice of deduction is based on computational tractability
- The axiom sets which represent domain applications generate theories, but much of the reasoning within a theory is to determine if a formula is in the theory of the axiom set
- An axiom set will, in general have multiple distinct logical models
- The theories generated form a lattice and knowledge development process is concerned with operations on theories, but the current talk is restricted to single axiom sets and their theories
- In this context an ontology is a general reusable axiom set which codifies domain knowledge, there are some ontologies implicitly used , but not explicitly mentioned

Outline

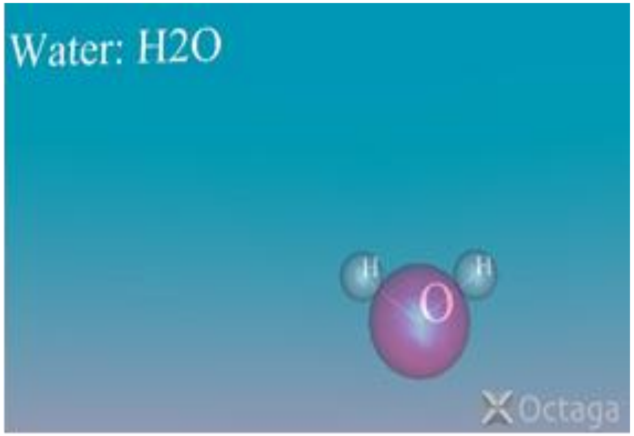
- **Use Cases**
- **Requirements Analysis for Use Cases**
- **Design solution for engineering modeling**
- **Mathematical Results**

Example 1: An Engineering Model of H2O and a Realization

Engineering Model



Realization
(Simulation)

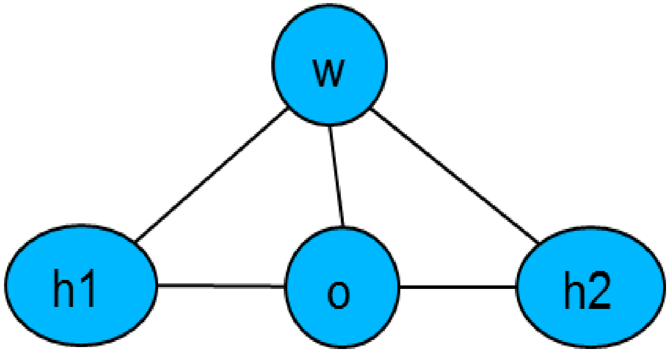


The full model contains enough information to generate the simulation

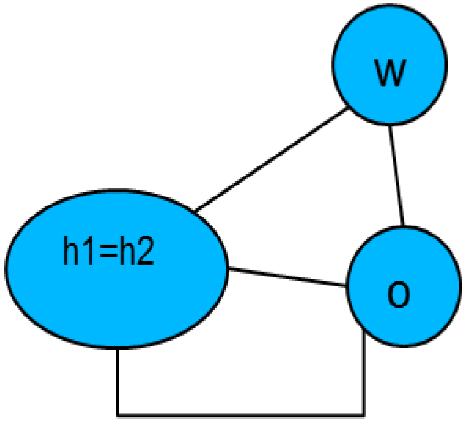
Graves, Integrating Reasoning with SysML, 2012

