Ajitha Rajan et al.: Assessing Requirements Quality Through Requirements Coverage, Proceedings of The Sixth NASA Langley Formal Methods Workshop, p.12–14

Assessing Requirements Quality Through Requirements Coverage*

Ajitha Rajan[1], *Mats Heimdahl*[1], *Kurt Woodham*[2]

[1] *University of Minnesota*
[2] *L-3 Communications*
arajan@cs.umn.edu, heimdahl@cs.umn.edu, kurt.woodham@l-3com.com

Extended Abstract

In model-based development, the development effort is centered around a formal description of the proposed software system—the "model". This model is derived from some high-level requirements describing the expected behavior of the software. For validation and verification purposes, this model can then be subjected to various types of analysis, for example, completeness and consistency analysis [6], model checking [3], theorem proving [1], and test-case generation [4, 7]. This development paradigm is making rapid inroads in certain industries, e.g., automotive, avionics, space applications, and medical technology. This shift towards model-based development naturally leads to changes in the verification and validation (V&V) process. The *model validation* problem—determining that the model accurately captures the customers' high-level requirements—has received little attention and the sufficiency of the validation activities has been largely determined through ad-hoc methods. Since the model serves as the central artifact, its correctness with respect to the users' needs is absolutely crucial. In our investigation, we attempt to answer the following two questions with respect to validation (1) *Are the requirements sufficiently defined for the system?* and (2) *How well does the model implement the behaviors specified by the requirements?* The second question can be addressed using formal verification. Nevertheless, the size and complexity of many industrial systems make formal verification infeasible even if we have a formal model and formalized requirements. Thus, presently, there is no objective way of answering these two questions. To this end, we propose an approach based on testing that—when given a set of formal requirements—explores the relationship between *requirements-based* structural test-adequacy coverage and *model-based* structural test-adequacy coverage.

The proposed technique uses requirements coverage metrics defined in [9] on formal high-level software requirements and existing model coverage metrics such as the Modified Condition and Decision Coverage (MC/DC) used when testing highly critical software in the avionics industry [8]. Our work is related to Chockler et al. [2], but we base our work on traditional testing techniques as opposed to verification techniques.

To objectively assess whether the high-level requirements have been sufficiently defined for the system, we produce a set of test cases that achieve a certain level of structural coverage of the high-level requirements, and then measure coverage achieved by the test suite over the model. If a test suite provides high requirements coverage but yields poor coverage of a model, it may be

* This work has been partially supported by NASA Ames Research Center Cooperative Agreement NNA06CB21A, NASA IV&V Facility Contract NNG-05CB16C, and the L-3 Titan Group.

due to one or more of the following: (a) there are missing or implicit requirements, (b) there is behavior in the model that is not derived from the requirements, or (c) the set of tests derived from the requirements was inadequate. On the other hand, to objectively assess how well the model implements the behaviors specified in the requirements, we generate a set of test cases that achieve structural coverage of the model, and then measure requirements coverage achieved. Poor requirements coverage is an indicator of either (a) the model does not adequately implement the behaviors specified in the requirements, or (b) the model is correct and the requirements are poorly written.

To illustrate the technique, we use a rigorous requirements coverage metric *Unique First Cause* (UFC) coverage defined in over requirements formalized as Linear Temporal Logic (LTL) properties [9]. We use the Modified Condition/Decision Coverage (MC/DC) criterion [5] to measure structural coverage over the model. In a preliminary study, we use five industrial case examples from the civil avionics domain. For each of these systems, we perform two kinds of assessment—(1) generate test suites to provide UFC coverage over the requirements and measure MC/DC achieved over the model, and (2) generate test suites to provide MC/DC over the model and measure UFC coverage achieved over the formal requirements. We analyze the relationship between requirements coverage and model coverage to make an assessment of the quality of the sets of requirements as well as the models.

