



# **Logic Design Tool & T-VEC Pairing Evaluation**

**Document TBV-E-0001**

**Version 1.1**

Prepared By: Steve Morton, Software DER

Date: February 8, 2015

*TBV Associates* LLC

22892 Kissinger Road  
Leavenworth, Kansas 66048

### **Revision History**

| <b>Revision</b> | <b>Author</b> | <b>Date</b> | <b>Notes</b>                      |
|-----------------|---------------|-------------|-----------------------------------|
| 1.0             | SD Morton     | 02/04/2015  | Initial creation                  |
| 1.1             | SD Morton     | 02/12/2015  | Minor revision and clarification. |
|                 |               |             |                                   |

---

## **Table of Contents**

|       |   |   |
|-------|---|---|
| 1     | Introduction.....   | 4 |
| 1.1   | Scope.....  | 4 |
| 1.2   | About the Author.....                                     | 4 |
| 1.3   | Acronyms .....  | 4 |
| 1.4   | Referenced Documents .....                                | 4 |
| 2     | Logic Design Tool Coupling with T-Vec.....                | 4 |
| 3     | Formal Methods vs Testing under DO-178C.....              | 5 |
| 4     | Tool Qualification Considerations .....                   | 5 |
| 5     | Recommendations .....                                     | 6 |
| 5.1   | Technical Evaluative Experiment.....                      | 6 |
| 5.1.1 | LDT Screening Effectiveness Experiment .....              | 6 |
| 5.1.2 | LDT Requirements Correction Effectiveness Experiment..... | 6 |

# 1 Introduction

TBV Associates was contracted by Dave McFarland to evaluate the potential value of coupling the Logic Design Tool (LDT) with the T-VEC test vector generation tool by T-VEC Technologies. This brief technical paper is the result of the requested evaluation.

## 1.1 Scope

The technical evaluation presented herein is not based on exercise of the tools described, but is rather an overview of the technical aspects of the tools and their pairing within the aerospace software market that is governed under RTCA/DO-178C Software Considerations in Airborne Systems and Equipment Certification.

Also included in this analysis are the subjects of Formal Methods (under RTCA/DO-333 Formal Methods Supplement to DO-178C and DO-278A) and Tool Qualification (under RTCA/DO-330 Software Tool Qualification Considerations). Some elements of this evaluation are considered in light of the definitions for types of models presented in RTCA/DO-331 Model Based Development and Verification Supplement to DO-178C and DO-278A.

## 1.2 About the Author

Steve Morton is a software DER with more than 20 years' experience in the development and certification of airborne software. Steve has an extensive background in software tools, including roles as a tool developer, user, and as the certification authority accepting the tool qualification data for various tools. Steve was a leading member of the RTCA special committee SC-205, which authored RTCA/DO-178C, RTCA/DO-330, and RTCA/DO-333.

## 1.3 Acronyms

The following acronyms are used throughout this document:

DER.....Designated Engineering Representative

LDT .....Logic Design Tool

LLR .....Low-Level Requirement

## 1.4 Referenced Documents

- |     |              |  |
|-----|--------------|--|
| [1] | RTCA/DO-178C | Software Considerations in Airborne Systems and Equipment Certification; RTCA, Inc.; December 13, 2011.    |
| [2] | RTCA/DO-330  | Software Tool Qualification Considerations; RTCA, Inc.; December 13, 2011.                                 |
| [3] | RTCA/DO-331  | Model Based Development and Verification Supplement to DO-178C and DO-278A; RTCA, Inc.; December 13, 2011. |
| [4] | RTCA/DO-333  | Formal Methods Supplement to DO-178C and DO-278A; RTCA, Inc.; December 13, 2011.                           |

# 2 Logic Design Tool Coupling with T-VEC

LDT and T-VEC in this pairing are used differently. That is, LDT is employed as an analytical tool which looks at the *design model* [3] of the software under development. LDT identifies inconsistencies and ambiguities in the logic represented by the model – that is the *requirements represented by the model* [3] are analyzed for correction by the tool.

T-VEC, similarly, analyzes the design model, but instead of producing analytical output, T-VEC is used to create full-path test vectors for exercising the model's implementation. By using a test-based

approach, T-VEC more closely matches the preferred functional verification style in RTCA/DO-178C section 6.

The two methodologies are complementary. By cleaning up inconsistencies and ambiguities in the design model, LDT helps to produce a better model-based statement of the low-level requirements (LLRs) [1] for the software. By creating LLR based test vectors which may be used to verify the implementation's functionality, T-VEC helps to automatically satisfy some of the verification objectives in RTCA/DO-178C.