The H2O Graphical Model Has Multiple Distinct Realizations

… unless more information is added to the model



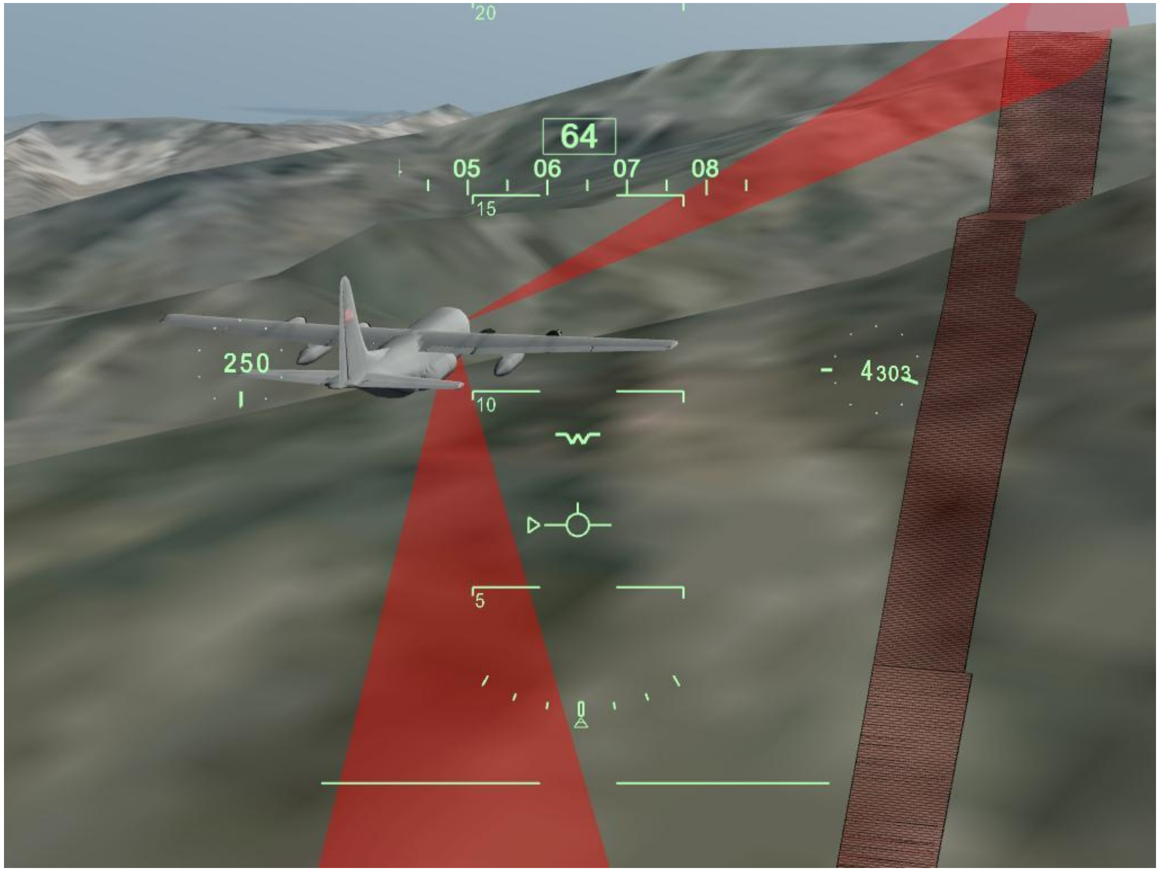
Standard Realization with 3 distinct atoms connected as expected



Realization where the hydrogen atoms are not distinct

Engineering models often underdetermine realizations, how to fix?

