ASKE Milestone 5 Report for AMIDOL

Eric Davis[1], Alec Theriault[1], and Ryan Wright[1]

[1]Galois, Inc

1 Introduction

The initial prototype of AMIDOL has been implemented as a proof of concept, and platform for developing the theories, algorithms, core architectures, domain specific languages, and intermediate representations from our original proposal. This prototype represents an advancement on the state-of-the-art for creating, solving, and analyzing complex models from scientific, physical, social, and hybrid domains. AMIDOL aims to reduce the overhead associated with the model life cycle and to enable domain experts and scientists to more easily build, maintain, and reason over models in robust and highly performable ways, and to respond rapidly to emerging crises in an agile and impactful way. The initial prototype achieves these goals by implementing several Visual Domain Specific Ontological Languages (VDSOLs), a novel intermediate representation whose representation is based on existing formalisms such as Markov models [10], Generalized Stochastic Petri-nets with inhibitor arcs [6], and stochastic activity networks [15, 17].

AMIDOL's VDSOLs are designed to lower the barrier for entry associated with formal modeling languages, and the long term costs associated with model maintenance. VDSOLs allow the expression of rich mathematical concepts using visual diagrams for systems and processes. Because AMIDOL uses a universal representation for this intermediate form, models defined with AMIDOL can be easily ported to off the shelf solution techniques which have already been vetted by the community, and optimized for high performance computing. AMIDOL models are also amenable to sharing, as their common intermediate representation means they can translated to different languages and representations, and composed with other models, even those generated with other formalisms.

The Phase 1 Prototype for AMIDOL shows a proof of concept of the core capabilities required to achieve this full vision in Phase 2, and has helped to identify the necessary next steps in realizing a the full framework as part of the ASKE program. We include in our prototype three VDSOLs to show flexibility, and two backend targets already in use by the community. The backend targets utilized by our prototype further help demonstrate our core capabilities by translating models in AMIDOL's IR into PySCeS based models, and systems of differential equations, showing AMIDOL's ability to utilize the IR to translate into other formalisms easily, automatically, and performably.

2 Initial Prototypes

The initial prototype for AMIDOL is implemented as a front-end user interface, using HTML, CSS, and Javascript, communicating with the Scala back-end using JSON over HTTP. This client/server approach leverages standard browser technologies for rapid development of rich interactions tailored for the specific needs of scientific modeling. It also decouples the concerns of visual presentation

