**Proceedings of the ASME 2012 International Design Engineering Technical Conferences &
Computers and Information in Engineering Conference
IDETC/CIE 2012
August 12-15, 2012, Chicago, Illinois, USA
DETC2012-70710**

GUIDING AND VERIFYING EARLY DESIGN USING QUALITATIVE SIMULATION

Matthew Klenk
Palo Alto Research Center
Palo Alto, CA, USA

Johan de Kleer
Palo Alto Research Center
Palo Alto, CA, USA

Daniel G. Bobrow
Palo Alto Research Center
Palo Alto, CA, USA

Sungwook Yoon
Palo Alto Research Center
Palo Alto, CA, USA

John Hanley
Palo Alto Research Center
Palo Alto, CA, USA

Bill Janssen
Palo Alto Research Center
Palo Alto, CA, USA

ABSTRACT

Design of a system starts with functional requirements and expected contexts of use. Early design sketches create a topology of components that a designer expects can satisfy the requirements. The methodology described here enables a designer to test an early design qualitatively against qualitative versions of the requirements and environment. Components can be specified with qualitative relations of the output to inputs, and one can create similar qualitative models of requirements, contexts of use and the environment. No numeric parameter values need to be specified to test a design. Our qualitative approach (QRM) simulates the behavior of the design, producing an envisionment (graph of qualitative states) that represents all qualitatively distinct behaviors of the system in the context of use. In this paper, we show how the envisionment can be used to verify the reachability of required states, to identify implicit requirements that should be made explicit, and to provide guidance for detailed design. Furthermore, we illustrate the utility of qualitative simulation in the context of a topological design space exploration tool.

INTRODUCTION

The field of qualitative reasoning has its roots in capturing human reasoning about the physical world. Such reasoning about the interactions of connected elements is at the heart of an early design process, where a designer is attempting to achieve some desired overall behaviors, and avoid unwanted interactions. Consider the drivetrain model in Figure 1. A qualitative analysis will show that it can move smoothly up through the gears, increasing speed over level terrain. But it will also show that the engine may stall because of excessive load for certain combinations of design parameters, and driving and terrain patterns. This qualitative analysis can provide guidance for parameter and component selection during detailed design. By linking our qualitative reasoning system to a standard tool for interactive graphical design (Open Modelica, http://www.openmodelica.org/), we are enabling designers to use qualitative analysis as part of their standard work practice.

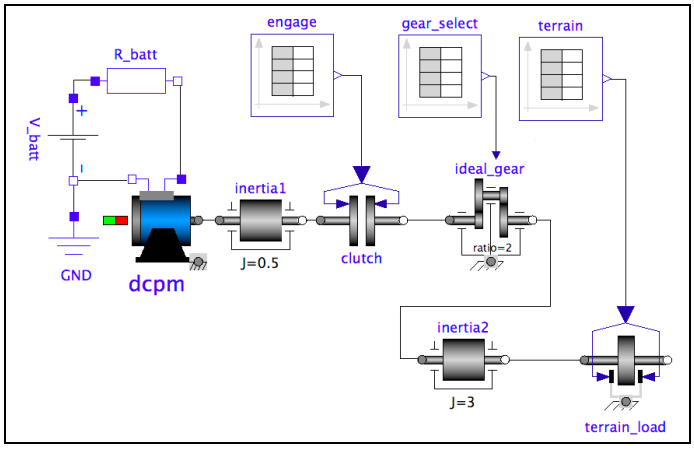

FIGURE 1: MODELICA MODEL OF A VEHICLE DRIVETRAIN

While developing models of systems with fully specified parameters, engineers frequently have to determine whether their numerical results conform to expected behaviors or are in fact errors in their modeling or simulation. This process relies on an understanding of the constraints on possible dynamics of the system (e.g., when the engine is running, and the vehicle is in a forward gear it should not go backwards, it is possible for the engine to stall, etc.). Qualitative reasoning automates this form of reasoning.

