

# An Iterative Empirical Strategy for the Systematic Selection of a Combination of Verification and Validation Technologies

Margaret A. Wojcicki and Paul Strooper  
*School of Information Technology and Electrical Engineering*  
*University of Queensland, Queensland 4072*  
*{wojcicki, pstroop}@itee.uq.edu.au*

## Abstract

*The development of verification and validation (V&V) technologies is rapid and produces a multitude of V&V technologies; it is therefore difficult for practitioners to select appropriate V&V technologies. Since most V&V technologies will be combined with existing or other novel V&V technologies, it is important to be aware of how the technologies should be combined (for example, the order of application) and the cost-effectiveness of these combinations. This paper presents a strategy for selecting and evaluating particular V&V combinations that focuses on maximising completeness and minimising effort, and involves metrics such as effort and defect-detection effectiveness. This strategy for combining V&V technologies can benefit from empirical evidence. The paper presents a systematic approach for applying empirical information regarding the costs and capabilities of V&V technologies for the selection of cost-effective combinations of such technologies.*

## 1. Introduction

Verification is concerned with checking that a computer program conforms to its specification, whilst validation is concerned with checking that it meets the requirements of the client; together they are often referred to as V&V. The problems that V&V technologies can detect in programs can be referred to as defects, faults, bugs (erroneous patterns of code) or failures (occurrences of incorrect program behaviour which is the result of a fault in code); this paper will refer to all these problems as defects.

Studies comparing V&V techniques date back to the 1970s [4, 12, 14, 23, 36] and their overriding message is that a combination of testing techniques should outperform applying these techniques individually.

The benefits of combinations of V&V technologies (specifically structural and functional testing with code inspection) have been evaluated in Selby's [26] landmark study, which was replicated by Wood et al. [36]. More recently the combination of V&V technologies has been investigated with static analysis and model checking [22], inspection and testing [16], as well as automated static analysis and code inspection in the context of concurrent components [34].

Enders and Rombach [8] presented a law (Hetzel-Myers law) regarding V&V technologies that states that "a combination of different V&V methods outperforms any single methods alone", and standards such as the ISO/IEC 61508 (Functional safety of electrical/electronic/ programmable electronic safety-related systems) mandate that a combination of V&V technologies is required for verification, but no work has stated how these technologies should be combined and moreover how one can determine what combinations of V&V technologies are most cost-effective in a particular context. Most practitioners will apply V&V technologies in combination, but with a large and increasing number of V&V technologies to choose from it has become overwhelming to decide what combination of V&V technologies should be used.

This paper presents a systematic strategy for selecting and evaluating particular V&V combinations that focuses on maximising completeness and minimising effort, and examines metrics such as effort and defect-detection effectiveness. The strategy is iterative and brings together decision support and empirical information [25] by using data collected after the application of a combination of V&V technologies to determine whether or not the combination performed as expected and whether or not adjustments should be made. The strategy will be evaluated in future work in the application area of concurrent programs. An initial

application of the first and part of the second step of the combination strategy is presented in this paper.

A concurrent program consists of two or more processes that cooperate in performing a task [1]. Communication between the two processes is facilitated by shared variables or message passing. Concurrent programs are often non-deterministic in that they may return different outputs for the same inputs and as a result they are more complex than sequential programs. Concurrent programs are also prone to specific defects such as interference and deadlock [20]. Interference is an interleaving of threads that results in incorrect updates to the state of a shared object. Deadlock is a situation in which there are no eligible actions to be performed by the program. V&V in the context of concurrency deals with these complexities and is quite diverse.

The TestCon method is used to detect concurrency defects in concurrent Java components [19]; it combines code inspection, the automated static analysis tools FindBugs [13] and Jlint [2], and dynamic analysis using ConAn [18]. FindBugs and Jlint inspect Java bytecode for occurrences of bug patterns (sections of code that appear to have a defect). Both tools can report several concurrent bug patterns; in TestCon FindBugs is used to detect defects related to interference and Jlint is used to detect deadlock cycles. Empirical analysis of the TestCon method has shown that the combination of code inspection and automated static analysis may not be sufficient for deadlock detection [24, 34]. Therefore the method may need to be modified with the use of the combination strategy to produce a more cost-effective V&V technology combination, considering novel technologies such as the Java PathFinder model-checker [11].

