Facilitating Model–Based Design and Evaluation for Sustainability

Natasha Jarus, Sahra Sedigh Sarvestani, and Ali Hurson
Department of Electrical and Computer Engineering
Missouri University of Science and Technology
Rolla, USA 65409
Email: {jarus, sedighs, hurson}@mst.edu

***Abstract*—Relating various models of a system is an essential part of model transformation, model composition, and other metamodeling tasks. The objective of this doctoral research is to create a provably correct approach to this problem.**

I. Introduction

Design and rigorous evaluation of a complex, sustainable system requires the creation of models to capture various aspects of the system's behavior. For instance, a sustainable water distribution network must have the capacity necessary to service all its customers and it must conserve water and power throughout the distribution process. It should be dependable in the face of pipe ruptures and other component failures and be secure from both physical and cyber attacks. All of this should be accomplished with lean physical infrastructure so that excess capacity is not wasted. Model-based design and evaluation is an effective approach to achieving these goals if all models used are accurate and consistent. In other words, the constraints imposed by one model and modeling formalism and the assumptions underlying them must not contradict those of any other. As a simple example, a reliability model that assumes two components are placed in parallel is not an accurate representation of a system topology where those components are placed in series.

The intended contribution of the doctoral research described in this paper is to create a broadly applicable and provably correct model transformation method. The first goal of this approach is to be applicable to a variety of complex systems and modeling formalisms and to be capable of complex transformation operations. For example, we seek to relate a continuous–time system dynamics model to a discrete–time control algorithm model, or a model of a system's resilience to a model of the system's topology. Our second goal is for every transformation to be provably correct. As the system design evolves, we should be able to propagate new information across all models of a system. To our knowledge, no existing approach addresses all of these challenges.

We postulate that sound approximation of semantics is key to provably correct model transformation. The semantics of a model arise from two sources: the semantics of the modeling formalism (e.g., discrete or continuous time, independence of state variables) and the semantics of the system that the model describes (e.g., interdependencies between components). Each model of a system serves as an approximation of that system's behavior; the extent of the approximation is determined by how well the semantics of the corresponding modeling formalism align with the semantics of the system.

This understanding of model semantics can be applied to other challenges encountered in developing or analyzing sustainable systems. A salient challenge in complex sustainable systems is model composition, which is closely related to model transformation. System–level models can be generated by modeling each component individually, then composing the models guided by the system's physical and functional topology. Composition of heterogeneous models, where the models can differ in formalism, is especially challenging, as it has to enable interoperation of the models' respective evaluation methods while correctly interpreting their respective semantics.

II. Research Contributions

We anticipate the following contributions at the conclusion of this research:

1) A theory of sound approximation of system and model semantics
2) A method for transforming system models
3) A method for heterogeneous model composition and evaluation

In previous work [1], we have outlined a model transformation approach based on *abstract interpretation* [2]. Initially developed for static analysis of computer programs, abstract interpretation is a theory of sound approximation of program semantics. The core of our proposed transformation is based on defining connections between each modeling formalism and a domain that captures the properties of the system. Each connection has certain attributes that allow us to demonstrate that it constitutes a sound approximation of the system semantics. These connections can then be used to transform a model from one formalism to another, as well as to propagate changes across models in various formalisms. For example, from a reliability model we can deduce part of a system's semantics, including independence or interdependence of specific components. Based on these semantics, we can construct a set of system topology models whose semantics are compatible with those deduced from the reliability model.

978-1-5386-7466-6/18/$31.00

Model composition can be facilitated by interpreting the outputs from one model in terms of another model's semantics. Once reinterpreted, these outputs can be used as inputs to this second model. This process can be automated to allow for co-simulation. It may also be possible to perform this interpretation at a higher level of abstraction, creating hybrid model formalisms.

III. Abstract Interpretation of Models

Transforming models between formalisms is generally not a one–to–one operation. For instance, when converting a topology model into a reliability model, the topology model encodes no information about component reliability. Therefore each topology model is consistent with multiple reliability models. Furthermore, defining model transformation directly between two formalisms is not scalable: for n formalisms, we have to define $O(n^2)$ transformations.

We address both of these issues by formalising model transformation in terms of abstract interpretation. Rather than relating a model of one formalism to a model of another, we relate sets of models. This allows the formalism to capture the approximating nature of system modeling; since each model abstracts some part of the system's semantics, it is necessarily the case that every model is consistent with several different systems. Thus, we say that a given topology model describes any system with that topology, regardless of the reliability of that system's components.

