Challenges for qualitative reasoning for engineering design

Daniel G. Bobrow and **Matthew Klenk** and **Johan de Kleer** and **Bill Janssen** and **John Hanley**
Palo Alto Research Center
3333 Coyote Hill Road, Palo Alto, CA 94304 USA
dekleer@parc.com

Abstract

We believe that qualitative reasoning (QR) is central to how engineers do engineering design. However, QR is not being used by practicing engineers. DARPA is planning to make improved tools for engineering design available to a wide community. To have QR included in this toolset, we need to integrate QR into a widely available tool chain. Modelica, with its open source implementations, is a good candidate for such integration. This short paper outlines how we have started this integration with OpenModelica and some of the challenges that must be met to make QR more broadly useful to the design process. The core challenge is to find tasks that QR can do that engineers want/need to be done, and are not done by existing tools.

Introduction

We believe (Weld and de Kleer 1989) qualitative reasoning is the fundamental basis upon which engineers reason about physical systems. Qualitative reasoning plays a key role in every facet of designing a system ranging from early stage design (Kurtoglu and Tumer 2007) through understanding of simulation results, to planning how designs need to be modified to meet requirements. Unfortunately, none of the commonly used design/analysis tools provide computational QR support for these tasks. Leaving qualitative reasoning entirely to the human engineer risks missing critical inferences. Our vision is to create a design tool chain in which qualitative reasoning supports every segment of the life cycle of a product.

The DARPA AVM program plans to provide open source tools for crowd sourcing designs for miliitary vehicles. One of these tools is OpenModelica (OM), which supports the widely used design language Modelica(Fritzson 2004). Modelica is an object oriented-language, with a general class structure, and a powerful inheritance framework. The Modelica association has released a large standard library of reusable models across multiple domains (e.g. electrical, mechanical, fluids, thermodynamics).

Behavior of models in Modelica is primarily based on equations. These include ordinary differential equations and algebraic constraints among model variables. Such sets of Differential Algebraic Equations (DAE's) provides acausal modeling, and a declarative semantics. This supports more extensive reuse of models.

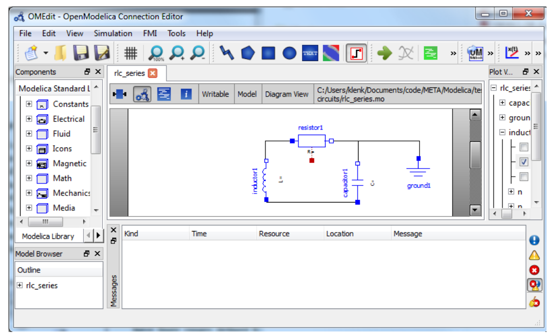

Figure 1: OMEdit, graphical editor for Modelica models

Modelica provides a facility for hybrid simulation, where at events determined by inequalities, discrete variables can change value, and continuous functions can make non-continuous jumps. OpenModelica links to a mathematical engine that can simulate a fully specified composite model to determine its behavior over time, given a driving use case and initial parameters. Composite models are created by connecting component models through specified ports.

The features of Modelica described above are all directly mappable into our Qualitative Modeling Language (QML). The real domains and ranges of the DAE's are mapped into an ordered space consisting of landmark values, and the intervals between those values. In the simplest version of the mapping, the only landmarks are 0, and plus-infinity and minus-infinity. The DAE's are turned into constraints on the mapped values. While Modelica simulates a single path determined by the numeric values of parameters, our simulation engine (QRM) creates an envisionment of all possible paths, given the mapping of initial conditions into intervals of the qualitative domain.

What follows is a description of our approach for integrating QRM into the OpenModelica tool. After which we present two sets of design tasks we think could be addressed by qualitative reasoning: (1) engineering challenges, which include tasks addressed by QR and must be integrated into design tools, and (2) research challenges, which are design tasks that we believe are problems ripe for QR solutions.