In Section 2, we introduce the combination strategy. In Section 3, we discuss the advantages and drawbacks of the combination strategy. In Section 4 we review the related work. Section 5 describes a preliminary evaluation of the combination strategy in the context of the TestCon method which will be continued in future work, whilst Section 6 provides concluding comments.

## 2. The combination strategy

The combination strategy depicted in the flow chart of Figure 1 consists of four steps: pre-selection gathering of cost-effectiveness data, two arguments to maximise completeness and minimise effort, followed by application protocol and updating cost-effectiveness data. Cost-effectiveness data is stored in a V&V matrix; there are three instances of this matrix in the strategy denoted M1 (the skeleton of the matrix), M2

(populated matrix), and M3 (relative data matrix). The steps are described in detail below using the labels from Figure 1.

Step 1 - Pre-selection: gathering cost-effectiveness information

(a) To determine a cost-effective combination of V&V technologies one needs to know what defects will need to be detected. This information may depend on the application area or it may be influenced by the results of prior defect detection.

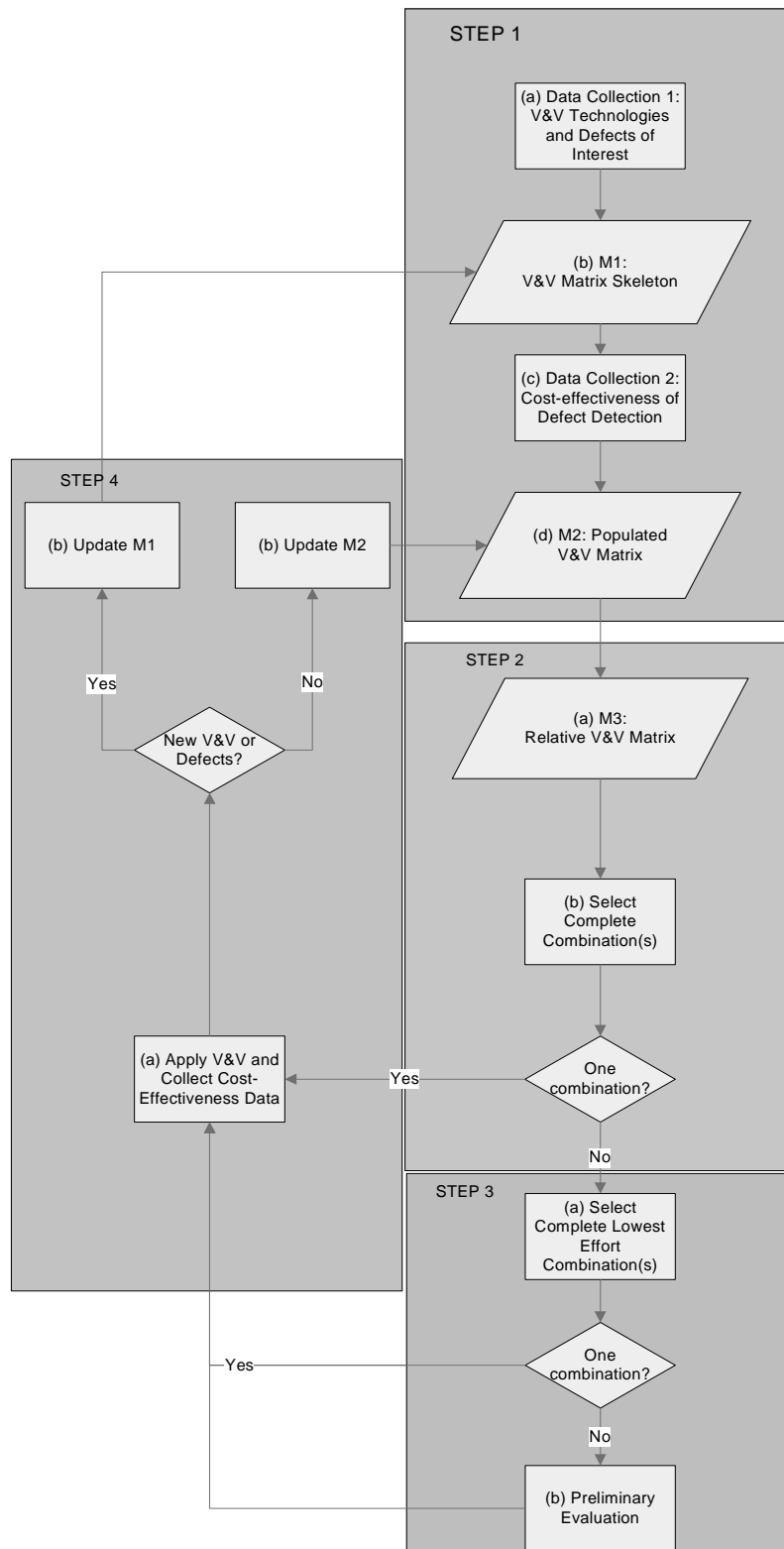
Once the defect types of interest are determined (these can be particular defects but we will interpret a defect as a very particular defect type), information needs to be collected on the V&V technologies available for detecting those types of defect.