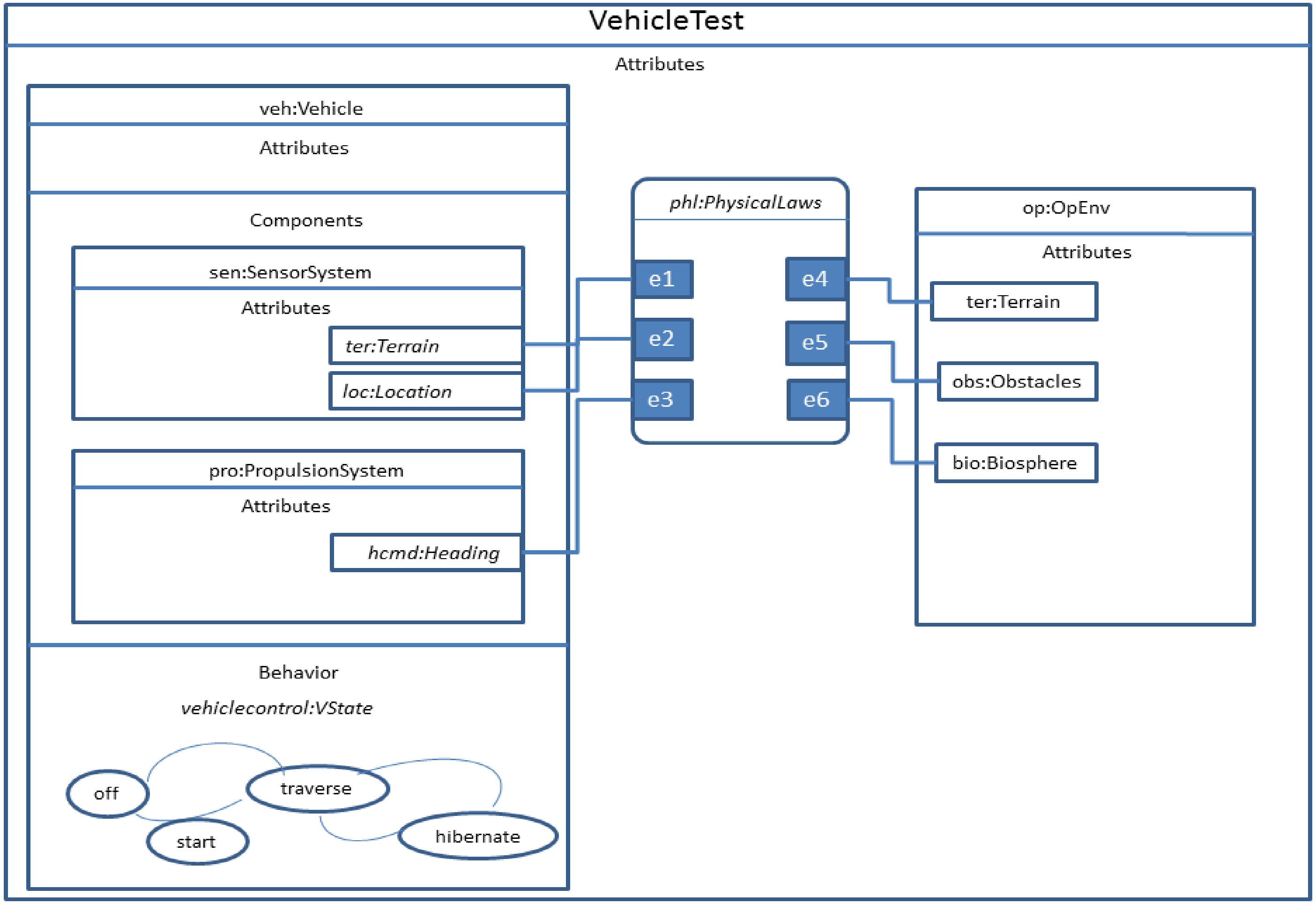
Example 2: Simulation of Vehicle Test Model



This is a snapshot from a dynamic real-time simulation

Graves et. al, Air Vehicle Model-Based Design and Simulation Pilot, 2009

Example 2: The Model From Which the Simulation Was Generated



This a view of a SysML vehicle test model used to generate the simulation. The behavior is described by state charts.