Integration of QML/QRM into the design process

Modelica's popularity in industry and academia has resulted in numerous open source tools for design construction and simulation (OpenModelica and JModelica). While these tools are still immature, they have been created around the design process. For example, there is a visual user interface (OMEdit) for constructing composite models by instantiating model components from the library, and connecting them with "wires" on the screen. Successfully integrating qualitative reasoning with such tools could be a boon for making automated qualitative analysis available to designers.

Although a graphical interface (e.g., OMEdit, shown in Figure 1) is useful for specifying designs, the Modelica also includes a model construction language as part of the models. The OpenModelica compiler interprets this language to produce a composite model with a simplified set of DAEs suitable for numeric simulation. We found that we could import these DAEs and initial conditions into QML and perform qualitative simulation.

Designs are tested against requirements. To integrate requirements into this process, we translate requirements expressed in English into Linear Temporal Logic (LTL) (Emerson 1995). These requirements can be used in the numeric simulation and the corresponding qualitative requirements can be used by QRM.

Where are the gaps in qualitative reasoning for engineering design?

Most research in the qualitative reasoning community has focused on isolated pieces of the engineering design task. The purpose of this paper is to consider from the point of view of the designer, where QR could help, as well as, challenge the QR community to extend their methods to better address the needs of designers.

Engineering Challenges

The first set of challenges concern design tasks that we believe would benefit from current QR techniques. The challenge is integrating these techniques into tools used by designers.

Design Space Exploration One aspect of design space exploration is topological design (i.e., determining the configuration of abstract components). Qualitative reasoning has the ability to rule out topologies of components with underspecified parameters with respect to specified requirements. Therefore, qualitative reasoning can be used with a computational topological design exploration tool to automatically select promising designs for parameter selection.

Interpreting Results of Numeric Simulation The guaranteed coverage theorem (Kuipers 1994) enables qualitative reasoning to be useful in interpreting the results of numeric simulation. For example, while a stiff system may lead to numeric instability (shown in Figure 2), there is no trajectory corresponding to this simulation in the envisonment because the changes resulting from the numeric instability are not seen in any qualitative trajectory. Therefore, we can identify the flawed numeric simulation. Another way qualitative reasoning may be used to interpret numeric results is to identify phenomena at different time scales. Consider a numeric simulation appears to simply be reaching an asymptote, but given the envisionment, we know this is an oscillatory system.

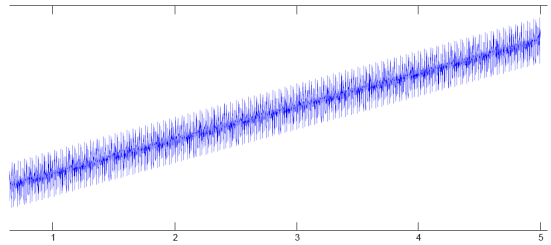

Figure 2: Numeric simulation with instability

Parameter exploration for fault finding/removal Given a fully specified design, we are able to trace a trajectory through the envisionment. If this design did not meet requirements, we would like to identify nearby trajectories that may meet requirements. If this design does meet requirements, we would like to identify nearby trajectories that may result in failures. We can identify nearby trajectories by performing comparative analysis (Weld 1989) or by adding constraints between parameters which must be satisfied by a trajectory. One class of constraints we have explored is ones that create a total ordering among landmarks.

Focus PCC analysis on important paths Probabilistic Certificate of Correctness (PCC) is an analysis of a fully specified design whose parameters are specified by distributions. The standard approach involves sampling the parameter space and performing simulations. Extremely rare events are either ignored, or poorly estimated due to limit samples. Given an envisionment, we can identify which paths lead to these failures and also perform selective re-sampling to better estimate their probabilities.

Implicit Requirement Discovery Designers frequently create derived requirements which encapsulate assumptions about their design. For example, a piston-crank system will have two kinematic singularities which control engineers will need to avoid in detailed design. The envisionment of such a system would identify such states thereby automating this process usually performed by teams of engineers

Research Challenges

The second set of challenges concern design tasks that we believe that pose interesting problems for the qualitative reasoning community.