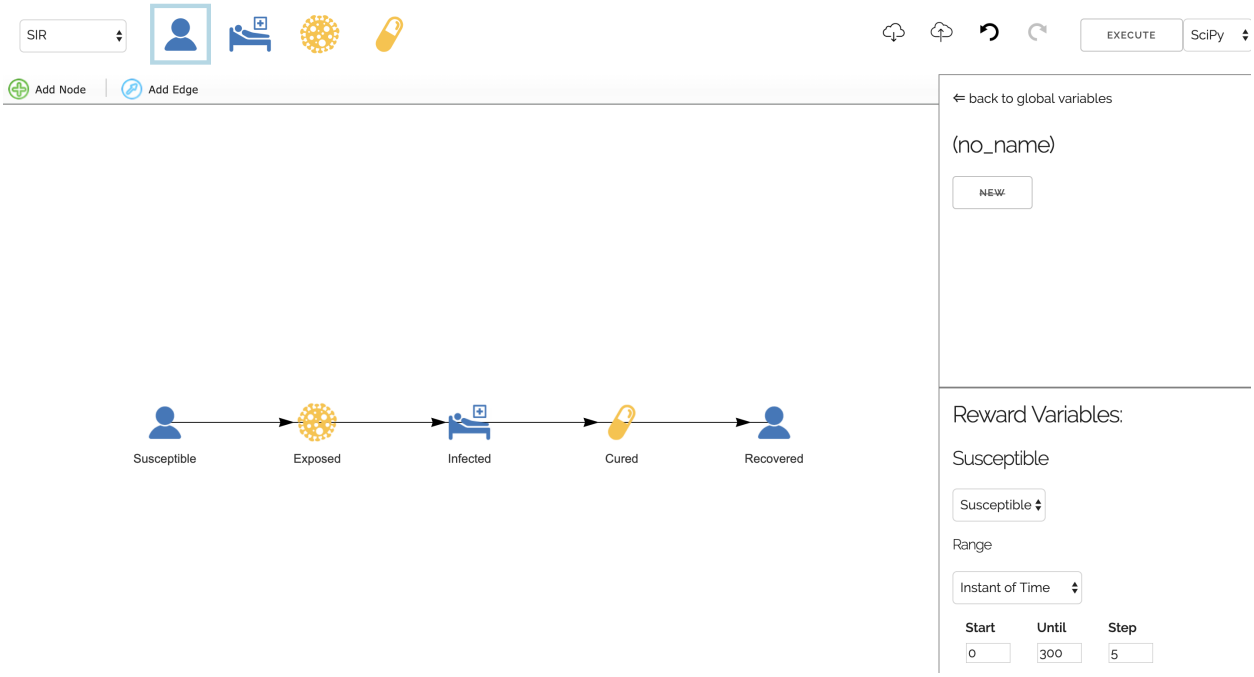

Figure 1: AMIDOL prototype with SIR Model

and manipulation from the underlying representations of model semantics and helps to hide the details from domain experts reducing their cognitive load when designing models.

Our initial prototype employs a rich visual editor allowing users to select a palette to work in, to save and load VDSOL diagrams, undo and redo operations, to customize parameters exposed by nouns and verbs, to define reward variables on a diagram, and to call multiple backend solver targets to solve those reward variables. Currently, the user interface implements a visual editor for directed graphs having labeled vertices and edges, which represent the nouns and verbs of a VDSOL.

Each VDSOL is implemented as a palette which is defined externally and provides a set of nouns and verbs specific to the domain, along with a translation protocol from the VDSOL into the IR.

Figure 1 shows the a standard SIR infection model implemented using the SIR palette. This model can be solved either through translation into ODEs, or to the PySCeS model formalism, but it can also be used an example of the power of VDSOLs and palette extension. The palettes defined for AMIDOL can be modified, and even extended, easily by adding new nouns and verbs. This allows for the long term curation of models by domain experts, who can easily extend their earlier work as new processes and interactions are discovered.

Figure 2 shows an example of such an extension, adding verbs for birth and death from a population. Our hope is this capability improves the longevity of models defined in AMIDOL, ensuring fewer models die on the shelf from neglect and bitrot, or simply are not reused due to changes in solver technology, or advancements in algorithms.

As an example of a wholly different palette for a VDSOL, and to demonstrate the capability of our Phase 1 prototype to handle models in different domains, we have also implemented a domain