While many new designs are instantiations of previous successful systems that leverage new components and/or capabilities of materials, innovative design requires exploring a larger space of designs including new topologies of

components. Doing detailed parametric design for each element of the space is costly; qualitative verification helps prune this space by efficiently analyzing component topologies without the need to specify all component parameters needed for numeric simulation. Qualitative modeling supports the rapid exploration of designs that are only specified using the mathematical form of the relationships between a component's inputs and outputs. The systems do not need to be piece-wise linear; non-linear models are fine. Given a model, qualitative simulation generates all possible behavioral trajectories of the system's variables. Analyzing these trajectories can determine whether with appropriate parameter selection, a design could satisfy the requirements, or whether it can never fulfill certain requirements.

This paper begins by introducing qualitative simulation, its representation and semantics. We then discuss our design architecture QRM and our approach to creating models. We illustrate this using an example of a door system based on an infantry fighting vehicle, and highlight how qualitative simulation verifies requirements and guides detailed design by identifying implicit failures. In a second example dealing with electric circuits, we show how qualitative simulation drastically prunes a design search space. We close with a discussion of related approaches, scaling and future work.

QUALITATIVE SIMULATION

Qualitative simulation [1][2][3], or envisioning, is the process of projecting forward, from an initial situation and a model, all possible qualitative states that may occur. Qualitative representations of continuous quantities (e.g., the voltage across a diode) are central to this process. In our familiar Newton-Leibnitz calculus we use variables to represent quantities that can take any value from the real number line, and vary with time. Variables can have arbitrarily many higher-order derivatives. Likewise, in qualitative reasoning, these variables and their derivatives take on values – except that the values are qualitative. Each variable (or derivative) has a *quantity space* consisting of an ordered set of *landmark values* representing important points for understanding the behavior of the model (e.g., the turn-on voltage for a diode). A qualitative value is either a landmark or the open interval denoted by two adjacent landmarks. For a door, there are two landmark values: *Closed* and *Open*. The doors position can be at one of these two landmarks, or between the (Closed, Open). The qualitative value also has a direction (a qualitative derivative) of *increasing, decreasing* or *steady*. The most common quantity space uses just the sign of the real quantity. We represent the interval x<0 as Q-, x=0 as Q0, and x>0 as Q+).

A qualitative *state* is an assignment of qualitative values to variables in the model. We represent equations as qualitative constraints. Consider the equation governing a resistor, V=I*R, where voltage, V, and current, I, are quantities and R is a fixed parameter with a positive value. The resulting multiplication constraint ensures that the qualitative product of I and R is V. Because R is a positive constant value, if I is a negative value, then V must also be a negative value. Furthermore, their derivatives must also match. Figure 2 defines qualitative addition and multiplication for sign values.

+	Q-	Q0	Q+
Q-	Q-	Q-	?
Q0	Q-	Q0	Q+
Q+	?	Q+	Q+

*	Q-	Q0	Q+
Q-	Q+	Q0	Q-
Q0	Q0	Q0	Q0
Q+	Q-	Q0	Q+

FIGURE 2: QUALITATIVE ARITHMETIC TABLES

Non-linear relationships are captured in a number of ways. For polynomials, simple power constraints are used. For other non-linear functions, such as exponentials, we use the M+ constraint

One of the most significant consequents of the coarseness of qualitative values is that variables may be qualitatively constant for long periods of time (perhaps infinite). Hence, qualitative simulation need only consider the instants of time at which there is a possible change in qualitative value,. The passage of time is represented as an alternating sequence of instants and intervals. A qualitative state can either describe an instant or an interval. Qualitative simulation determines all trajectories through the qualitative state space from an initial state. Given a state, qualitative simulation computes possible successors for each quantity value and uses constraints to determine how they may be combined to form a next, if any, state or states. The rules for generating successor values and directions are based on the mean value theorem from calculus [4].

Consider a position quantity that was between the open and closed landmarks and moving toward closed. There are four possible successors for this quantity. Its value may remain in the interval or reach the closed landmark and it may continue increasing or become steady (its derivative stays positive or becomes Q0). Figure 3 illustrates the qualitative integration rule for an instant to the following interval where are variables are continuous.

