Confidence-Based Reliability and Statistical Coverage Estimation

William E. Howden
University of California, San Diego
La Jolla, Ca howden@cse.ucsd.edu

Abstract

In confidence based reliability measurement we determine that we are at least C confident that the probability of a program failing is less than or equal to a bound B. The basic results of this approach are reviewed and several additional results introduced, including the adaptive sampling theorem which shows how confidence can be computed when faults are corrected as they appear in the testing process. Another result shows how to carry out testing in parallel. Some of the problems of statistical testing are discussed and an alternative method for establishing reliability called statistical coverage is introduced. At the cost of making reliability estimates that are relative to a fault model, statistical coverage eliminates the need for output validation during reliability estimation and allows the incorporation of non-statistical testing results into the statistical reliability estimation process. Statistical testing and statistical coverage are compared, and their relationship with traditional reliability growth modeling approaches is briefly discussed.

1. Introduction

Software developers are often faced with the need to estimate the reliability of a program, and to decide when to stop testing. One of the more commonly used methods at the earlier stages of development is test coverage. There will typically be a requirement that some percentage of program branches be covered on at least one test, say 90%. This paper reviews the basic theory that is used for statistical testing and introduces several new results. It then describes the application of these methods to coverage based testing. An approach is described in which it is possible to compute reliability oriented partial coverage measures, and to avoid the problems of infeasible coverage items. Potential applications of the method, including a review of an industrial project that can be interpreted as having used it in an informal way, are discussed.

2. Theoretical foundations

The basic statistical approach that is used is hypothesis testing. In this approach we formulate an hypothesis H and design a statistical experiment E for verifying H. E is designed so that the assumption that H is false determines an upper bound on the outcomes of E. If the assumption that H is false produces an upper bound U on an observed outcome e of E, we conclude that H is true with confidence at least C = 1-U. Intuitively, the approach is based on the idea that if the assumption that H is false leads to an outcome bound U that is small, then the assumption that H is false is led to an unlikely event, indicating that H is probably true.

The application of hypothesis testing that is used in this paper involves the estimation of the "size" of a subset D' of D. The size of a subset D' is the probability of selecting an element from D' when items are selected at random from the whole set D. It is assumed that elements are selected at random according to some probability distribution for the set D. The hypotheses are that the size of D' is less than or equal to a bound B. It is assumed in the following results that probabilities are discrete, so that the domains of interest are finite and have distributions in which each element has some probability of occurrence when items are randomly selected.

The first two theorems are well known results in the area of statistical testing.

Theorem 1 Suppose that in a sequence of N random samples from a domain D we see no elements of a subset D'. Then we can have confidence at least C, where

$$C = 1-(1-B)^N$$

that probability of choosing an element from D' is less than B.

A more general result, when some items from D' are seen, can also be derived.

1071-9458/97 $10.00 © 1997 IEEE
Proceedings of the 8th International Symposium on Software Reliability Engineering (ISSRE'97)
0-8186-8120-9/97 $10.00 © 1997 IEEE

Theorem 2 Suppose that in a sequence of N random samples from a domain D, we see n elements from a subset D'. Then we can have confidence at least C, where

$$C=1-\left(\binom{N}{0}B^0(1-B)^N+\binom{N}{1}B(1-B)^{N-1}+\ldots+\binom{N}{n}B^n(1-B)^{N-n}\right)$$

that the probability of selecting an item from D' is less than or equal to B.

The discussion of confidence based testing has periodically reappeared over the last 20 years, and variations of the above formulae have been described by different authors [e.g. 1-8].

In the above results, we assumed that sampling was done with replacement, and that the domains D and D' were fixed at the beginning of the statistical process. In the case of testing and test coverage this is often too restrictive. If D' is the subset of a program's domain D over which a program fails, then if we correct faults as they occur, D' will change, and grow smaller. We present here an alternative result, called *adaptive sampling,* in which D' can change in this manner.