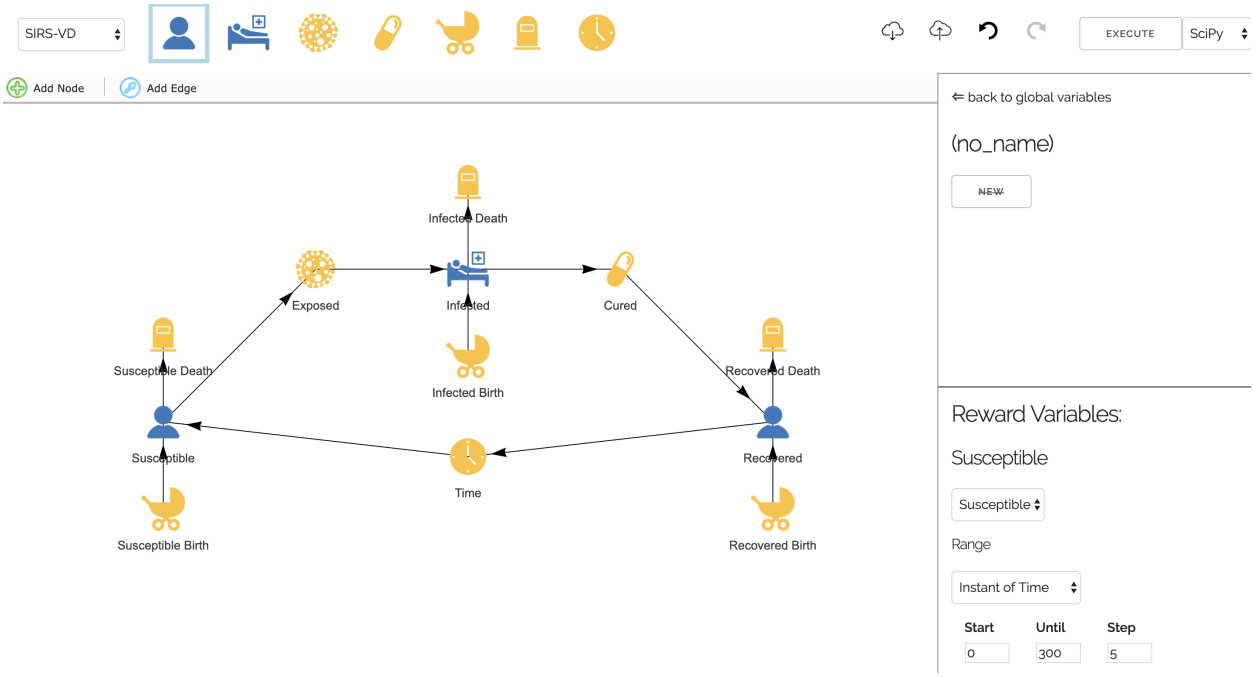

Figure 2: AMIDOL prototype with SIR Model with Vital Dynamics

model for the Lotka-Volterra model of predator-prey relationships, shown in Figure 3 [7, 4, 9]. Frequently used to describe the basic dynamics of biological systems with two species interactions, the Lotka-Volterra model assumes a single predator, and single prey species. While a simple model, this system is an example of a Kolmogorov model which is used as the underlying framework for modeling other dynamic systems, such as competing species in a niche, the impact of disease on a species, and mutualistic relationships. It is provided in our initial prototype primarily as another model for which it is easy to validate and generate expectations, and to demonstrate the capacity of our prototype to work in multiple domains.

$$\frac{dx}{dt} = \alpha x - \beta xy \tag{1}$$

$$\frac{dy}{dt} = \delta xy - \gamma y \tag{2}$$

3 Road Map for the Future

The Phase 1 prototype for AMIDOL has proved the core technologies and functionality required for success in this project are feasible, achievable, and provide the necessary functionality to support our vision for Phase 2. Phase 2 will work on extending this base functionality and building on top of our current architecture. During Phase 2 we will work with more complex models, including full models of H3N2 and H5N1 which include multiple geographic compartments. Phase 1 has also helped us to understand key challenges and capabilities which need to be developed in Phase 2 in order to make AMIDOL a more fully functional system for complex modeling. We divide these challenges into three core areas: **Model Creation**, **Model Analysis**, and **Long-term Goals**. The next sections will address these challenges, as well as the sub challenges identified below.