(b) The skeleton of a V&V matrix (M1) is then created to hold data on the cost-effectiveness of defect detection of the available V&V technologies. The V&V technologies populate the left-most column of the matrix whilst the defect types populate the top row.

(c) Cost-effectiveness data is collected on the V&V technologies to populate the matrix; this data includes the defect-detection effectiveness and the effort required to apply the V&V technologies. If no information on a V&V technology's effectiveness or effort data is available, the appropriate cell of the matrix should be filled with a question mark. When selecting combinations of V&V technologies in later steps, if there are question marks it will be difficult to make a decision.

(d) The matrix M1 is populated with V&V technology information in the form X [Y] where X is the effort data and Y is the effectiveness data. People-hours needed to apply the V&V technology can be used as a metric for the former aspect whilst percentage of defect type detected by the V&V technology can be used for the latter aspect. People-hours may not be a sensible measure for the effort for all different V&V technologies; for instance, the effort for code inspection is critically dependent on the size of the code. To make the data contained in the matrix useful, the practitioner needs to record what the data means to make a properly informed decision.

Moreover comparing some V&V technologies using the same metrics may not be appropriate; for example comparing inspection vs a fully automated tool based on people-hours spent applying these technologies may be inaccurate, because the tool will take much less time to execute compared to inspection. The time required to execute the tool may also not reflect all the effort



**Figure 1. The combination strategy for the selection of cost-effective combinations of V&V technologies; it consists of 4 steps depicted with 4 boxes.**

required; costs such as setup and the cost of finding the defect in the program (which would be lower when applying inspection vs testing) may need to be considered. Section 3 further elaborates on the aspects of cost that may be included in the strategy.

An example of a populated V&V matrix M2 is shown in Table 1. If previous data exists on the cost-effectiveness of combinations of V&V technologies, this data can also be added to the matrix in a separate row for these combinations.

**Table 1. V&V matrix M2. Defects types are in the columns (A-D) and V&V technologies are in the rows (1-4).**

defect types V&V	A	B	C	D
1	0.2[0.9]		0.2[0.5]	
2	0.5[0.9]	0.5[0.9]		
3		0.2[0.5]		
4			0.9[0.9]	0.2[0.5]

Step 2 - Argument 1: maximise completeness

(a) Matrix M2 can be transformed to the relative V&V matrix (M3) (that is, how the V&Vs perform in relation to each other). Transforming M2 to relative data can lead to the loss of detail; if necessary, a choice can be made to use actual data. Comparison of the relative effort of some V&V technologies based on metrics such as people-hours may not be sensible or appropriate, but for brevity this simplified approach is used to describe this example.

Relative data can be presented either as L (low), M (medium) and H (high) or through integers 1 (low), 2 (medium) and 3 (high). The range of the relative categories is determined by the practitioner. In the simple example of Table 1 the values are 0.2, 0.5, and 0.9 which we translate to L, M, and H in Table 2.

**Table 2. V&V matrix M3. Defects types are in the columns (A-D) and V&V technologies are in the rows (1-4).**

defect types V&V	A	B	C	D
1	L[H]		L[M]	
2	M[H]	M[H]		
3		L[M]		
4			H[H]	H[M]

If the data available is already relative (which may be the case if it is anecdotal or analytical), the

transformation will not be needed. It is important to save the matrix with the original data (M2) for future use when the data is updated.

(b) The effectiveness data in M3 is used to maximise completeness. A combination of V&V technologies has to be selected to ensure that all defect types of interest are covered by the V&V technologies. V&V technologies that detect defects uniquely (that is, there is no overlap in the matrix, such as for technology 4 in Table 2) must be chosen as part of the combination in order to ensure completeness. If only one combination of V&V technologies can ensure completeness, then step 3 is omitted and step 4 is performed.