Theorem 3 (adaptive sampling). Suppose that a sequence S of N samples is made from a domain D, with a subset D'. Suppose that each time a sample item comes from D', it and possibly other items are moved from D' to D~D', and a new D' set is constructed. Suppose that during the sequence S thare are n samples in which an item is from the current version of D' occurs. Let D" be the final version of D', constructed using S. Then we can have confidence at least C, where

$$C= 1-\left(\binom{N}{0}B^0(1-B)^N+\binom{N}{1}B(1-B)^{N-1}+\ldots+\binom{N}{n}B^n(1-B)^{N-n}\right)$$

that the probability of selecting an item from D", when a random item is selected from D, is less than or equal to B. i.e. the same formula that was used for estimating D' in Theorem 2 can also be used for estimating D", the adapted version of D'

Proof. The proof is based on the following observation. Suppose that D' and D" are two subsets of D. Suppose that the probability of choosing an item from D" is less than choosing one from D'. Consider the probability P' of drawing at most n (i.e. ≤n) items from D' in a set of N samples from D, and P" the probability of drawing at most n items from D" in a set of N samples from D.

The probability of drawing at most n items from D' will be smaller than that of drawing at most n from D", since for each sample from D' the probability of getting an item is higher than for D" so that we would have expected more subset items to be seen, i.e. $P" \geq P'$.

In the above we assumed that items are drawn from fixed sets D' or D". Call this the *simple sampling* method, as opposed to adaptive sampling, under consideration in this theorem. In adaptive sampling we start with a domain D and a subset $D' = D_1$ of special items, and when a item is found in D_1, it and possibly other items are "moved out" of D_1 into $D \sim D_1$, to create a new special item subset $D' = D_2$. Assume that in a sample of N items, new items from a subset D_i are seen on n of those samples. Let $D" = D_n$.

Let P' be the probability of seeing at most n special items when the adaptive sampling method is used, and where the final set D_n is such that the probability of selecting an item from D_n is greater than some bound B. This means that each time a sample was drawn, there was a probability of at least B that a special item would be seen. Let P" be the probability of seeing at most n special items, if simple sampling were used on a set for which the probability of seeing a special item was = B. Now $P' \leq P"$, because the probability of seeing a special item during the simple sampling process will be less than that of having seen a special item during adaptive sampling, so it will be more likely for there to be at most n special items in the simple process, i.e. $P" \geq P'$. Hence, if the probability of seeing a special item from the final version $D" = D_n$ of D' were >B, the probability of having seen special items from earlier versions of the sets D' on at most n samples of the N tests is bounded by

$$P' \leq P" =\binom{N}{0}B^0(1-B)^N+\binom{N}{1}B(1-B)^{N-1}+\ldots+\binom{N}{n}B^n(1-B)^{N-n}.$$

This implies that we can have confidence at least C = 1-P" that the probability of seeing an item from the final version D" of D' on the next sample is less than or equal to B. §

In addition to the above basic theorems, we can also develop theorems that involve combinations of results. The first theorem below, which was first reported in [9], addresses the important situation where we have independently verified the same result several times. It indicates how we can combine results from parallel efforts.

Theorem 4 (parallelization) Suppose that from K independent sets of N samples we can have confidence at least C_i that the size of a subdomain D' of D is less than or equal to B, $1 \leq i \leq K$. Then we can have confidence at least C, where

$$C = 1-(1-C_1)(1-C_2)\ldots(1-C_K)$$

that the subdomain D' is less than or equal to B.

Proof: Suppose that p is the probability of choosing an element from D'. If we have determined from the i'th experiment that we can have confidence at least C_i, $1 \leq i \leq K$,

Proceedings of the 8th International Symposium on Software Reliability Engineering (ISSRE'97)
0-8186-8120-9/97 $10.00 © 1997 IEEE

that p is less than or equal to B, then we have seen the joint occurrence of K independent events each of which has the following property. The assumption that p>B implies that the probability of their occurrence was less than $1-C_i$, $1 \le i \le K$. This means that the probability of their joint occurrence was at most

$$(1-C_1)(1-C_2)...(1-C_K)$$

so that we can have confidence at least C that $p \le B$, where

$$C = 1-(1-C_1)(1-C_2)...(1-C_K). \ \S$$

The above theorem could be used in the following way. The earlier theorems show how many tests are needed in which a certain fraction of the elements do not fall in a subset D' in order to achieve a required level of confidence C in a bound B. If B is very small, then an enormous number of tests are needed to achieve a reasonable level of confidence C. What we do is to instead choose K small confidences C_i, reducing the size of the number of required tests, and then carry out K testing experiments in parallel in order to establish a cumulative C at a reasonable level.