∫	Dec	Std	Inc
Q-	Q-	Q-	Q-
Q0	Q-	Q?	Q+
Q+	Q+	Q+	Q+

FIGURE 3: CONTINUITY OF NEXT VALUES FROM AN INSTANT

From basic calculus if a variable is non-zero at an instant, it will remain at that qualitative value in the following interval. If the variable is 0, it will have the qualitative value of its derivative over the following interval. There is one ambiguous case: if a variable and its derivative are both 0, the qualitative value on the following interval is ambiguous (but the variable and its derivative must be qualitative equal during the interval). Consider x=t^2 when t=0. The qualitative values x and dx/dt are both Q0, but x=Q+ on the following interval. For other nonlinear equations, M^+ and M^- constraints are used to create piece-wise monotonic functions [3].

Cyber-physical systems include dynamics that are discrete as well as continuous (e.g., an input signal to open the door, the changing of gears in a drive train, a diode switching from off to on). We model such changes through *modes*, which include an entry condition, initial values for variables, and equations that are valid within that mode. During simulation, discrete changes occur at instants when mode entry conditions are satisfied. The initial values and equations govern the behavior of quantities in the following interval and subsequent states. Modes are different than the operating regions in that they allow for the modeling of hysteresis.

QUALITATIVE SIMULATION SEMANTICS

For qualitative reasoning to be useful for verification it must have a well-defined semantics. One can prove a theorem: Given a qualitative model with the appropriate abstractions for the ODE's used in, say Modelica, to define continuous behavior for a numeric simulation, the qualitative simulation will contain a path which describes the trajectory of the numeric simulation [3].

This is illustrated more concretely in Figure 4. This particular example is of an electric vehicle, but the details of the model are irrelevant here. Qualitative simulation produces envisionments from qualitative models (left downarrow). An envisionment corresponds to a real system in the following way. First, the qualitative models describe an infinite number of possible systems (all possible numerical assignments to parameters as well as all possible conventional component models which satisfy the qualitative model, including non-linear ones). Each of those systems will have a particular behavior (right downarrow). Each such behavior will map to a sequence of qualitative states (leftarrow). Each possible real behavior corresponds to a trajectory in the envisionment. Hence, if a desired behavior does not appear in an envisionment, it cannot occur in with any possible assignment of parameters to this system. This is an extremely important property.

One would also like the converse to be true: that every state in the envisionment can actually occur in a real system. Although this property often holds, and there are complex conditions under which it holds. However, we cannot guarantee it in general [6]. Elimination of spurious transitions and states has been an active research area in the qualitative reasoning community.

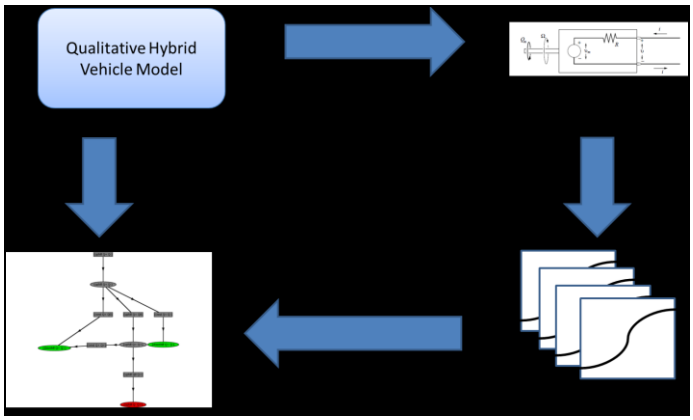

FIGURE 4: QUALITATIVE SIMULATION SEMANTICS

DESIGN ARCHITECTURE

In this section, we describe our view of an automated, or semi-automated, design process (shown in Figure 5). A human designer or an automated Design Space Exploration (DSE) tool starts with a high level functional specification of the desired system to be designed. This search produces tentative topologies for analysis. These topologies are expressed in the Modelica connection language only specifying components and their connections; it need not contain any parameter values.