Step 3 deals with V&V technologies that exhibit overlap in terms of defect-detection effectiveness (for example, technologies 1, 2 and 3 in Table 1).

Step 3 - Argument 2: minimise effort

(a) To minimise effort one needs to use the relative effort data and select a combination of techniques that take the least amount of effort. Minimising effort of a combination of V&V technologies does not necessarily mean minimising the overlap of effectiveness. It is important to note that a combination of V&V technologies may result in a lower combined effort than the technologies applied on their own (as seen in the results of a study evaluating the combination of static analysis tools and code inspection [34]). If one combination of V&V technologies appears to take less effort than any other, one can proceed to step 4.

(b) Otherwise, there may be a number of V&V technology combinations that seem to require a similar effort and therefore some preliminary evaluation may need to be performed to select the primary combination for defect detection.

Step 4 – Application and updating: gathering and updating cost-effectiveness information

(a) Once the combination of V&V technologies is selected, these technologies are applied in defect detection and performance data regarding their effectiveness and the effort needed to apply them is collected.

(b) The populated V&V matrix M2 can now be updated with the empirical data. Additionally, it is important to know whether or not the technologies were actually applied together (that is on the same artefact(s) possibly by the same group/individual) or separately by different groups/individuals. If they were applied separately the data collected will correspond to

the technologies on their own, but if they were applied together the data collected will correspond more specifically to that combination of V&V technologies and if a row in the matrix for this combination of V&V technologies does not yet exist, it will need to be added. If V&V technologies are applied together and the order of application is significant, this information needs to be taken into account as well and may result in different row entries in the matrix depending on the order of application.

Following the update to the V&V matrix M2, a second iteration of the strategy can be applied. It is possible that new V&V technologies (or combinations of V&V technologies) as well as new defect types may have been discovered, therefore the V&V matrix skeleton M1 may need to be updated with the relevant data.

### 3. Discussion

V&V matrix M3 may need to encompass additional information such as:

1. how effort and effectiveness data should be weighed with respect to each other;
2. cost to fix defects and/or severity of defects;
3. confidence levels about the data in the matrix.

The weight of effort and effectiveness data can affect the decision-making process. For example, cells 1A and 2A in Table 2 show that technology 1 can detect fewer defects with a lower effort than technology 2. If the practitioner needs to spend less effort on defect detection they may prefer technology 1 despite its lower effectiveness compared to technology 2. The cost to fix defects and the severity of defects may also influence choice of V&V technologies. These metrics can associate some type of priority with the defects that could be reflected by including a keyword or numeric value next to the defect type.

Wagner's [32, 33] model of quality economics presents a cost vs benefits analysis related to the effectiveness, effort, as well as cost to remove defects of V&V technologies. Since a lot of data that is necessary to apply the model may be unavailable, the model has undergone a sensitivity analysis [31] to reveal the most important metrics. Effort and effectiveness were found to be important in the analysis, but defect distribution, removal costs of the defect in the field, and failure probability were highly significant as well. Unfortunately there is currently little data on metrics other than effort and effectiveness for V&V technologies (and there is little data on these

metrics as well) [30], therefore further empirical investigations will be required before the quality economics model can be adequately applied.

Confidence levels can be included to reflect the reliability of the effectiveness and effort data; this data can be anecdotal, analytical, or empirical. Empirical data sources (specifically those in the same application area) may reflect the highest confidence; analytical data may reflect middle confidence, whilst anecdotal data may reflect the lowest confidence. The confidence levels for different types of data or different data sources can be a metric between 0 and 1 (0 being lowest confidence, 1 being highest confidence) and the related data can be placed in the matrix prior to decision making.

#### 3.1 Advantages

The main advantage of this strategy is that it is systematic and no longer relies on simple trial and error to determine what the best combination of V&V technologies will be. The strategy is also iterative to ensure that the results are checked and analysed to determine if the combination performs as expected. On each iteration, the strategy takes into account information regarding new technologies that may become available. Unexpected defects that may be discovered can also influence the combination of V&V technologies and further ensure that the combination selected will be the most complete and cost-effective.

A practitioner survey run in the context of V&V technologies for concurrent components [35], reported that both research literature as well as empirical information from industry is used in the decision-making process for V&V technology selection; this strategy allows practitioners to incorporate both sources of information. The practitioner survey also reported that it is important to evaluate V&V technologies using real defects; data collection during application ensures this is the case so that evaluation results will be relevant for modification of the V&V technology combination.