Reasoning about architectural (functional) design, and knowledgeable mapping to topologies While there has been some efforts integrating functional and qualitative reasoning (Everett 1999)(Bell *et al.* 2005)(Wetzel and Forbus

2012), there are still challenges to making this useful for designers. One reason for this is likely the lack of agreement in the functional modeling community with respect to the content and structure of functional models (Erden *et al.* 2008). Another issue is that commercial design tools are inadequate for functional modeling, and the ones that allow functional modeling do not support tying these models to topologies or simulations.

Explore tradeoff space among different requirements A major problem facing designers and procurement officers concerns the requirements for the design. Understanding the tradeoffs between requirements and when the requirements are inconsistent could be a major cost savings in designing new products. Qualitative models of the requirements could be used to identify networks of competing requirements while eliciting a rank ordering of importance amongst requirements early in the design process.

Explore the space of use case scenarios Moving from requirements to testable use cases during design, simulation and prototyping can be incredibly difficult. Consider the requirement, "The vehicle shall be designed to operate for a minimum of 25 years(T), 30 years(O), corrosion service life, which will include varying or extended periods in corrosive environments...". Envisionment, by generating all possible outcomes, is well suited to identifying problematic cases.

Compensatory Design Much of the work in engineering design involves compensating for non-ideal components. Consider the grooves in sidewalks which allow the concrete to expand and contract due to temperature changes without cracking. Qualitative reasoning could be used to identify potential topology changes to compensate for known errors in the models.

Compare two designs wrt requirements As design progresses, engineers will frequently look at competing designs and make judgments about which is better. Supporting this task with qualitative reasoning would require new ideas from qualitative reasoning perhaps including tighter integration with quantitative reasoning.

Discussion

In this paper, we reported on the challenges that must be overcome to incorporate automated qualitative reasoning into the design process. We have grouped these challenges into two categories: (1) engineering challenges, which we believe that QR is already providing techniques that benefit design and need only be integrated into design tools, and (2) research challenges, which we are design tasks that the QR community could be well-positioned to address by expanding the field.

Acknowledgements

We thank Peter Struss for discussions concerning some of these challenges. This work was sponsored by The Defense Advanced Research Projects 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 DARPA or the US government.

References

Jonathan Bell, Neal Snooke, and Chris Price. A language for functional interpretation of model based simulation. In *Pages 15471548 of: Proceedings of international joint conference on artificial intelligence (IJCAI*, 2005.

E. Allen Emerson. Temporal and modal logic. In *HANDBOOK OF THEORETICAL COMPUTER SCIENCE*, pages 995–1072. Elsevier, 1995.

M. s. Erden, H. Komoto, T. j. Van beek, V. D'amelio, E. Echavarria, and T. Tomiyama. A review of function modeling: Approaches and applications. *Artif. Intell. Eng. Des. Anal. Manuf.*, 22(2):147–169, January 2008.

John Otis Everett. Topological inference of teleology: Deriving function from structure via evidential reasoning. *Artificial Intelligence*, 113:149–202, 1999.

P. Fritzson. *Principles of Object-Oriented Modeling and Simulation with Modelica 2.1*. Wiley-IEEE Press, Piscataway, NJ, 2004.

Benjamin Kuipers. *Qualitative reasoning: modeling and simulation with incomplete knowledge*. MIT Press, Cambridge, MA, USA, 1994.

Tolga Kurtoglu and Irem Y. Tumer. Ffip: A framework for early assessment of functional failures in complex systems. In *International Conference on Engineering DesignICED07*, 2007.

D.S. Weld and J. de Kleer. *Readings in Qualitative Reasoning about Physical Systems*. Morgan Kaufmann, 1989.

D. S. Weld. Automated model switching: Discrepancy driven selection of approximation reformulations. Technical Report 89-08-01, Department of Computer Science and Engineering, FR-35, University of Washington, Seattle, WA 98195, October 1989.

Jon Wetzel and Kenneth Forbus. Teleological representations for multimodal design explanations. In *Proceedings of International Workshop on Qualitative Reasoning*, 2012.