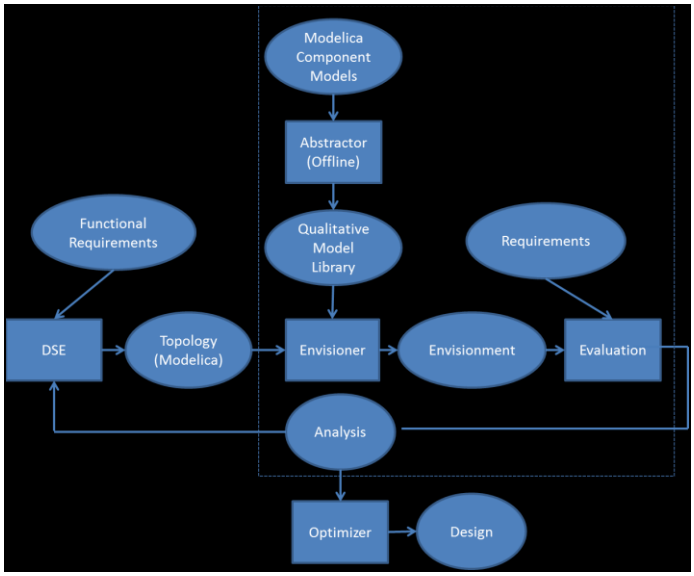

FIGURE 5: QRM SYSTEM ARCHITECTURE

Given the qualitative models of the components and their connections, the envisioner constructs the envisionment of the system. The requirements (converted to qualitative terms) are evaluated against the envisionment. Some requirements may be met, some may not. If the requirements are not satisfied, this analysis identifies which requirements fail to be met and why. This is then presented to the designer or the automated Design Space Exploration tool. If the requirements are adequately met, subsequent analysis selects parameter values which optimize the

requirements. This optimization may discover no assignment of values to parameters meets the numerical requirements in which case this analysis will be fed back to the designer or DSE.) This paper focuses on the fully implemented qualitative reasoning aspect of this process which we call the Qualitative Reasoning Module (QRM).

Our design architecture uses component models from a standard library. The vast majority of our qualitative model library is obtained from Modelica models which are abstracted only once, and comprise 'well written' Modelica models abstract directly. More complex Modelica models require human intervention. Inclusion of Modelica function blocks, algorithm blocks or complex conditionals are difficult to translate automatically. These abstractions need be done only once and form the qualitative component model library.

BUILDING QUALITATIVE MODELS

Component-based modeling is becoming increasingly popular in industry (e.g., Modelica [5]) due to savings incurred by reusing existing models for new applications. Component modeling efforts take lots of resources; therefore, we align our models as much as possible with Modelica to facilitate our ongoing automatic translation efforts (see the corresponding models in Figure 6). The composition of models occurs through connections that are domain specific (e.g., electrical pin). The composition of the models creates additional constraints on the flow and effort variables of the models governed by Kirchhoff's current and voltage laws. One area where we differ from Modelica representation concerns our use of modes instead of conditional equations. Modes offer the following advantages: (1) they localize the definition of hybrid behavior for the component, and (2) they provide a natural way to model various faulty behaviors.

Modelica	QML
model Capacitor extends onePort; parameter Capacitance C; equation i = C * der(v); end Capacitor	(defprototype Capacitor :extends onePort :parameters ((C)) :equations ((= i (* C (deriv v 1)))))

FIGURE 6: QML CAPACITOR MODEL IS SIMPLE TRANSFORMATION FROM MODELICA

To illustrate our modeling approach, Figure 7 contains our definition of an ideal diode. We present this here in our internal S-expression syntax which highlights this localization.[1] This model is a subclass of the electrical one port model, which defines two electrical connections, a positive pin and a negative pin, and variables for the current and voltage of the diode. The redefinition of the voltage variable v is essential to create the quantity space including `Q0`, representing 0V, and `OnVoltage`, representing the turn on voltage for the diode. The diode has two modes, `off` and `on`. The component is in a mode until the entry conditions for another mode have been satisfied, in this case, if the diode was `off`, the equation stating that no current was passing through the diode would be enforced. This persists until the instant when the voltage transitioned to `OnVoltage`, at which point the equation holding the voltage constant would be enforced and current would be allowed to flow through the diode.

```
(defprototype ideal-diode :extends (one-port)
   :variables ((v voltage :landmarks (Q0 OnVoltage)))
   :mode (off :entry ((= i Q0))
              :equations ((= i Q0)))
   :mode (on :entry ((= v OnVoltage))
             :equations ((= v OnVoltage))))
```