Theorem 4 cannot be applied when adaptive sampling is used, since each of the K experiments would produce results for different final sets D". This indicates that, as it stands, it should be used with systems where we will carry out a reliability testing process during which the set of items of interest, D', is fixed. However, similar results can be achieved if we assume the following. During an adaptive sampling process elements are moved from D' to D~D', resulting in a new D', and eventually a final set D". We may not know what is in D' or D", but we may be able to assume that we will know the contents of the subsets R' of D' that are moved in this way. Let R" be the union of all of the sets R' that are identified during a sampling process. This means that at the end of an adaptive sampling process we will know that D" = D'~R", but still of course not necessarily know the original D' or the final D". Call the sets R" the *sample change* sets.

Theorem 5 (adaptive sampling with parallelization) Suppose that a set of K independent testing experiments is carried out, and that we are able to establish with confidence C_i that the reduced sets D_i'' that are produced at the end of each of the experiments are bounded in size by B, $1 \le i \le K$. Let R_i'' be the sample change subset for the i'th experiment. Let R" be the union of all the sample change subsets for the independent experiments. Then we can conclude with confidence at least C, where

$$C = 1-(1-C_1)(1-C_2)...(1-C_K)$$

that the probability of selecting an element from

$$\prod_{i=1}^{K} D_i'' = D_j'' \sim R''$$

is less than or equal to B, where j is any index in the range $1 \le j \le K$, where the "π" symbol stands for set intersection, and the "~" for set subtraction.

The implication of the above is that we carry out K independent experiments and record the sample change sets for each experiment. At the end of all of the experiments, we choose any individual experiment and use the total sample change set R" to remove any additional items from its final D" set that were removed in the other experiments. The "size" of the resulting set D" will be ≤ B.

Proof: The last equation holds because it will subtract off from any D_j'' the subsets of items that were removed from D' in forming the D_i'' for each of the experiments, leaving them all the same.

Our hypothesis will be that there is at least one j, $1 \le j \le K$, for which the final set D_j'' is ≤ B. If this is not true, then all of the final sets D_i'' must be larger than B. This implies that we have seen K independent experimental results the probability of whose separate occurrences are less than $(1-C_i)$, $1 \le i \le K$, or whose joint occurrence is bounded by the product of these terms. Hence we can have the above confidence that this is not true, that at least one final D_j'' is ≤ B. Now if at least one of these is ≤ B, the intersection of all of the D_i'' must be ≤ B, so that we can have the same level of confidence in this derived result, proving the theorem. §

The following theorem has a variety of applications.

Theorem 6 (domain decomposition) Suppose that a domain D can be partitioned into K disjoint subsets D_i and that D'_i = intersection(D',D_i), $1 \le i \le K$. Recall that the size of D'_i relative to D_i is defined to be the (sample distribution weighted) fraction of D_i that lies in D'_i , i.e. it is the conditional probability of choosing an item from D'_i given that an item from D_i is chosen. Suppose that we can have confidence C_i that the relative size of D'_i is less than or equal to B, $1 \le i \le K$. Then we can have confidence min{C_i, $1 \le i \le K$} that the size of D' is less than or equal to B.

Proof: Suppose that the size of D' is >B. Since the sets D'_i are disjoint, the probability of choosing an item from their union is the sum of the probabilities of choosing an item from each of the sets, i.e.

$$\text{size}(D') = \text{size}(D'_1)+\text{size}(D'_2)+...+\text{size}(D'_K).$$

Also, the sum of the sizes of the D_i, $1 \le i \le K$, must be 1. This implies, that if size(D')>B, that

$$\text{size}(D'_1)+\text{size}(D'_2)+...+\text{size}(D'_K) > B(\text{size}(D_1)+\text{size}(D_2)+...+\text{size}(D_K))$$

0-8186-8120-9/97 $10.00 © 1997 IEEE

so that for some j, where $1 \le j \le K$,

$size(D'_j) > B(size(D_j))$,

since otherwise the inequality would be false. The probability of choosing an item from D'_j, given that an item from D_j is chosen, is given by the Bayesian formula

probability(item from D'_j and item from D_j)
    ÷ probability(item from D_j)

and since D'_j is a subset of $D_{j,}$ this is equal to

(probability(item from D'_j) ÷ probability(item from D_j)).