To make the approach scalable, we introduce a *system properties domain* that encodes system properties such as component reliability and interdependencies. Model transformation is split into *concretization* and *abstraction* steps.

Concretization maps a model to the properties that hold for the systems described by that model; i.e., it generates a set of systems described by a given model.

To complete the transformation, a set of models is abstracted from the given properties, producing a set of models consistent with the initial model. Each formalism requires us to define only abstraction and concretization functions, meaning that for n modeling formalisms we define $O(n)$ operations.

Figure 1 illustrates this transformation for modeling formalisms $\mathbf{Model_1}$ and $\mathbf{Model_2}$. First, a model $m_1 \in \mathbf{Model_1}$ is lifted to the set $\{m_1\}$. Concretization takes a set of models (in this case, $\{m_1\}$) and produces the properties that hold for all those models. These properties are then abstracted to a set of models of the $\mathbf{Model_2}$ formalism. All the models in this set are consistent with the initial model; the system designer must then select the correct model from this set by introducing new information not present in the existing system properties. For example, transforming a parallel topology model into a set of reliability models will produce models of the reliability of a parallel system, but the designer must specify the reliability of each component to select a single reliability model from the results.

Dividing transformation into abstraction and concretization also offers a means to formalize the soundness of transformations. Let α be the abstraction operator for a given modeling

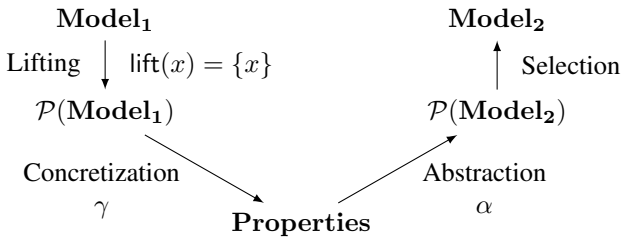

Fig. 1. Transforming Sets of Models

domain $\mathbf{Model}$ and γ the associated concretization operator. Furthermore, we will define a *specificity order* on all our domains. For $\mathcal{P}(\mathbf{Model})$, this is the subset ($\subseteq$) relationship: fewer possible models in a set means more information about the system in that set. We define $\mathbf{Properties}$ such that it has an order operator $\sqsubseteq$ which behaves in a similar fashion.

By requiring $(\mathbf{Properties}, \alpha, \gamma, \mathcal{P}(\mathbf{Model}))$ to be a Galois connection, i.e., requiring the following properties for α and γ, we can ensure that these transformations preserve the soundness of the models on which they operate.

$$P \sqsubseteq (\gamma \circ \alpha)(P), \quad \forall P \in \mathbf{Properties} \tag{1}$$

$$(\alpha \circ \gamma)(M) \subseteq M, \quad \forall M \subseteq \mathcal{P}(\mathbf{Model}) \tag{2}$$

Equation (1) informs us that if we start with some properties of a system, abstract a set of models from them, then concretize system properties from those models, we get properties that are at worst less specific than the initial ones. This allows abstraction to discard properties not relevant to the semantics of the modeling formalism, but forbids it from producing contradictions. Equation (2) requires the properties domain to fully represent each modeling formalism. (1) and (2) combine to make α and γ sound with respect to system and model semantics.

IV. Current Status

We are currently creating formalisms for system reliability and topology, which will demonstrate the constraints reliability places on topology and vice versa. The bulk of our body of research is on quantitative modeling of large–scale networked systems, with emphasis on cyber–physical critical infrastructure such as smart grids and intelligent water distribution networks. One objective of our work on model transformation is to increase the impact and applicability of these quantitative models by enabling transformation of models from one application domain to another, e.g., power to water, or from one system attribute to another, e.g., resilience to survivability. This will considerably accelerate the adoption of sustainable infrastructures by facilitating their assurance.

References

[1] N. Jarus, S. Sedigh Sarvestani, and A. R. Hurson, "Models, metamodels, and model transformation for cyber–physical systems," in *7th International Green and Sustainable Computing Conference*, pp. 1–8, Nov. 2016.

[2] P. Cousot and R. Cousot, "Abstract interpretation: a unified lattice model for static analysis of programs by construction or approximation of fixpoints," in *4th ACM SIGACT-SIGPLAN Symposium on Principles of Programming Languages*, pp. 238–252, ACM, 1977.