FIGURE 7: DIODE MODEL WITH TWO MODES

QUALITATIVE SIMULATION FOR VERIFICATION

In addition to specifying a topology of connections between qualitative component models, it is necessary to encode requirements in a formal language. We work with a variety of temporal logic specifications [7]. While Linear Temporal Logic is common in verification, Computational Tree Logic is a sibling of LTL that is better suited to qualitative verification of requirements over multi-trajectory envisionments. Requirements may be evaluated over an individual qualitative state in the envisionment (e.g., a variable should never exceed a particular level); requirements may also take into account a sub path of a trajectory.

Success and failure conditions for our envisionment algorithm can terminate simulation along a trajectory when one of the conditions is met. The requirement, "the door shall not overshoot the closed position," considers a state a *failed* if the door's position is below the closed landmark.

After the envisionment graph has been created, QRM provides the following analysis. If none of the trajectories violate requirements, then all possible numeric values for the system parameters will satisfy all requirements. (Recall the completeness guarantee of the envisionment graph.) If some trajectories violate requirements and others do not, then the design may satisfy the requirements with appropriate constraints on parameter values. In this case, detailed design is required to determine an assignment of parameter values that will satisfy the requirements. If all of the trajectories violate requirements, detailed design is not necessary because no set of parameter values will satisfy the requirement.

[1] We have defined a standard template for expressing modes that is acceptable to current Modelica compilers, but do not include it here.

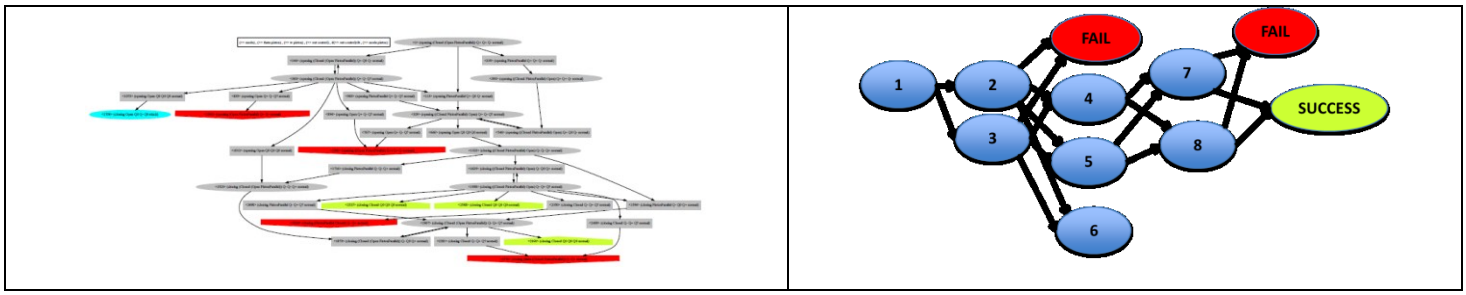

FIGURE 9: THE ENVISIONMENT GRAPH FOR THE DOOR SYSTEM MODEL
THE ENVISIONMENT COMPUTED BY QRM IS ON THE LEFT, AND A SIMPLIFIED VERSION ON THE RIGHT

VERIFICATION EXAMPLE: VEHICLE DOOR LINKAGE

To illustrate qualitative simulation consider the door system shown in Figure 8. The architectural model shows quantity spaces for the positions of the piston that moves the door, and the door itself. The system consists of a PD controller, which uses position and velocity sensors from the door, a piston, whose linear motion applies a torque on the door, and finally the door slab itself. An input signal to the controller specifies the desired position for the door. In this case, the door has two landmarks in the angular position quantity space, closed and open, and the piston has one landmark on the linear position quantity space, piston parallel, representing the position where the piston acts in parallel with the hinge. We will evaluate this design against the requirements that the door should always be able to be closed, the door's position should operate between the door open and door closed position inclusively. For a context of use, from an initial situation in which the door is closed, we will consider two discrete transitions: (1) the command is given to open the door, and (2) when the door has reached the open position, the command will be given to close the door.