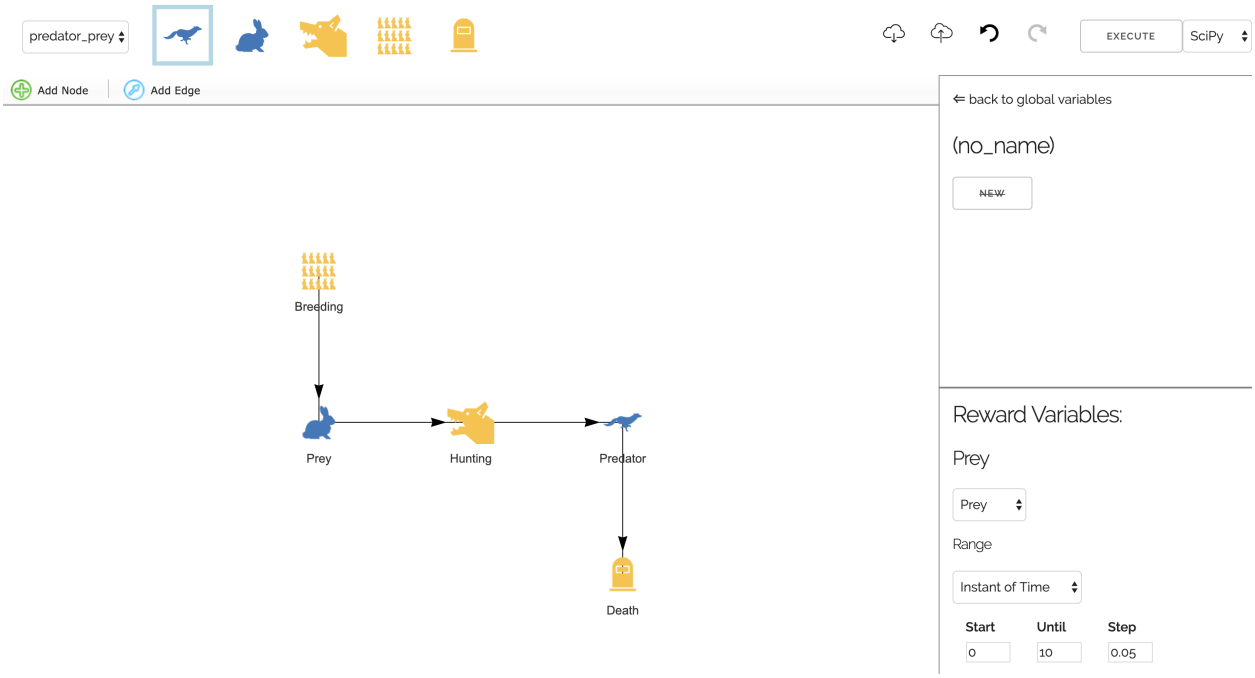

Figure 3: AMIDOL prototype with Lotka-Volterra Model

- Model Creation
  - VDSOL Expressivity
  - Composability of Models
  - Model Transformation and Optimization
- Model Analysis
  - Reward Definition
  - New Solver Targets
  - Results Database
- Long-term Goals
  - Model Synthesis

4 Challenges to Model Creation

4.1 VDSOL Expressivity

One major improvement our Phase 1 work has highlighted is the need to extend our VDSOLs to include support for multigraphs and hypergraphs [2, 3]. While current VDSOL palettes implemented by our prototype allow for some expressive models in their chosen domains, we have identified several additional capabilities regarding edge assignment, and node state-variable exposure, which warrant additional UI/UX innovation. Take, for example, the following model of viral replication [18, 8]:

$$\text{LTR} \overset{k_{\text{basal}}}{\rightarrow} \text{LTR} + \text{nRNA} \tag{3}$$

$$\text{nRNA} \overset{k_{\text{export}}}{\rightarrow} \text{cRNA} \tag{4}$$

$$\text{cRNA} \overset{k1_{\text{translate}}}{\rightarrow} \text{GFP} + \text{cRNA} \tag{5}$$

$$\text{cRNA} \overset{k2_{\text{translate}}}{\rightarrow} \text{Tat} + \text{cRNA} \tag{6}$$

$$\text{Tat} \overset{k_{\text{bind}}/k_{\text{unbind}}}{\leftrightarrow} \text{pTEFb}_d \tag{7}$$

$$\text{LTR} + \text{pTEFb}_d \overset{k_{\text{acetyl}}/k_{\text{deacetly}}}{\leftrightarrow} \text{pTEFb}_a \tag{8}$$

$$\text{pTEFb}_a \overset{k_{\text{transact}}}{\Rightarrow} \text{LTR} + \text{nRNA} + \text{Tat} \tag{9}$$

$$\text{GFP} \overset{d_{\text{GFP}}}{\rightarrow} \emptyset \tag{10}$$

$$\text{Tat} \overset{d_{\text{Tat}}}{\rightarrow} \emptyset \tag{11}$$

$$\text{cRNA} \overset{d_{\text{CYT}}}{\rightarrow} \emptyset \tag{12}$$

$$\text{nRNA} \overset{d_{\text{NUC}}}{\rightarrow} \emptyset \tag{13}$$

$$\tag{14}$$

This fairly simple model has some rather complex dynamics, and requires a VDSOL capable of implementing stoichiometry. In particular, our current palettes do not support the catalyzing reactions, such as that represented by the first equation. LTR is not consumed in this reaction, yet its concentration is vital to the state dependent rate represented by k_{basal}. In order to support this, our palette must support linking of inputs and outputs in a verb, to ensure easy definition of preserved reactants, and specification of partial and total orders on inputs and outputs when inputs and outputs cannot be treated as a single composed state variable.