    $= size(D'_j) \div size(D_j)$

so that if size(D')>B, then for some j

probability(item from D'_j, given an item from D_j) >B.

But we know with confidence C_j that this probability is $\le$ B. This means that if size(D')>B, then during the determination of the bound for $size(D'_j)$, an experimental sample occurred whose probability was less than $1-C_j$. i.e. we can have confidence at least C_j that size(D')≤B. Since we do not know the value of j, we can take the maximum of the event probabilities $1-C_i$, $1 \le i \le K$, i.e. the minimum of the confidence levels C_i, $1 \le i \le K$. §

The following corollary makes it possible to replace a sampling problem involving a non-uniform distribution with K sampling uniform distribution problems.

Corollary 6.1 (uniform decomposition). Suppose that the distribution of items in D is uniform over D_i, $1 \le i \le K$. Suppose that for each D_i we use random sampling according to the uniform distribution to establish with confidence at least C that the size of D'_i = intersection(D', D_i), relative to D_i, is less than or equal to B. Then we know with confidence at least C that the size of D' is less than or equal to B.

The special case of Corollary 6.1 where no items from D' are found during sampling, and the same number of tests can be used to establish a common confidence level for each subdomain, is reported in [10], in which [6] is referenced.

The following corollary is used later in the paper to deal with situations where the input to a program consists of both internal data I, and external data E.

Corollary 6.2 (projective domain decomposition) Suppose that D = I*E is the cross product of two domains I and E. Let D' be a subset of D. Assume that I can be decomposed into K disjoint subdomains I_i, $1 \le i \le K$, having the following properties. Let slice(I_i) be the subset of the input domain corresponding to the cross product I_i*E. Assume that we can have confidence at least C_i that the size of intersection(slice(I_i), D') relative to slice(I_i) is less than or equal to B, $1 \le i \le K$. Then we can have confidence min{Ci, $1 \le i \le K$} that the size of D' is less than or equal to B.

The following three corollaries describe situations in which Corollary 6.2 is applicable.

Corollary 6.3 (projective distribution decomposition) Suppose that D, I, E and D' are defined as above. Assume that for each subdomain I_i*E, there is a separate probability distribution p_i that gives the frequency of occurrence of items in slice(I_i), relative to slice(I_i). Suppose that for each i, $1 \le i \le K$, we sample from slice(I_i) using the sample distribution p_i, and determine with confidence C_i, that the size of intersection(D', slice(I_i)), relative to slice(I_i), is less than or equal to B. Then we can conclude with confidence min{C_i, $1 \le i \le K$} that the size of D' is less than or equal to B.

Corollary 6.4 (uniform projection) Suppose that D = I*E is the cross product of two sets I and E, and assume that there are probability distributions p and q for I and E, such that for each (x,y) in I*E, the probability of occurrence of (x,y) is p(x)*q(y). Assume that there is a partition of I into disjoint sets I_i, $1 \le i \le K$, such that the distribution p is uniform over each I_i. Consider the following factored approach to choosing samples from slice(I_i) = I_i*E. First an element x is chosen randomly from I_i, and then combined with a random element y from E. In the case of I_i, the uniform distribution is used, and in the case of E, its distribution q is used. Suppose that this method of sampling is used to establish with confidence C_i that the size of the intersection of slice(I_i) with a set D' in I*E, relative to slice$(I)_i$, is less than or equal to B. Then we can have confidence min{C_i, $1 \le i \le K$} that the size of D' ≤B.

