Extracting Model Structure for Improved Semantic Modeling

James Fairbanks, Georgia Tech Research Institute
jpfairbanks.com/doc/aske

12/5/2018

Goals

1. Extract a knowledge graph from Scientific Artifacts (code, papers, datasets)
2. Represent scientific models as a high level abstraction, (code as data)
3. Build metamodels by combining models in hierarchical expressions using reasoning over KG (1).

Running Example: Influenza

Modeling the cost of treating a flu season taking into account weather effects.

1. Seasonal temperature is a dynamical system
2. Flu infectiousness is a function of temperature

Running Example: Modeling types

Modeling the cost of treating a flu season taking into account weather effects.

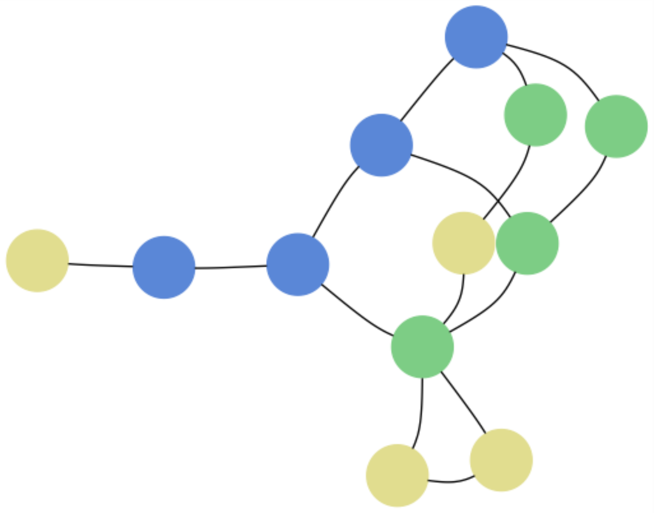
1. Seasonal temperature is approximated by 2nd order linear ODE
2. Flu cases is an SIR model 1st order nonlinear ode
3. Mitigation cost is Linear Regression on vaccines and cases

Scientific Domain

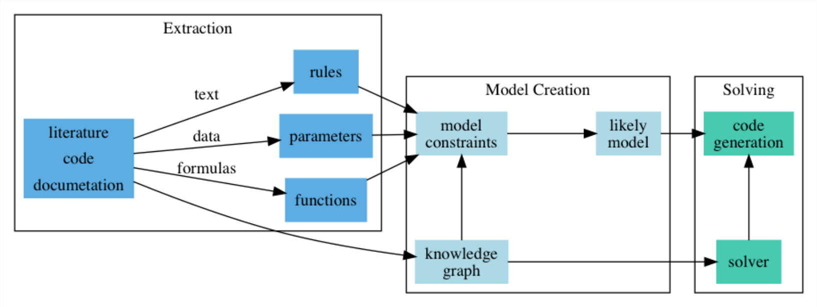
We focus on Susceptible Infected Recovered model of epidemiology.

1. Precise, concise mathematical formulation
2. Diverse class of models, ODE vs Agent-based, determinstic vs stochastic
3. FOSS implementations are available in all three Scientific programming languages

Graph of SIR Model



Knowledge Extraction Architecture



Example Input Packages

1. EMOD, Epimodels, NetLogo, and FRED are established packages, given their maturity and availability of published papers citing these packages.
2. Pathogen and NDLib are newer packages, we expect easier to work with and more future adoption.
3. Textbooks [Voit 2012] and lecture notes[1] will be a resource for these simple models that are well characterized.

[1] http://alun.math.ncsu.edu/wp-content/uploads/sites/2/2017/01/epidemic_notes.pdf

Model Representation and Execution

Representation of models occurs at four levels:

- **Executable**: the level of machine or byte-code instructions
- **Lexical**: the tradition code representation assignment, functions, and loops
- **Semantic**: a declarative language or computation graph representation with nodes linked to the knowledge graph
- **Human**: a description in natural language as in a research paper or textbook

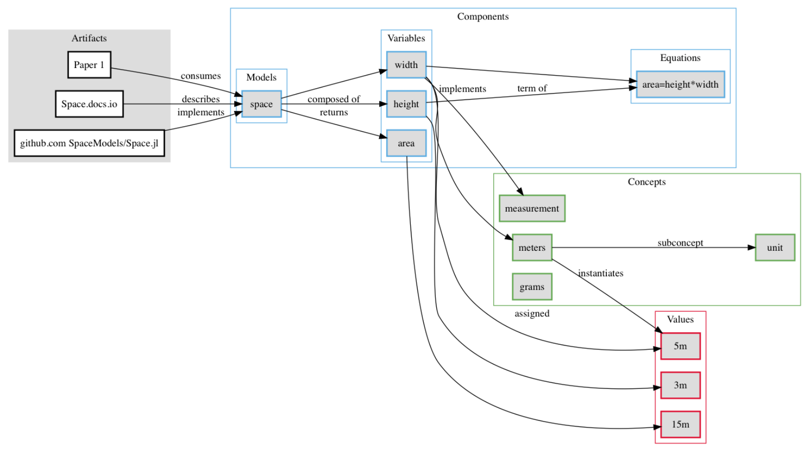
Model Representation and Execution

Julia provides tools at every level:

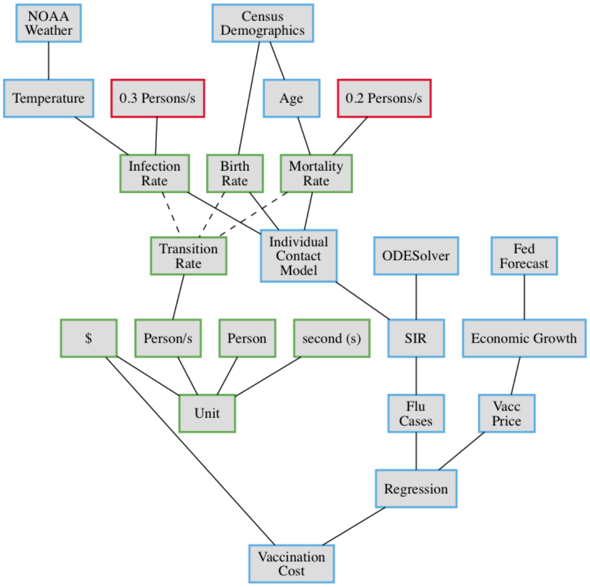
- **Executable**: LLVM bytecode and compiler
- **Lexical**: metaprogramming, Lisp style macros, code as data
- **Semantic**: type inference + computation DAG
- **Human**: Text, Jupyter Notebooks, and Graphviz diagrams

Knowledge Graph Schema

A preliminary design for types of knowledge in our knowledge graph.

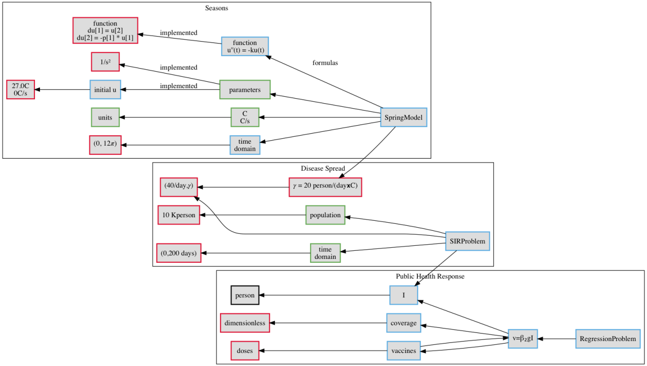


Knowledge Graph Sample



Flu Metamodel Pipeline

Here is the DAG for our running example. See FluModel Notebook for worked out example.