#### 3.2 Drawbacks

If the practitioner wishes to record the defects of interest by classifying them through defect type, this classification is not trivial. Available defect classification schemes (such as [6]) may not provide information detailed to a particular application context and more importantly they may be difficult to apply as practitioners may find it difficult to determine what categories their particular defects belong to [7].

There may be no prior data on the application of V&V technologies in the relevant context – the decision-making process may therefore become dependent on analytical and anecdotal data. Evaluations of the V&V technologies of interest may have been performed on components that are too different in terms of complexity, size, or other relevant attributes, and therefore comparing effectiveness and effort data between the technologies may not be accurate. Empirical data collected once the strategy is applied will reduce this inaccuracy.

The application of the strategy relies on a large amount of data collection. Although this may be a costly activity for the practitioner, it may be automated to take as little time as possible. Additionally, more than one combination may provide the lowest effort approach and therefore costly initial empirical evaluations may be required. The strategy itself depends on a simplification of the data required in the decision process as there are many aspects to cost-effectiveness that may be significant to practitioners [35], but practitioners can choose not to use certain V&V technologies if they do not meet their other cost-effectiveness requirements.

#### 4. Relation to existing work

The combination strategy, as presented in this paper, simplifies cost-effectiveness aspects of V&V technologies to effectiveness and effort. Effort (for example, measured in terms of the people-hours required to apply the V&V technology) may not encompass all the costs of interest to practitioners, therefore the application of Wagner's [32, 33] software quality economics could be incorporated in the future.

Defect analysis has been used previously to improve software process [7] and defect information influences the choices made in using the strategy. Barrett et al.'s [3] mapping matrix was used in test process optimisation by analysing data on V&V technologies and their ability of finding types of defects. If a V&V technology detects a type of defect in the mapping matrix, the cell corresponding to the row of the V&V technology and column of the defect type is marked with an X, otherwise it is left blank. The combination strategy extends Barrett et al.'s work with the definition of a process of forming and utilizing the matrix (M2) and including cost information in the matrix itself.

Combination of V&V technologies has been explored through analysis and redefinition of the classification of V&V technologies [37]. Since the traditional classification of V&V technologies as static and dynamic did not seem adequate, a new taxonomy

of V&V technologies was presented that broke them down to technologies that sample the space of possible executions and technologies that fold the space (by abstracting away details). However, the combinations recommended were based on a further classification to technologies with optimistic (failure to reject an incorrect program) and pessimistic accuracy (failure to accept a correct program) that actually corresponds to the static (pessimistic) and dynamic (optimistic) classification.

Murnane et al.'s [21] tailoring of black-box methods for test case selections is more fine-grained than the combination strategy because it does not look at the combination of V&V technologies in general; it focuses on test case selection, which is only one aspect of testing and V&V more generally.

The combination strategy presented in this paper provides an empirical framework for V&V technology evaluation like that seen in the framework presented by Bradbury et al. [5], but it differs because their work focuses on testing and formal analysis and does not include automated static analysis and other types of technologies, and currently focuses on gathering information on V&V technologies for detecting defects in C. Bradbury et al.'s framework also deals with debugging, which is not the focus of the strategy presented in this paper. Bradbury et al.'s framework relies on mutation for defects and automated V&V technologies, so it does not accurately reflect the process of a practitioner applying the V&V technologies on actual defects.

Vegas and Basili [28] have developed a characterization schema to store a vast amount of information on testing techniques. The schema consists of a classification system that starts with three levels: tactical, operational and historical. Each level, breaks down to corresponding elements and the elements are further broken down into attributes. For example, the tactical information level consists of "objective" and "scope" elements. The tactical element "objective" consists of the attributes: purpose, defect type and effectiveness. A V&V technology is inserted into the schema once its attributes have been classified. For example, a decision coverage technique has the purpose of finding defects, the defect type it can detect is control and its effectiveness is that it can detect 48% of control defects [29]. The attributes can then be compared to the testing requirements to decide on the appropriate V&V technology or combination of technologies to use, but there is no process associated with the schema to assist practitioners in selecting these techniques. There are many factors to consider, many of which are rather fine-grained and specific to test

case selection techniques, making it hard to characterize other V&V technologies such as model-checking and automated static analysis tools.