QRM produces the envisionment (shown to illustrate scale in Figure 9) providing the following feedback to the designer. The design may reach a successful situation (shown in green).

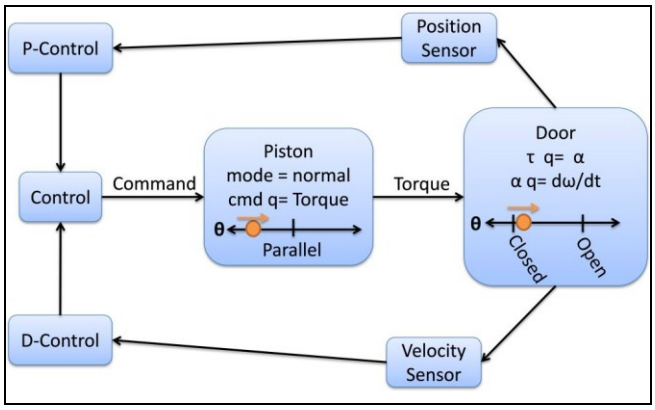

FIGURE 8: ARCHITECTURAL OF THE DOOR SYSTEM

Red nodes in the graph are qualitative states that violate requirements. Therefore, appropriate parametric assignment will be needed to ensure that trajectory for each failed state is avoided. A metric for estimating how difficult it will be to verify the design is the ratio of successful states to terminal states, in this case 1/3.

Further analysis of the envisionment provides additional guidance for detailed design. There is a terminal situation, (node 6 in the simplified graph, cyan in the full graph), that does not satisfy the success or failure conditions of the system. This dead-end state implies the need for additional requirements to guide the designer to avoid this state. In the example, this situation results from a kinematic singularity in the piston door connection. That is, when the acting angle of the piston is parallel to the angle of the door, the piston produces no torque. While this is part of the piston component model, it only leads to a quiescent (terminal) state if the door is stationary at this point. To identify this risk requires simulating the system with a use case where the door first opened and then closed. This analysis happens very early in the design process, when alternative system topologies are being considered. In the next section, we illustrate how qualitative verification could be used within an automated design space exploration system.

QUALITATIVE VERIFICATION IN TOPOLOGICAL DESIGN SPACE EXPLORATION

Innovative design searches for configurations of existing components (new topologies) to achieve some specified functionality. Consequently, this search space is exponentially large in the number of components in the design. Qualitative verification prunes the design space in two ways. The first is use of qualitative models of components, where the component models capture only significantly different behaviors of the models. That is, each qualitative component model corresponds to a many quantitative component models. The second is use of a qualitative simulation to identify bad topologies from which no choice of parameters will satisfy the requirements, and to guide parameter selection in detailed design. Qualitative simulation graphs are much smaller than those that explore parameter spaces. Therefore, qualitative verification can eliminate designs for large parts of the parameter space.

Consider the following example of designing a system that turns on a light after a short delay of a switch being flipped. If the available components include batteries, switches, resistors, capacitors, inductors, and diodes, the topological design space includes every configuration of these components. To illustrate the utility of qualitative verification, we will consider a design space exploration tool that searches the design space by taking one of the following design actions: adding a component in parallel or series with an existing component, removing a component, or flipping a component in the circuit. Figure 10 illustrates the starting design, which includes just a battery, switch and diode.