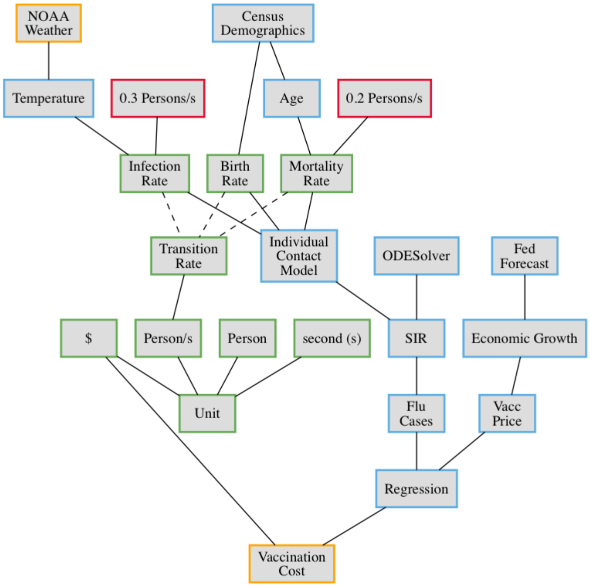


Pipeline of extraction

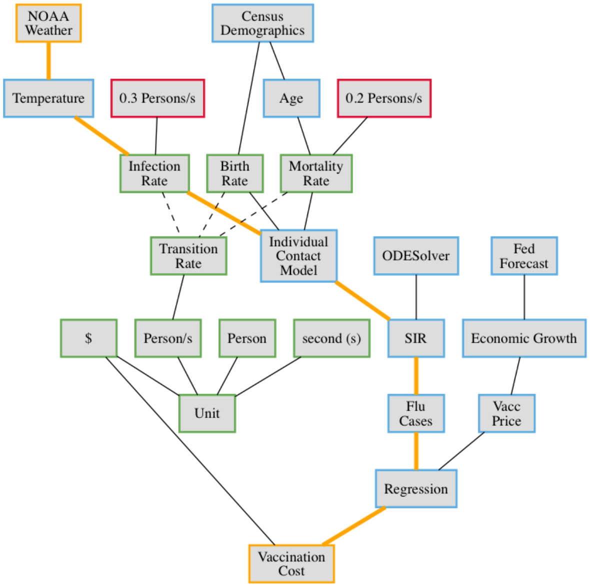
Knowledge Graph Reasoning

1. Define Model representations / KG schema
2. Extract KG from artifacts
3. **Reason over KG to build metamodel**
4. CodeGen/Execution of Metamodel

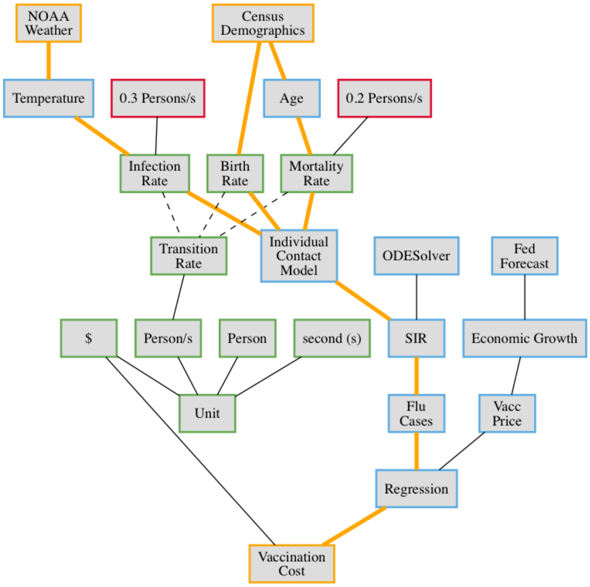
How do we get from Weather to Cost?



How do we get from Weather to Cost?



How do we get from Weather+Demographics to Cost?

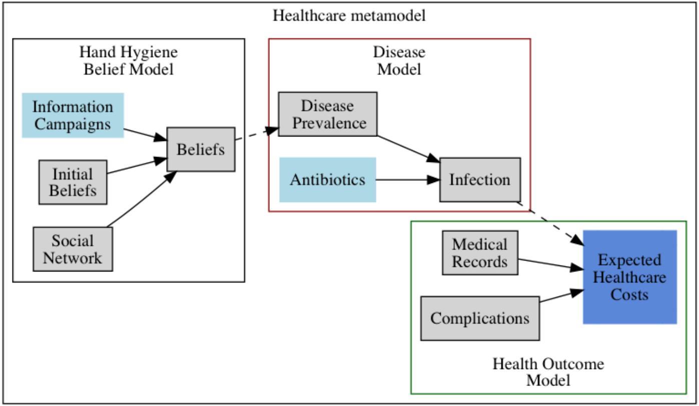


Knowledge Graph Reasoning Open Questions

- What rules for path/flow computations are necessary and sufficient for a metamodel?
- Can we implement those rules by choosing weights?
- How do we handle uncertainty and near matches?
- How do we determine “necessary dependencies” better than “connected component”
- What about supplying expert information?

Infectious Disease Metamodel

- A more ambitious example of a metamodel
- Requires Agent-based simulations of information diffuision and disease spread



A metamodel for predicting healthcare costs

Static vs Dynamic Graph

- Inherent tradeoff between flexibility and static analysis
- We will build the computation graph through the execution of code
- Metaprogramming will be used to generate the executable codes

Validation

- Extraction of KG elements from artifacts
- Metamodel construction
- Metamodel quality

Error and Residual

- Analogize the metamodel construction error and the model quality to the error and residual in numerical solvers. Given $f(x) = 0$ solve for x
- Measure both the error and the residual.
- Error $\mid x - x^\star \mid$, the difference from the correct solution
- Residual $\mid f(x) - f(x^\star) \mid$ or the difference from quality of optimal solution

Next Steps

- Incorporation of feedback today
    - the types of artifacts in scope
    - domain coverage and desired extensibility
    - inclusion/exclusion of particular package(s) and/or knowledge artifact(s)
- Construction of a proof-of-concept version of our knowledge graph and end-to-end pipeline
- Tailor running example to DARPA objectives
- A automatic transformation of models at the Semantic Level