Requirements Analysis 1

- **Scalability**
- **Reasoning**
  - Computational tractability
  - Justification of correctness
- **Expressiveness sufficient for use cases**
  - Directed Graphs
  - Higher order logic
- **Practical Considerations**
  - Use familiar syntax and conventions for community
  - Integrate with existing languages and tools

Graves & Bijan, Using formal methods with SysML in aerospace design and engineering, 2012

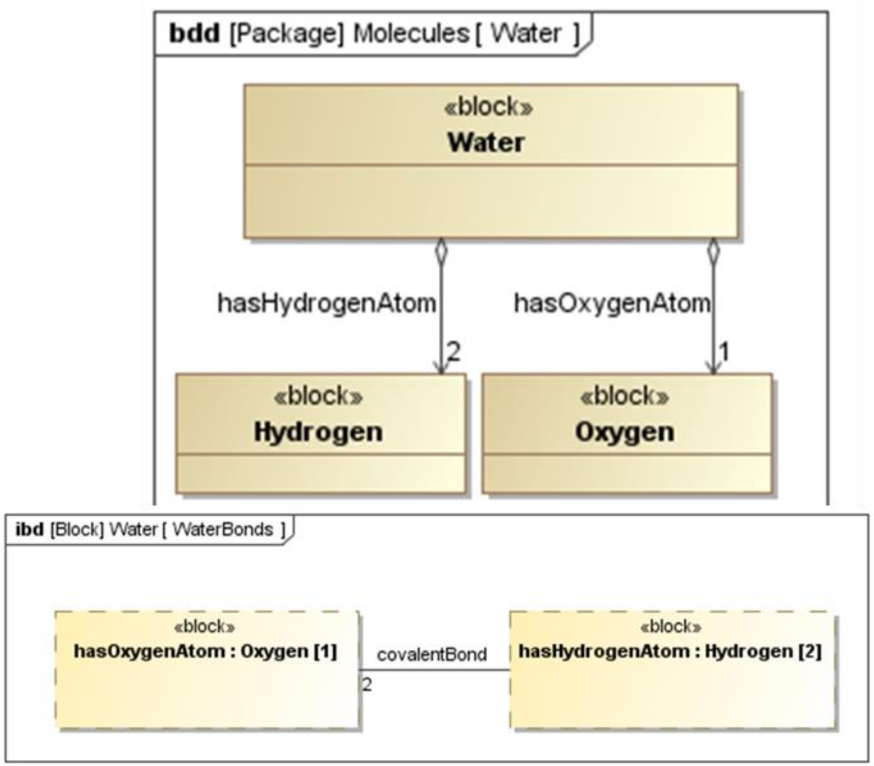
Requirements Analysis 2: Justification of Reasoning

- **Logic provides a paradigm for justifying correctness**
- **As in logic, reasoning from an engineering model is correct if it is true in all realizations (logic models)**
- **For H2O you need to add information to graphics, such as atoms are disjoint classes and the part maps all have distinct values.**
- **Then all realizations are structurally isomorphic and reasoning gives expected results**

Requirements Analysis 2: Representing Directed Graphs

Engineering Model

Embedding Model in Axiom Set



Signature

hasOxygen: Water → Oxygen
hasHydrogen: Water → Pow(Hydrogen)[2]

hasHydrogen1: Water → Hydrogen

covalentBond1: hasOxygen:Oxygen → hasHydrogen:Hydrogen

Axioms

Oxygen ⊥ Hydrogen
hasOxygen. covalentBond1 = hasHydrogen1

A directed graph can be embedded in the signature of the language when the signature has sorts for nodes and arrows with source and target functions

Requirements Analysis 2: Engineering Questions vs. Logic Questions

Many engineering questions are equivalent to consistency of axiom set

- **Most engineering design models are inconsistent establishing inconsistency is high value**
- **Models such as Vehicle Test are often inconsistent when physics laws are incorporated**

Analysis questions are equivalent to whether a formula is implied by the axioms

- **Capability analysis often has the form**

  $\text{Axioms} \vdash p.f.x \Rightarrow q.f.x$

  Where formula on the right is a Horn clause

Design Solution For a Rule System is:

- **First Order Horn Logic with equality**

  P1,…,Pn ⇒ Q

- **First order function symbols as map and type constructors with distinction between constructors which are first order functions and maps, e.g.,**