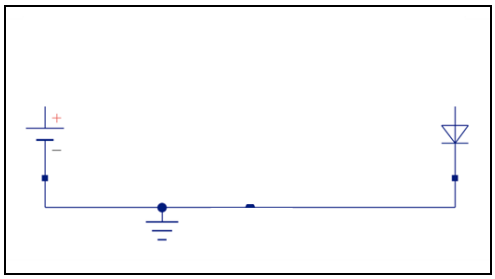

FIGURE 10: STARTING POINT FOR TOPOLOGICAL DESIGN SPACE EXPLORATION

After each design action, we attempt to build a qualitative model and simulation for the current design candidate. Now many of these candidates are actually shorted or open circuits and QRM identifies them because their initial conditions are inconsistent. If the design candidate has consistent initial conditions, QRM generates an envisionment and analyzes the results. Consider a circuit with a resistor in completing the circuit in Figure 10. The envisionment of this will begin with both the switch and diode off, and has two trajectories for the instant the switch is turned on. In one, the diode is on, and, in the other, the diode is off. The trajectory of the actual system depends on the ordinal relationship between the on voltage for the diode and the battery's voltage. Because neither of these trajectories satisfies the requirement that there exists a delay before the light turns on, qualitative verification eliminates this topology without considering all possible combinations of battery voltages, resistances and on voltages.

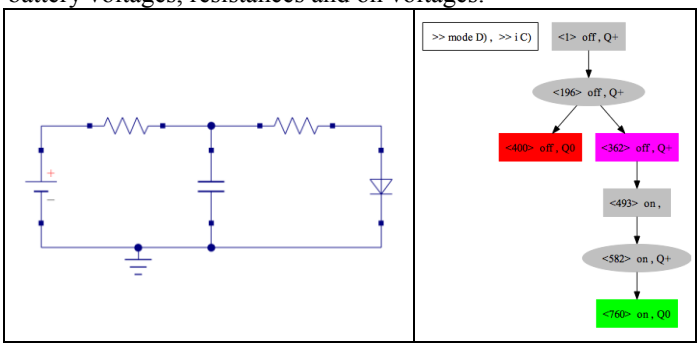

FIGURE 11: THE ENVISIONMENT ON THE RIGHT PROVES THAT THE TOPOLOGY ON THE LEFT CAN SATISFY THE DESIGN REQUIREMENTS

Now consider the design in Figure 11. QRM produces an envisionment with two trajectories. They are identical in the interval after the switch is turned on, the capacitor is charging and the voltage across the diode is increasing. This interval terminates in one of two instants: (1) the current ceases flowing into the capacitor and the system reaches a steady state, and (2) the voltage across the diode reaches the on voltage landmark causing a mode transition (shown in magenta) resulting in the diode turning on. This second trajectory satisfies the requirement. Therefore, this topology is a candidate for parameter selection and care should be taken to ensure that the battery voltage is greater than the turn on voltage of the diode.

SCALING

One of the promises of Qualitative Reasoning applied to design is its performance. By answering simple questions, requirements can be evaluated for surprisingly complex systems very quickly. We draw on decades of experience on building fast qualitative envisioners. In particular, we draw on advances developed in the recent DARPA Deep Green program [8].

Qualitative Reasoning has the advantage it only needs to address qualitative distinctions – that alone often severely contracts the search space. The complexity of QRM is not driven by the number of variables in the system – it is more determined by the dynamics of the system. If the dynamics are simple, analysis will be simple. We can successfully analyze systems of tens of thousands of variables in seconds. On the other hand, we can construct a pathological example with a few dozen components that cannot be solved (e.g., the voltage across a series of unsynchronized oscillators).

One important way QRM improves its performance (first developed in Deep Green) is to include requirement evaluation during envisioning. If a state or a combination of states do not meet the requirements, QRM immediately cuts off generation of any subsequent states: after all there is no necessity to analyze the consequences of states that do not meet requirements.

QRM also includes a qualitative solver which determines when qualitative variables are locked (state dependent) together and thus can be completely eliminated. For this, it uses a form of qualitative algebra. This greatly reduces the complexity of most analyses.

COMPARISON TO OTHER APPROACHES

There is a broad literature on formal verification of hybrid systems. However, almost all approaches require quantitative models and numerical parameters. Such information is often not available in early design. In contrast to our approach of constructing a model from components, verification with HybridSAL[10] begins with a set of equations, with numeric parameters chosen. HybridSAL [10] is also limited to linear models.

HybridSAL has the advantage of being able to answer quantitative questions about a design (e.g., will the vehicle

reach 30 mph in 6 seconds). Answering such queries for fully specified designs is an important part of our future work; we believe that the envisionment can improve the efficiency of our version of this analysis. It is an open question what classes of non-linear equations can be analyzed in this manner.