To make an appropriate selection, a V&V practitioner has to consider not only the technologies on their own but also in combination. The characterization schema contains one attribute referring to the relationship a particular test case selection technique has with others, but this relationship characteristic does not encompass how well these techniques perform together, that is, how the combination of the techniques influences their effort and effectiveness.

Shull and Turner's [27] best practices clearinghouse acts more like a repository rather than a process for V&V technology selection, as does the characterization schema [28] for the selection of software testing techniques. The combination strategy provides guidance for practitioners regarding how they can use the cost-effectiveness data on the V&V technologies available to them in order to aid their selection and/or decision-making process.

Koomen and Pol's test process improvement (TPI) [15] includes a test maturity matrix that splits up the test process into key areas (different points of view of testing) that includes metrics, test tools and techniques such as static test and test specification. Each key area has a level associated with it that refers to the maturity of the key area on a scale that defines the area as controlled, efficient or optimising (in order of increasing maturity). Although TPI assesses and suggests improvements to the test process and includes evaluation of the improvements so TPI can be performed iteratively, it does not refer to particular V&V technologies, nor does it provide a strategy to combine them.

## 5. Preliminary evaluation of the combination strategy

The combination strategy will be applied and evaluated in the context of concurrent components. A failure analysis of concurrency components classified concurrency defects into three main categories: interference, deadlock and functional defects [19]. The classification of concurrency defects lead to the development of the TestCon method. The method currently combines automated static analysis, code inspection and dynamic analysis; but the strategy can suggest modifications using data on the cost-effectiveness of model-checkers, assured evolution tools (such as Fluid [10]), and other dynamic tools (such as ConTest [9]).

Some preliminary cost-effectiveness data has been collected from previous studies on V&V technologies [17, 24, 34] available for concurrent components (see Table 3). Effort data is recorded as minutes required to apply V&V technology, since the components in all the studies were small. Most data is derived from a study evaluating static analysis and inspection [34]; data in the study was collected on code inspection steps alone, automated static analysis (using Jlint and FindBugs) and a combination of automated static analysis tools and code inspection. Code inspection was combined with static analysis tools to examine the reports they generate, because the tools can fail to report actual defects or report false positives. Since the V&V technologies were applied to the same set of components and defects, the study provided comparable data for the matrix.

**Table 3. V&V matrix M2 in the context of the TestCon method and the JPF model-checker.**

defect types V&V	Interference	Deadlock	Functional Defects
Inspection	7.25 [0.375]	3.375 [1]	6.625 [0.187]
Static Analysis Tool	7.25 [.437]	3.25 [0.5]	10.5 [0.5]
Dynamic Testing	? [0.57]	? [0.667]	? [1]
Model-Checking	? [0.857]	11.6 [1]	? [0.200]
Inspection and Static Analysis Tools	3.354 [0.675]	4.73 [0.712]	4.75 [0.5]

The remaining data is derived from exploratory studies using mutants [17] and comparing the JPF model-checker with Jlint and code inspection [24]. The mutant study provides data on dynamic testing using ConAn; it also includes data on code inspection and Jlint and FindBugs, but it does not include effort data. The effort and effectiveness data of inspection combined with Jlint is an average of the data provided by Ngui et al. [24] and Wojcicki and Strooper [34]. Different components and defects were used in the three studies and this can affect the accuracy of the comparison between technologies.

Utilising this data, the strategy will be employed to determine the most cost-effective combination. Setting the relative effort and effectiveness ranges to  $0 < L \leq 0.33, 0.33 < M \leq 0.66, 0.66 < H \leq 1$ , data from M2 (Table 3) has been transformed to M3 (Table 4).

The combination will be evaluated through a case study focusing on industrial components and/or a controlled experiment focusing on toy components. Reusable constructs for data collection will be developed to assist the process of experimentation as well as the strategy itself.

**Table 4. V&V matrix M3 in the context of the TestCon method and JPF.**

defect types V&V	Interference	Deadlock	Functional Defects
Inspection	H [M]	L [H]	M [L]
Static Analysis Tool	H [M]	L [M]	H [M]
Dynamic Testing	? [M]	? [M]	? [H]
Model-Checking	? [H]	H [H]	? [L]
Inspection + Static Analysis Tools	M [H]	M [H]	M [M]

Once the evaluation is complete, an analysis will be conducted to determine whether the new combination of V&V technologies is more cost-effective, thus evaluating the success of the combination strategy. The strategy may then be adjusted based upon the evaluation. The strategy can then be applied on the combination of V&V technologies once more to improve the cost-effectiveness of the combination and evaluate the adjusted strategy.