As such, the relationship of the two tools is similar to a word processor program to the document produced by the program. Microsoft Word, for example, contains numerous tools and capabilities to assist the writer in creating a better document, from spell checking to grammar analysis. However, the actual quality of the produced document can only be influenced by those Word features. The balance of the quality factor is whether the words, as arranged, contain all of the information required to convey the intended, unambiguous meaning intended by the author. It is common for technical documents to be reviewed to evaluate whether the document is complete, clear, and unambiguous. This review is instrumental in determining the success of the author in using Word to create a valid arrangement of linguistic fragments that, together, convey the correct meaning to the reader.

LDT functions in a manner like the analytical tools within Word. LDT may be used to refine the arrangement of logic requirements in the design model, but LDT cannot determine if the arrangement necessarily meets the higher level requirements from which the design model was developed [3].

T-VEC, on the other hand, serves in the same role as the document review – helping to evaluate not only the technical correctness of the design model, but also the ability of the implementation to meet the higher level requirements from which the design model was developed.

### 3 Formal Methods vs Testing under DO-178C

RTCA/DO-333 introduces the idea that mathematical analysis may be used to replace or supplant some of the testing required by RTCA/DO-178C. A formal method is defined as *[d]escriptive notations and analytical methods used to construct, develop and reason about mathematical models of system behavior* [4]. Using formal methods may allow an applicant to substitute analysis for some elements of the testing required under RTCA/DO-178C.

It is unclear if LDT would be considered a formal methods tool under this definition. It may be worthwhile exploring this avenue of inquiry, if the specific analyses performed by LDT could be used to obviate the need for some aspects of testing, such as structural coverage analysis, for the analyzed software.

### 4 Tool Qualification Considerations

Tool qualification is required if processes from RTCA/DO-178C are *eliminated, reduced, or automated by the use of [the tool] without its output being verified as specified in section 6 [of RTCA/DO-178C]* [1]. That is, tool qualification is used to establish trust in the function(s) performed by the tool so that the output of the tool no longer requires separate verification.

If LDT screened design models are subjected to T-VEC based testing, LDT may not be suitable for qualification, as the tool's output is verified as specified in RTCA/DO-178C section 6. If LDT is considered a formal methods tool, qualification of the analysis function may be valuable, as it could allow an applicant to eliminate some of the testing steps otherwise required.

LDT and T-VEC employ different technologies, which are complementary. By making transition between the two tools easy, a value as significant as qualifying LDT is introduced. LDT qualification would have more value when not paired with T-VEC, although the exact level of claimable objective satisfaction is not yet established.

Before a more full analysis of the benefit of pursuing tool qualification can be made, it will have to be determined if LDT meets the definition of a formal methods tool. If so, the tool would be subject to a TQL-5 qualification [1], which is essentially the presentation of user-perspective operational

requirements and demonstration (by testing the tool) that those Tool Operational Requirements are met.

## 5 Recommendations

Pairing LDT with the dissimilar approach used by T-VEC adds value to the tool chain. The author has been unable to identify any other commercial tools which provide similar test generation capabilities as T-VEC. One candidate, the BEACON Automatic Unit Test Tool, appears to have been withdrawn from the market and is no longer available.

The coupling of analysis with subsequent test fits the RTCA/DO-178C paradigm well, and may be valuable in the marketing of the tool. A specific market-based analysis of this value is beyond the scope of this evaluation, and is beyond the experience of the author.

However, it is recommended that a technical evaluative measurement of LDT's efficacy in finding and eliminating specification or implementation errors may be undertaken using the LDT to T-VEC pairing.

### 5.1 Technical Evaluative Experiment

The experiment to measure LDT's effectiveness can be carried out along two lines:

1. Pass both the original and LDT screened models through T-VEC to show the degree by which LDT identifies and helps eliminate errors and inconsistencies. The LDT screened models should exhibit significantly lower levels of T-VEC identified issues.
2. Quantify the effect of using LDT on the requirements represented by the model by showing how the requirements from which the model is generated can be fixed using the LDT improvements in the model.

Neither of these approaches offers a clear numerical value to the use of LDT, but may be translated by a prospective customer by their own metrics into estimating the value of adding LDT to their process. It is known that the earlier an issue can be identified and resolved, the lower the cost of resolving the issue – in many cases by a logarithmic scale.

#### 5.1.1 LDT Screening Effectiveness Experiment

1. Take one or more models and pass them through T-VEC, measuring structural coverage and the number of errors identified.
2. Pass those same models through LDT, then put the corrected models through T-VEC, measuring structural coverage and the number of errors identified.
3. Statistically analyze the two results to show the efficacy of LDT, and hence its value in finding and eliminating errors and issues earlier in the process than is possible in a purely test-based paradigm.

#### 5.1.2 LDT Requirements Correction Effectiveness Experiment

For this experiment, both the model on which LDT is to be run and the requirements from which the model was developed must be available.

1. Run the model through LDT.
2. Translate any model corrections into the corresponding corrections in the requirements from which the model was developed.
3. If possible, identify guidelines to ensure correct translation of model fixes into requirements corrections.
4. Analyze the efficacy of LDT in providing feedback corrections to the requirements from which the model was developed.