Other researchers have explored the use of PRISM [9] to perform verification of cyber-physical systems. PRISM models have the advantage that they can consider probabilistic state transitions. Probabilistic state transitions make PRISM particularly useful for verifying requirements about the likely reliability of systems given failure rates of components (e.g, "what is the probability that vehicle will be able to operate continuously for 570 hours"). A challenge for doing this analysis is that there is no automatic way to move from equations specifying components to the models used by PRISM.

As we have shown in this paper, even when we know all the models and values, QRM can help verify requirements very quickly. Almost all formal verification tools are very general and their performance scales very poorly with number of variables or components. For more complex systems, QRM can verify requirements when formal methods cannot. QRM has the advantage of algorithms specifically tuned to continuous systems developed over decades in the AI community.

DISCUSSION

We have presented our QRM approach for early design verification using qualitative simulation. In particular, we have illustrated how envisionments can verify requirements and guide detailed design by identifying implicit requirements. Furthermore, we have shown that qualitative verification using QRM is able to eliminate large areas of the intractable search space of design from components.

Our initial explorations have opened a number of promising directions for future work. As stated earlier, automatically incorporating available quantitative information about parameters would allow us to verify a large set of requirements. We intend to build on existing work on semi-quantitative simulation [11]. Another important future direction concerns the interaction between design space exploration and qualitative simulation. In the case of design flaws, QRM could use the envisionment to produce diagnoses guiding topological search. In the case of potentially successful designs, the envisionment could provide guards, or inequalities, to guide parameter selection.

ACKNOWLEDGMENTS

Peter Bunus (Linkoping) for helping us understand the nuances of Modelica models. This work was sponsored by The Defense Advanced Research Agency (DARPA) Tactical Technology Office (TTO) under the META program and is Approved for Public Release, Distribution Unlimited. The views and conclusions in this document are those of the authors and should not be interpreted as representing the official policies, either expressly or implied, of the Defense Advanced Research Projects Agency or the U.S. Government.

REFERENCES

[1] de Kleer, J. and Williams, B.C. 1992. Special volume on Qualitative Reasoning about Physical Systems II, *Artificial Intelligence*. Elsevier.

[2] Forbus, K. 1984. Qualitative process theory. *Artificial Intelligence*, 24. Elsevier 85-168.

[3] Kuipers, B. 1994. *Qualitative Reasoning: Modeling and Simulation with Incomplete Knowledge*. Cambridge, MA: MIT Press.

[4] Williams, B. 1984. The Use of Continuity in Qualitative Physics. In *Proceedings of the National Conference on Artificial Intelligence*, Austin, TX, pp. 350-354.

[5] Fritzson, P. 2004. *Principles of Object Oriented Modeling and Simulation With Modelica 2.1*. Piscataway, NJ: IEEE Press.

[6] Say, C. and Akin, H. 2003. Sound and complete qualitative simulation is impossible, *Artificial Intelligence*, v.149 n.2, p.251-266.

[7] Konur, S. 2010. A survey on temporal logics. *CoRR*, abs/1005.3199.

[8] Hinrichs, T. R.; Forbus, K. D.; de Kleer, J.; Yoon, S.; Jones, E.; Hyland, R.; and Wilson, J. 2011. Hybrid qualitative simulation of military operations. In *Proceedings of Innovative Applications of Artificial Intelligence*. San Francisco.

[9] Kwiatkowska, M., Norman, G., and Parker, D. 2011. PRISM 4.0: Verification of Probabilistic Real-time Systems. In *Proceedings of the 23rd International Conference on Computer Aided Verification (CAV'11)*, volume 6806 of LNCS, pages 585-591, Springer.

[10] Tiwari, A. 2008. Abstractions for hybrid systems. *Formal Methods in Systems Design*, 32:57–83,

[11] Berleant, D. and Kuipers, B. 1997. Qualitative and quantitative simulation: bridging the gap. *Artificial Intelligence* 95(2): 215-255.