## 6. Conclusions

The combination strategy presented in this paper provides a systematic approach to the selection of combinations of V&V technologies based on cost-effectiveness data. Unlike repositories of V&V technologies, the strategy provides decision-making support by guiding the selection of V&V technologies. The strategy is a practice-based approach that can benefit from appropriate empirical data. Although the data may be analytical or anecdotal to start with, the strategy is iterative and depends on the collection of empirical data once a combination of V&V technologies is applied in order to improve upon the combination.

## References

- [1] G. Andrews, *Concurrent Programming: Principles and Practice*: Addison-Wesley, 1991.
- [2] C. Artho, "Finding faults in multi-threaded programs": Federal Institute of Technology, Zurich-Austin, 2001.
- [3] N. Barret, S. Martin, and C. Dislis, "Test Process Optimization: Closing the Gap in the Defect Spectrum", In *Proceedings of the International Test Conference*, 1999, pp. 124-129.
- [4] V. R. Basili and R. W. Selby, "Comparing the Effectiveness of Software Testing Strategies", *IEEE Transactions in Software Engineering*, vol. 13, 1278-1296, 1987.
- [5] J. S. Bradbury, J. R. Cordy, and J. Dingel, "An Empirical Framework for Comparing Effectiveness of Testing and Property-Based Formal Analysis", In *Proceedings of the 6th ACM SIGPLAN-SIGSOFT Workshop on Program Analysis for Software Tools and Engineering*, 2005, pp. 2-5.
- [6] D. Card, "Managing Software Quality with Defects", In *Proceedings of the 26th Annual International Computer Software and Applications Conference*, 2002, pp. 472.
- [7] L. O. Damm and L. Lundberg, "Identification of Test Process Improvements by Combining Fault Trigger Classification and Faults-Slip-Through Measurement", In *Proceedings of the 2005 International Symposium on Empirical Software Engineering (ISESE'05)*, 2005, pp. 152-161.
- [8] A. Endres and D. Rombach, *A Handbook of Software and Systems Engineering*: Addison Wesley, 2003.
- [9] E. Farchi, Y. Nir, and S. Ur, "Concurrent Bug Patterns and How To Test Them", In *Proceedings of the 17th International Parallel and Distributed Processing Symposium (IPDPS 2003) - 1st International Workshop on Parallel and Distributed Systems: Testing and Debugging*, 2003
- [10] A. Greenhouse, T. J. Halloran, and W. L. Scherlis, "Observations on the Assured Evolution of Concurrent Java Programs", *Science of Computer Programming*, vol. 58, 384-411, 2005.
- [11] K. Havelund and T. Pressburger, "Model Checking Java Programs using Java Pathfinder", *International Journal of Software Tools for Technology Transfer (STTT)*, vol. 2, 366-381, 2000.
- [12] W. C. Hetzel, "An experimental analysis of program verification methods": University of North Carolina, 1976.
- [13] D. Hovemeyer and W. Pugh, "Finding Concurrency bugs in Java", In *Proceedings of the 23rd Annual ACM SIGACT-SIGOPS Symposium*