  A x B – for type constructor

  <a,b> - for map constructed using tuple constructor

- **Axioms for term constructions with additional application axioms, e.g., for H2O.**
- **Reasoning – unification and term rewriting**
- **Model theory does not require functions to be total, only defined when type conditions are met**

Graves & Blaine, Algorithm Transformation and Verification in Algos,1985

The Result is an Algebraic Form of Set Theory, called topos theory

firstBorn : Man x Women $\rightarrow$ Human
age : Human $\rightarrow$ Number
firstBorn.age : Man x Women $\rightarrow$ Number - dot is composition

isFather : Man $\rightarrow$ Ω,
{x : Man | isFather.x = true } $\sqsubseteq$ Man

father(mary; tom) = true
father : Human x Man $\rightarrow$ Ω,
fatherBy : Man $\rightarrow$ Pow(Human) – non-deterministic map

{x : Man | $\exists$ y:fatherBy(y,x) = true}

Highly expressive, first order Horn logic with two signature sorts, maps and types, different from HiLog as only uses constructors with computation rules

Behavior is Represented Within Algos By Adding State Space Axioms, e.g., time

Consider the notation

f():X

for a constant as a map with zero arity. Following topos theory

f():X ≡ f:One → X

Axioms such as

One = T, where T is linear discrete time, can be added.

The notation below can be read as “a at time t”

a@t = a|{t} = incl{t}.a, where t is a singleton

Time-based pre and post conditions can be written as

p.f.x@t ⇒ q.f.x.@t+k

Model theory for axiom sets with time is functions defined for time

Graves, Category Theory Foundation
For Engineering Modelling, 2013

Some Mathematical Results

- **Justification of reasoning**
  - Soundness, completeness for Horn clauses
- **Tractability of reasoning**
  - Canonical irreducible form for terms, at least the lambda calculus part
- **Usable graphics-based syntax**
  - SysML is faithfully embeddable in Algos
- **Expressiveness**
  - Contains a version of HOL
  - Contains an extended Description Logic with decidability conditions

Afterward

- It is not a given that rule systems are sufficient for KR in science and engineering. Map and type computation axioms required considerable engineering
- Effort is required to extend engineering graphics to full fledged models that can support reasoning
- In general the Lindenbaum-Tarski model is not the only valid model of application axiom sets
- The Algos rule system is practically usable as it can be integrated with SysML tools
- The Algos rule system subsumes Description Logic

Bibliography

Graves, H., Blaine, L., Algorithm Transformation and Verifcation in Algos, Third International Workshop on Software Speciation and Design. IEEE Computer Society Press. August 1985.

Graves, H., Horrocks, I., Application of OWL 1.1 to Systems Engineering, OWL Experiences and Directions April Workshop, 2008. http://owl1-1.googlecode.com/svn-history/r503/trunk/www.webont.org/owled/2008dc/papers/owled2008dc_paper_9.pdf

Graves, H., Guest, S., Vermette, J., and Bijan, Y., Air Vehicle Model-Based Design and Simulation Pilot, Simulation Interoperability Workshop (SIW). Spring 2009. http://www.omgwiki.org/MBSE/lib/exe/fetch.php?media=mbse:09s-siw-056_mbds.pdf

Graves, H., Bijan, Y., Using formal methods with SysML in aerospace design and engineering, Annals of Mathematics and Articial Intelligence, Springer 2011. http://www.omgwiki.org/MBSE/lib/exe/fetch.php?media=mbse:integration_15-mar-12.pdf

Graves, H., Integrating Reasoning with SysML, INCOSE International Symposium (IS 2012), 2012. http://www.omgwiki.org/MBSE/lib/exe/fetch.php?media=mbse:integration_15-mar-12.pdf

Graves, H., Category Theory Foundation For Engineering Modelling, draft 2013 http://www.omgwiki.org/MBSE/lib/exe/fetch.php?media=mbse:mathemataical_foundation_engineering.pdf