Corollary 6.5 (reducible uniform projection) Assume that D, I and E are as in Corollary 10.4, and that D' is a subset of I*E for which the following also holds. Consider some I_i, where i is in $1 \le i \le K$. For all z in E , and all x and y in I_i, assume that (x,z) is in D' if and only if (y,z) is in D'. Suppose that we can show for at least one x from each I_i that we can have confidence at least C that the size of intersection(slice({x}), D') is less than or equal to B. Then we can have confidence at least C that the size of D' is less than or equal to B.

3. Coverage and fault models

The above theory can be applied to statistical testing in the following way. The subset size that we are trying to estimate will be that of the subset of a program's input domain over which the program fails, i.e. its *failure density*. Theorem 3 can be used if we run N tests, and see n failures, and we fix the corresponding fault each time a failure is seen. The only restriction on this application is that each time we fix a fault we reduce the failure density, or at least do not increase it, i.e. fault repair is either perfect or at least "productive". Theorem 4 indicates how stronger reliablity results could be obtained by carrying out independent parallel testing efforts. Theorem 5 indicates that we can achieve the given reliability result even while adaptive sampling is used if we keep track of the faults that are removed in each of the independent experiments, and also remove them from the others. As above, we have to assume that fault repair is productive, i.e. that the size of the failure density is not increased when a fault is repaired.

There are several major difficulties in using statistical testing. One is the large number of tests and subsequent output validations that must be performed. Others are the failure to distinguish between critical and non-critical cases, and between cheap and expensive tests. These problems encourage the use of a non-statistical, coverage oriented approach to testing.

Coverage testing is associated with a strategy for decomposing a program's input domain into subsets, and a requirement that at least one test be selected from each subset. In some cases the subsets are disjoint, and in others they overlap. A variety of different kinds of coverage strageties can be identified [e.g. 11-13]. The most widespread program coverage method requires that each branch in a program be executed on at least one test.

The coverage subsets that are generated by different coverage methods can be thought of as fault models. A fault model M is defined to be a set of subsets F of a program's input domain that has the property that if there is a fault of type M present, there is some subset in F over which the program fails for all elements of that subset. Coverage testing, from this viewpoint, is a method for insuring the absence of faults associated with different fault models.

In general, model based testing will only guarantee the absence of major faults in which a program fails for all data in a coverage element. In practice, a program may fail for part of a coverage element. This can be interpreted as resulting from two possible causes. The first is when a model is not refined enough, the subsets need to be smaller, defined perhaps using a compound coverage strategy that takes more program detail into account. The other cause is when a program contains a fault of omission, where code or data is missing that is needed to define more refined coverage subsets. If the information that is needed for the more refined definition is not available, then coverage based testing will be unsuccessful. Based on this, we conclude that coverage based fault models are good for errors of commission, in which the program fails for particular data or for particular program constructs, but weak for errors of omission.

The need for more refined models in which it would be less common for a program to fail for part of a model subset can be interpreted as one of the motivations for the development of compound models in which coverage subsets correspond to combinations of simple coverage elements. Current compound program oriented coverage methods include def-use coverage in which it is necessary to carry out tests which execute each pair of statements where the first defines data used by the second [14]. Others include LCAJS [15] testing, in which certain combinations of sequences of program statements have to be covered on at least one test.

Compound coverage can also be defined at a systems level, where coverage items correspond to possible interactions between different parts of a system. Different approaches may be identified with different kinds of systems. In a rule-based system, combinations of rules are used in response to different stimuli. Each combination can be thought of as an abstract system state that occurs when the program is used. In a distributed system, interactions may correspond to different possible subsequences of communication primitives, where one task calls another task while that task is in the process of calling a third.

4. Statistical coverage and model based reliability estimates

Two major problems in the use of coverage models are infeasible coverage elements, and the reliability interpretation of partial coverage.

Even for simple, non-compound, coverage, it is common to require less than 100% coverage. There can be several reasons for this. One possibility is the effort it takes to find tests for that last 10% of the program's branches or other coverage elements. Related to this is the problem of infeasible coverage elements: there may be no data that causes some items to be covered, i.e. their associated fault model subsets are empty.