On three of the five case examples, test suites generated to provide UFC coverage of the requirements provided reasonably good MC/DC of the models. This indicates that for these case examples, the requirements are well defined. Nevertheless, the test suites provided 10%-20% less than achievable MC/DC over the models. This is somewhat expected since the requirements (representing DO-178B high-level requirements) are typically less detailed than the model (representing DO-178B low-level requirements). Another reason may be that the UFC metric used for requirements coverage is not sufficiently rigorous and we thus have an inadequate set of requirements-based tests. On the remaining two case examples, test suites providing requirements UFC coverage gave very poor MC/DC on the model. Closer investigation revealed that on one example, there were many missing requirements. In the final case example, the requirements were good, however, their structure was so that the complexity of conditions in the requirements were hidden. For such requirements, the UFC metric that we use is not effective since the structure of the formalized requirements effectively "cheated" the UFC metric. One solution to this would be to restructure the requirements to reveal condition complexity. Another possible solution is to use a requirements coverage metric that is not as sensitive to the structure of the requirements. We hope to investigate this issue further in our future work.

We found that on all but one of the industrial systems, test suites providing MC/DC over the model achieved close to achievable requirements UFC coverage. This implies that the model exercises almost all the behaviors specified by the requirements for these systems. Nevertheless, on one model the MC/DC test suites did poorly, only achieving 30% of the achievable requirements coverage. This may either be because the model does not implement all the behaviors or the MC/DC metric is not rigorous enough. At this time we have not been able to determine the cause more closely, but we hope to do so in our future work.

To summarize, we found that analyzing the relationship between requirements coverage and model coverage provides a promising means of assessing requirements quality. Nevertheless, the effectiveness of this approach is highly dependent on the rigor and effectiveness of the coverage metrics used, and awareness of the pitfalls of structural coverage metrics is essential. For instance,

in this experiment we found that the UFC metric was surprisingly sensitive to the structure of the requirements, and one has to ensure that the requirements structure does not hide the complexity of conditions for the metric to be effective.

References

[1] S. Bensalem, P. Caspi, C. Parent-Vigouroux, and C. Dumas. A methodology for proving control systems with Lustre and PVS. In *Proceedings of the Seventh Working Conference on Dependable Computing for Critical Applications (DCCA 7)*, pages 89–107, San Jose, CA, January 1999. IEEE Computer Society.

[2] H. Chockler, O. Kupferman, and M. Y. Vardi. Coverage metrics for formal verification. In *12th Advanced Research Working Conference on Correct Hardware Design and Verification Methods, volume 2860 of Lecture Notes in Computer Science*, pages 111–125. Springer-Verlag, October 2003.

[3] Edmund M. Clarke, Orna Grumberg, and Doron Peled. *Model Checking*. MIT Press, 1999.

[4] Angelo Gargantini and Constance Heitmeyer. Using model checking to generate tests from requirements specifications. *Software Engineering Notes*, 24(6):146–162, November 1999.

[5] K.J. Hayhurst, D.S. Veerhusen, and L.K. Rierson. A practical tutorial on modified condition/decision coverage. Technical Report TM-2001-210876, NASA, 2001.

[6] Mats P. E. Heimdahl and Nancy G. Leveson. Completeness and consistency in hierarchical state-base requirements. *IEEE Transactions on Software Engineering*, 22(6):363–377, June 1996.

[7] A. Jefferson Offutt, Yiwie Xiong, and Shaoying Liu. Criteria for generating specification-based tests. In *Proceedings of the Fifth IEEE International Conference on Engineering of Complex Computer Systems (ICECCS '99)*, October 1999.

[8] RTCA. *DO-178B: Software Considerations In Airborne Systems and Equipment Certification*. RTCA, 1992.

[9] Michael Whalen, Ajitha Rajan, Mats Heimdahl, and Steven Miller. Coverage metrics for requirements-based testing. In *Proceedings of International Symposium on Software Testing and Analysis*, July 2006.