The primary challenge here isn't theoretical, such systems can easily be created with modest extensions of AMIDOL's Phase 1 capabilities. They are primary around the user experience, and human-machine teaming aspects of VDSOL representation. We intend to explore this design space, and to include such UI/UX considerations in our extension to formal palette definitions.

4.2 Composibility of Models

In addition to increased expressivity from VDSOL extensions, AMIDOL is being designed to support the composition of individual models in Phase 2 to enable model reuse, compositional methods for solution, and to allow domain scientists to experiment by swapping out components of a model which may represent complex hypotheses about individual elements. Composition is an essential element of the long-term model development lifestyle for AMIDOL, and a key feature for collaborative science and knowledge extraction.

Composition is currently partially implemented in the Phase 1 prototype as the method for building an IR model from nouns and verbs in a given VDSOL. For Phase 2, AMIDOL will be extended, providing an editor in the UI to define custom compositions of atomic models via state-sharing. Variables which are shared between two composed models have the same marking, or value, and are effected by events in both models.

Composition will allow more rich model development, and model development as a modular and collaborative task, similar to working on separate objects or packages in code development. VD-

$$\text{translate}(\text{VDSOL}_A) \rightarrow \text{IR}_A \quad (15)$$
$$\text{compose}(\text{IR}_A, \text{IR}_B) \rightarrow \text{IR}_C \quad (16)$$
$$\text{translate}(\text{IR}_A) \rightarrow \text{target}_A \quad (17)$$
$$\text{optimize}(\text{IR}_A)) \rightarrow \text{IR}_B \quad (18)$$
$$\text{equivalence}(\text{IR}_A, \text{IR}_B) \rightarrow \text{True} \vee \text{False} \quad (19)$$

Figure 4

SOL representations of atomic models can be synchronized across members of a team, and edited individually, then composed in a similar way to object code with external references in software engineering.

4.3 Model Transformation and Optimization

The last major target for improving model creation in AMIDOL is the implementation of a full set of primitive transformations on the IR representations of a model. AMIDOL's architecture is designed in an analogous way to LLVM [12, 11] and similarly is designed for multistage optimization and transformation on the intermediate representation [13]. AMIDOL's IR is, in a sense, like a byte code language for models allowing the backend to perform transformations on models defined in this IR, acting on models via well structured transformations as shown in Figure 4. While VDSOLs provide benefits by being diverse and adapted to various domains, the AMIDOL IR provides benefits by being universal, allowing transformations to be leveraged across domains in a consistent and verifiable manner.

Currently AMIDOL implements composition operators as the basic operation needed to combine nouns, verbs, and reward variables into an executable model, and the translation of multiple VDSOLs into the IR. In Phase 2 we will extend AMIDOL to account for all of this proposed functionality.

5 Challenges for Analysis

In addition to challenges in model definition and development, Phase 2 of AMIDOL will address challenges for model analysis, interrogation, and querying. The primary ways this will be accomplished will include richer reward definition, new solver targets to support more robust analysis, and the implementation of a results database.

5.1 Reward Definition

The current Phase 1 prototype of AMIDOL implements only rate rewards for instant of time. During Phase 2 we will implement impulse rewards, and enable interval of time and steady state reward variables of both rate and impulse types. Steady state reward variables will make heavy use of our ability to perform transformations on the IR, as they represent an often wholly different solution target.

In addition to supporting a wider variety of atomic reward variables, Phase 2 will focus on extending reward variables to allow for composed rewards as expressions defined over atomic and composed reward variables. While atomic reward variables form the basis for model evaluation, AMIDOL

will support expressions defined over reward variables using basic arithmetic operations allowing reward variables to be composed via normal mathematical expressions.

5.2 New Solver Targets

Phase 1 supports a limited number of solvers, which mostly focus on interpreting models as systems of differential equations. During Phase 2 will support additional transformations to allow for discrete event solution as a target, along with numerical solution via successive matrix multiplication for models with finite state spaces.

