Automatic Verification of Dynamic Constraints in LTI Control Systems Through Model Transformations

Sean Whitsitt, University of Arizona {whitsitt@email.arizona.edu}

Keywords— Model Transformations, Dynamic Constraints, Controls

1 Introduction

Using modeling tools it is possible to construct Domain Specific Modeling Languages (DSMLs) for cyber-physical systems. In developing these languages a metamodeler can specify structural constraints that prevent modelers from constructing invalid systems. However, these structural constraints do not prevent modelers from constructing systems that do not meet design requirements such as preventing roll-overs in autonomous ground vehicles or collisions between unmanned aerial vehicles. These requirements are referred to as dynamic constraints. In control systems dynamic constraints can be considered any requirement on the system that necessitates mathematical analysis (in second order systems: percent overshoot, rise time, settling time, etc.). The research proposed herein will lay the foundation for the process of incorporating dynamic constraints into the construction of DSMLs by exploring the process as applied to dynamic constraints in linear time-invariant (LTI) control systems.

2 Background and Motivation

The process for developing a DSML for a cyber physical system is well understood. In fact, creating DSMLs for cyber-physical systems is a common practice [1, 4]. However, existing research focuses on the structural concepts of modeling these systems. The outputs from these languages are structurally correct to operate their respective systems, but they may not meet the dynamic constraints on those systems. Instead, users of DSMLs are left to handle the verification of the output artifacts themselves.

Automatic verification of models is also a well understood field [3, 2]. However, most of the research into automatic verification deals with verifying that the structure of the artifacts being verified is correct or with moving the problem from a domain that is difficult to analyze to one that is simple to analyze. This still leaves users of cyber-physical system DSMLs to verify the dynamic constraints of their systems by hand.

The main motivation behind this research is safety. The verification of safety-critical requirements for cyber-physical systems is paramount to the development process. Incorporating dynamic constraint verification into the generation of output artifacts for DSMLs would standardize the process of showing that a system meets its requirements. Note that–as many techniques already exist for analyzing LTI systems–this research is not proposing to develop new techniques. This research is only proposing to implement existing techniques automatically within a new framework for DSML development.

3 Proposed Research

Fig. 1 shows the extension to the DSML process that is being proposed for LTI systems. Consider the modeling process as applied to a control system. For example, a user of Simulink might draw an LTI system and simulate that system providing some important information. Currently that is the extent of what the tool can do. Incorporating this proposed research into Simulink would extend it such that it could use the information it provides to the user to either suggest or implement changes to the original system to meet some desired specification.

This research will be performed in several stages. First, a simplified testbed DSML will be developed for basic LTI systems. Second, an interpreter will be developed to translate the models produced in that language into code that can be analyzed by external tools. Third, a basic set of possible dynamic constraints will be introduced into the DSML. Fourth, model transformations that can impact the basic set of dynamic constraints will be identified. Fifth, the interpreter will be extended to automatically run the analysis code and link the results to specific model transformations. Sixth, the interpreter will be extended to implement the necessary model transformations and to rerun the analysis code. This process will repeat until a solution is converged upon or for a specified number of iterations at which point the interpreter will return to the user with a failure notification.

The model transformations used in this research will be of two possible different types. First, the DSML will allow a user to indicate specific gain values in the system that can be altered by the transformations. Second, the DSML will give the user the ability to indicate where new controllers can be attached to the original system. This will allow for compensators–such as lead/lag or observers–to be added through model transformations.