- on Principles of Distributed Computing (PODC 2004) Workshop on Concurrency and Programs, 2004
- [14] E. Kamsties and C. M. Lott, "An Empirical Evaluation of Three Defect-Detection Techniques", In *Proceedings of the Fifth European Software Engineering Conference*, 1995, pp. 1-22.
- [15] T. Koomen and M. Pol, *Test Process Improvement: A practical step-by-step guide to structured testing*: Addison-Wesley, 1999.
- [16] O. Laitenberger, B. Freimut, and M. Schlich, "Combination of Inspection and Testing Technologies", Fraunhofer IESE 070.02/E, 2002.
- [17] B. Long, R. Duke, D. Goldson, P. Strooper, and L. Wildman, "Mutation-Based Evaluation of a Method for Verifying Concurrent Java Components", In *Proceedings of the 2nd International Workshop on Parallel and Distributed Systems: Testing and Debugging (PADTAD)*, 2004
- [18] B. Long, P. Strooper, and D. Hoffman, "Tool Support for Testing Concurrent Java Components", *IEEE Transactions in Software Engineering*, vol. 29, 555-566, 2003.
- [19] B. Long, P. Strooper, and L. Wildman, "A Method for Verifying Concurrent Java Components", *To appear in Concurrency and Computation: Practice and Experience*, 2006.
- [20] J. Magee and J. Kramer, *Concurrency: State Models & Java Programs*: John Wiley & Sons, 1999.
- [21] T. Murnane, K. Reed, and R. Hall, "Tailoring of Black-Box Testing Methods", In *Proceedings of the 2006 Australian Software Engineering Conference (ASWEC'06)*, 2006, pp. 292-299.
- [22] M. Musuvathi and D. Engler, "Some Lessons from Using Static Analysis and Software Model Checking for Bug Finding", *Electronic Notes in Theoretical Computer Science*, vol. 89, 378-404, 2003.
- [23] G. Myers, "A Controlled Experiment in Program Testing and Code Walkthroughs/Inspections", *Communications of ACM*, vol. 21, 760-768, 1978.
- [24] J. Ngui, P. Strooper, L. Wildman, and M. Wojcicki, "Comparing the Cost-effectiveness of Statically Analysing and Model Checking Concurrent Java Components for Deadlocks", In *Submitted to the Australian Software Engineering Conference (ASWEC '06)*, 2006
- [25] G. Ruhe, "Software Engineering Decision Support and Empirical Investigations - A Proposed Marriage", *Empirical Studies in Software Engineering*, vol. 2, 25-34, 2003.
- [26] R. W. Selby, "Combining Software Testing Strategies: An Empirical Evaluation", In *Proceedings of the ACM/SIGSOFT IEEE Workshop on Software Testing*, 1986, pp. 82-90.
- [27] F. Shull and R. Turner, "An Empirical Approach to Best Practice Identification and Selection: The US Department of Defense Acquisition Best Practices Clearinghouse", In *Proceedings of the 2005 International Symposium on Empirical Software Engineering (ISESE'05)*, 2005, pp. 133-140.
- [28] S. Vegas and V. Basili, "A Characterization Schema for Software Testing Techniques", *Empirical Software Engineering*, vol. 10, 437-466, 2005.
- [29] S. Vegas, N. Juristo, and V. Basili, "A Process for Identifying Relevant Information for a Repository: A Case Study for Testing Techniques", in *Managing Software Engineering Knowledge*, A. Aurum, R. Jeffery, C. Wohlin, and M. Handzic, Eds.: Springer-Verlag, 2003, pp. 199-230.
- [30] S. Wagner, "A Literature Survey of the Quality Economics of Defect-Detection Techniques", In *Proceedings of 5th ACM-IEEE International Symposium on Empirical Software Engineering (ISESE'06)*, 2006, pp. 194-203.
- [31] S. Wagner, "A Model and Sensitivity Analysis of the Quality Economics of Defect-Detection Techniques", In *Proceedings of the International Symposium on Software Testing and Analysis (ISSTA '06)*, 2006, pp. 73-83.
- [32] S. Wagner, "Modelling the Quality Economics of Defect-Detection Techniques", In *Proceedings of the Workshop on Software Quality (WOSQ'06)*, 2006, pp. 69-74.
- [33] S. Wagner, "Software Quality Economics for Combining Defect-Detection Techniques", In *Proceedings of the Net.Object Days 2005 (Node'05)*, 2006, pp. 559-574.
- [34] M. Wojcicki and P. Strooper, "Maximising the Information Gained from an Experimental Analysis of Code Inspection and Static Analysis for Concurrent Java Components", In *5th ACM-IEEE International Symposium on Empirical Software Engineering (accepted)*, 2006
- [35] M. Wojcicki and P. Strooper, "A State-of-Practice Questionnaire on Verification and Validation for Concurrent Programs." *Accepted to the 4th International Workshop on Parallel and Distributed Systems: Testing and Debugging (PADTAD)*, 2006.

[36] M. Wood, M. Roper, A. Brooks, and J. Miller, "Comparing and Combining Software Defect Detection Techniques: A Replicated Empirical Study", In *Proceedings of the 6th European Software Engineering Conference*, 1997, pp. 262-277.

[37] M. Young and R. N. Taylor, "Rethinking the Taxonomy of Fault Detection Techniques", In *Proceedings of the 11th International Conference on Software Engineering*, 1989, pp. 53-62.