Because different solution techniques are required by more complex reward variables, a key part of the backend for Phase 2 will be constraint identification and satisfaction, which will solve multiple reward variables with potentially different solvers and techniques for the same model, the results of which will be fed into a results database for later synthesis and composed reward solution. This backend architecture helps to ensure the performability of models designed in AMIDOL, and flexibility of solution technique helping researchers avoid the pitfall of being bound to a single target using hand coded models, which may or may not be performable.

5.3 Results Database

The final major component to our model analysis goals for Phase 2 is the creation of a results database to support explainability goals by tying the results of solving reward variables back to the primitives present in the VDSOL, storing intermediate results, allowing for the construction of complex composed rewards across models (allowing for model comparison), and allowing for the definition of "data-driven" models which can ingest traces, and real data from external sources as realizations of reward variables for model fitting, and testing.

Tying the results of model solution to elements in the VDSOL is a key goal of AMIDOL. Reward variable in models are currently defined on model primitives such as state variables and events, which are elements of our IR and not a given VDSOL. This is not ideal for our goals and desired outcomes which focus on expert interaction with the VDSOL while hiding details of the underlying model. During Phase 2 we will extend this definition to provide equivalent higher level constructs for rate and impulse rewards which are instead defined more naturally over nouns and verbs. We propose doing so by defining reward measurement points in an extension of our VDSO, indicating what a measure on a noun or a verb will return. This is a natural extension of our current palette definitions for VDSOLs which define input and output predicate connections. These new capabilities will allow users to naturally define reward variables and will allow us to infer information on **structural types** of nouns and verbs, determine equivalences, and capture knowledge from domain experts when they tell us they believe nouns in two different models represent the same abstract quantity.

The results of these reward variables will be stored in a database of prior results, and interacted with in a Design of Experiments interface. This Design of Experiments interface will allow a user to perform complex analyses on models, comparisons with past executions, comparisons with other models, and comparisons with empirical data using composed rewards. This interface and database will support the complex queries and prognostics required by the original BAA. This interface will borrow from concepts of Plackett-Burman designs, and factorial designs using individual reward variables and parameterizations defined over a given model as atomic components of expressions. Computations for given atomic components will be stored in the results database to avoid unneeded reevaluation, and to allow future experiments to build on past results, and be directly compared

and contrasted.

The Design of Experiments interface will also allow users to load data from external sources, and associate this data with expectations for state variables in the model. In our early tests we have achieved this by using data from the CDC's Fluview [5] data sets as series of instant-of-time observations. AMIDOL's interface will allow users to use regression to estimate the conditional expectation of dependent variables, given independent variables with fixed values from the data, allowing prediction, parameter estimation, and later risk assessment for a given regression through analysis of the distribution of possible realizations for state variables in a given model.

Our new interface will also allow direct comparison through user identification of reward variables, or expressions on reward variables, in two models which represent comparable properties or processes allowing the exploration of alternatives, conditional forecasting, counterfactual analysis, and comparative impact. Different strategies, configurations, or possible outcomes can be explored through examining different ways to parameterize a given model, or even differences in models with structural changes. Debugging experiments will be enabled through the connections to the VDSOL ontology. For instance, the interface will allow users to see the direct dependence of reward variables to noun's and verbs in the original ontology, better enabling researchers to understand the knowledge-based semantic dependencies normally hidden by traditional modeling techniques.

Because we will allow for factorial and Plackett-Burman experiment design users will be able to automatically perform one-factor-at-a-time [1, 16] sensitivity and uncertainty estimates. The initial prototype will also feature screening sampling-based methods [14] which have been shown to be computationally efficient and are further enabled by our VDSOL ontologies, as they help identify sources of uncertainty and error in the structure of the model. Correctness is enabled by loading external data from multiple time series or sources, and asking the interface to perform cross validation on the input. AMIDOL will automatically support k-fold cross validation on time series, allowing automatic partitioning of data sets.

6 Long-term Goals and Future Work

In addition to necessary features within AMIDOL we also intend to think about the ways in which AMIDOL will interact with the rest of the ecosystem being designed around the topic of automated scientific knowledge extraction, and will specifically be designing AMIDOL and it's representations to support model synthesis.