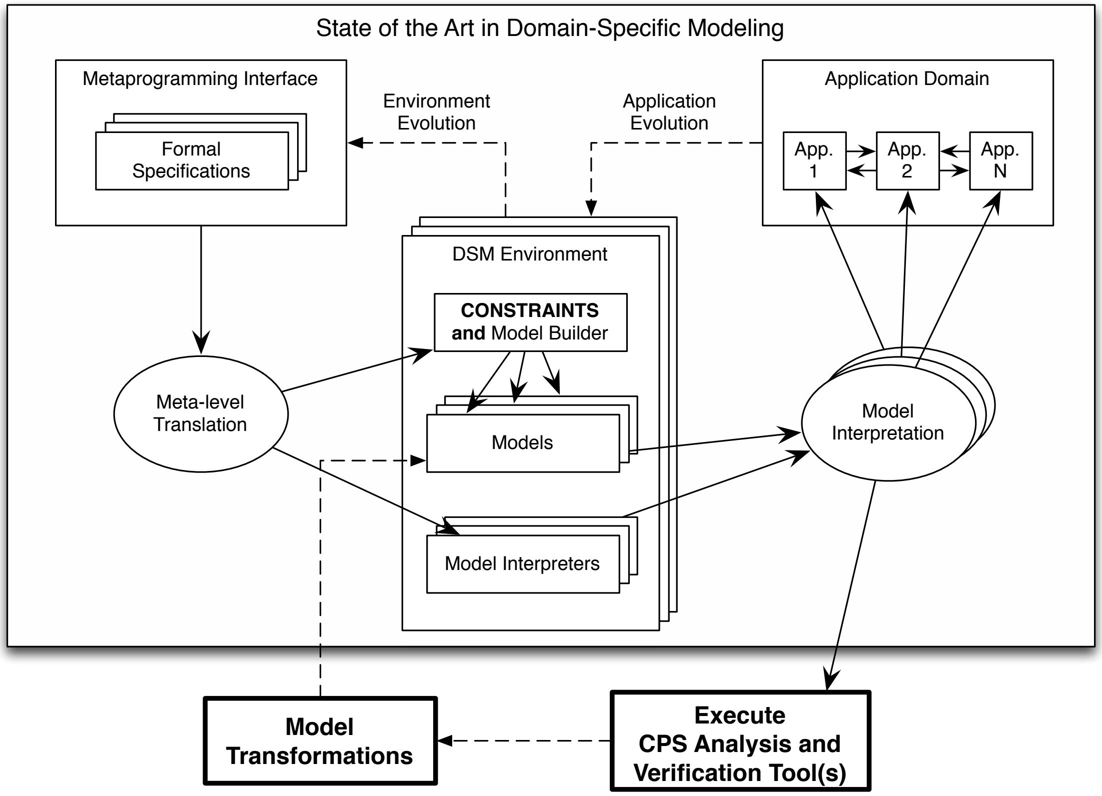

Figure 1: The modified DSML process with the model transformation loop.

4 Potential Impact

The result of this research will be an abstraction of the process for requirement verification for LTI control systems. More importantly, this will impact future research into DSMLs for control systems and cyber-physical systems by laying the ground work for automatic dynamic constraint verification through model transformations. Additionally, automatic verification will allow users that are not experts in the domain covered by the DSML to use the language and generate verified output artifacts. This could potentially reduce the costs associated with the development cycle of these systems. Also, A system could be more quickly adapted to changes to its dynamic constraints. Finally, changes to the constraints of a system could be managed by non-experts that may understand what the system needs to do, but not how to make certain that it does.

References

[1] D.J. Claypool, T.J. McNevin, W. Liu, and K.M. McNeill. Automated software defined radio deployment using domain specific modeling languages. In *Mobile WiMAX Symposium, 2009. MWS '09. IEEE*, pages 157–162, July 2009.

[2] L. Pedro, L. Lucio, and D. Buchs. Principles for system prototype and verification using metamodel based transformations. In *Rapid System Prototyping, 2006. Seventeenth IEEE International Workshop on*, pages 10–17, June 2006.

[3] A.A. Sani, F.A.C. Polack, and R.F. Paige. Model transformation specification for automated formal verification. In *Software Engineering (MySEC), 2011 5th Malaysian Conference in*, pages 76–81, Dec 2011.

[4] Sean Whitsitt and Jonathan Sprinkle. Modeling autonomous systems. *AIAA Journal of Aerospace Information Systems*, 10(8):396–413, 2013.