We intend to interact with other performers in the ASKE program, especially TA1 performers, to identify natural ways to import diagrams and models extracted from primary sources, and translate them into AMIDOL models. This will help show the expressive power of AMIDOL's system of model definition, and the suitability of the AMIDOL backend and IR as a universal model translation, optimization, and representation layer. We believe AMIDOL's results database, coupled with a graph representation of its VDSOLs allowing us to analyze provenance of structures within a model [19], possible equivalences, and to guide proofs of similarity or difference.

Our goals for AMIDOL are to have it adopted by the community, therefor we also plan to engage with members of the community who are undertaking efforts in domain modeling, and complex system analysis. We would like to build a long term community for the tool, and continue to grow it as an open source resource.

7 Resources, web sites, etc.

The current AMIDOL source code, including example models and documentation, is available at the AMIDOL Github site `https://github.com/GaloisInc/AMIDOL`.

References

[1] Robert Bailis, Majid Ezzati, and Daniel M Kammen. Mortality and greenhouse gas impacts of biomass and petroleum energy futures in africa. *Science*, 308(5718):98–103, 2005.

[2] Rangaswami Balakrishnan and Kanna Ranganathan. *A textbook of graph theory*. Springer Science & Business Media, 2012.

[3] Claude Berge. *Hypergraphs: combinatorics of finite sets*, volume 45. Elsevier, 1984.

[4] Fred Brauer, Carlos Castillo-Chavez, and Carlos Castillo-Chavez. *Mathematical models in population biology and epidemiology*, volume 40. Springer, 2012.

[5] CDC. National, regional, and state level outpatient illness and viral surveillance. https://www.cdc.gov/flu/weekly/fluactivitysurv.htm. Accessed: January 2019.

[6] Giovanni Chiola, Marco Ajmone Marsan, Gianfranco Balbo, and Gianni Conte. Generalized stochastic petri nets: A definition at the net level and its implications. *IEEE Transactions on software engineering*, 19(2):89–107, 1993.

[7] Herbert I Freedman. *Deterministic mathematical models in population ecology*, volume 57. Marcel Dekker Incorporated, 1980.

[8] Eric L Haseltine and James B Rawlings. Approximate simulation of coupled fast and slow reactions for stochastic chemical kinetics. *The Journal of chemical physics*, 117(15):6959–6969, 2002.

[9] Frank Hoppensteadt. Predator-prey model. *Scholarpedia*, 1(10):1563, 2006.

[10] Ronald A Howard. *Dynamic probabilistic systems: Markov models*, volume 1. Courier Corporation, 2012.

[11] Chris Lattner. Llvm and clang: Next generation compiler technology. In *The BSD conference*, volume 5, 2008.

[12] Chris Lattner and Vikram Adve. Llvm: A compilation framework for lifelong program analysis & transformation. In *Proceedings of the international symposium on Code generation and optimization: feedback-directed and runtime optimization*, page 75. IEEE Computer Society, 2004.

[13] Chris Arthur Lattner. *LLVM: An infrastructure for multi-stage optimization*. PhD thesis, University of Illinois at Urbana-Champaign, 2002.

[14] Max D Morris. Factorial sampling plans for preliminary computational experiments. *Technometrics*, 33(2):161–174, 1991.

[15] Ali Movaghar. Performability modeling with stochastic activity networks. 1985.

[16] James M Murphy, David MH Sexton, David N Barnett, Gareth S Jones, Mark J Webb, Matthew Collins, and David A Stainforth. Quantification of modelling uncertainties in a large ensemble of climate change simulations. *Nature*, 430(7001):768, 2004.

[17] William H Sanders and John F Meyer. Stochastic activity networks: Formal definitions and concepts. In *School organized by the European Educational Forum*, pages 315–343. Springer, 2000.

[18] Ranjan Srivastava, L You, J Summers, and J Yin. Stochastic vs. deterministic modeling of intracellular viral kinetics. *Journal of theoretical biology*, 218(3):309–321, 2002.

[19] Ryan Wright. Quine: A temporal graph system for provenance storage and analysis. In *International Provenance and Annotation Workshop*, pages 177–180. Springer, 2018.