The infeasible coverage element problem is exacerbated when compound coverage is used: many coverage items may be infeasible. For example, suggestions for testing and analysis of distributed systems often involve the use of system reachability graphs [16]. These show the set of all possible states and state transitions for a system. Such graphs are enormous, and suggestions that we test sequences of branches that correspond to interleaved possible states is impractical. Different reduction approaches to this problem have been suggested [e.g. 17], but, in general, the problem of determining which arcs in a reachability graph are feasible in order to determine if complete or partial coverage has occurred, may be very difficult.

The problem of infeasible coverage items is not just that many items may be infeasible, but that it is difficult to know which are feasible and which are not, so that it is difficult to know when "complete" coverage of all feasible items has been achieved. It will be just as difficult to know when partial coverage has been achieved. Even though it is not the general case, this can be a problem even when simple compound coverage is used, such as def-use. [14].

In addition to the infeasible element problem, one of the problems of coverage oriented testing is the lack of a reliability interpretation of partial coverage. What does it mean that we have tested 90% of all feasible coverage items? These problems can be approached using statistical coverage.

In statistical coverage, we do not require that the set of all possible compound coverage items be *a priori* identified in order to determine if coverage has been achieved. Instead, we generate test data according to a program's operational distribution, and continue to require additional coverage tests until it is unlikely that new coverage items will appear.

Statistical coverage solves the problem of finding a reliability interpretation for partial coverage in the following way. Suppose that M is a fault model associated with a set of coverage sets F for a program. Let F' be a subset of F that has been observed during a set of tests. Suppose that we are C confident that the probability of seeing a new coverage item from F~F' on a subsequent test is less than or equal to B, and we have established that the program does not fail on a test set in which there is at least one test from each subset in F'. Then we can be C confident that the probability of the program failing on a subsequent program execution due to a fault of type M is less than or equal to B. Recall that we say that a program has a fault of type M if and only if it fails on all elements of an associated subset in F.

The basic theory from the Theoretical Foundations Section of the paper can be used in the statistical approach to coverage in the following way. We assume that as testing progresses, there is some subset D' consisting of input data that would cause a new coverage item to be seen. Once a new coverage item is seen, it is no longer new so that the set D' changes to a new set D' when the input data that caused the new coverage item to be covered is "moved out" of D'. This situation corresponds to that described in Theorem 3, which gives a formula for computing confidence when n new coverage items have been seen in a sequence of N tests.

The adaptive sampling theorems (3 and 5) apply to the coverage testing situation in a natural way. Each time a new coverage item is seen, D' is reduced. Recall that in the case of statistical testing we needed to assume that the analogous event, observation of a fault, resulted in perfect or productive fault repair, in order to ensure that D' was not increased.

The statistical results allow us to start with an initial coverage set X, or with no previously covered items (i.e. X is empty), and then determine, as we go, when it is unlikely that we will see an item not in X, or not in an earlier part of the test sequence. We would then carry out a minimal set of tests that covers all of X and any new coverage items seen in the test sequence.

Statistical coverage testing retains some of the important advantages of coverage based testing in the following way. Suppose that the problem in statistical testing with large numbers of tests is the expense of validating large numbers of subsequent test outputs. In statistical coverage, we may have to run a large number of tests, as in statistical testing, but, as in coverage based testing, it is only necessary to validate output for a test set that covers each of the coverage elements. In statistical coverage we may have to run many tests to confirm that new coverage is unlikely, but it is only necessary to record and compare coverage for the tests. We could then use a minimal covering subset of these tests for output validation.

The use of an initial coverage set X allows us to first test for critical cases to make sure they are covered, as in traditional coverage, and then go on to a statistical phase. Since the critical cases coverage is included in the set of "already seen" coverage items, it contributes to the overall reliability measurement effort.

Statistical coverage may also retain the non-statistical coverage testing advantage that facilitates the use of cheap tests. If a test is only expensive due to the cost of its output validation, and if the input domain subsets for coverage elements contain both cheap and expensive test cases, then we can restrict the output validation phase of statistical coverage to cheap tests.

The fact that it is only necessary to validate output for a covering set of tests, and that during reliability estimation it is sufficient to record and compare coverage, opens up the possibility of establishing more extreme reliability results than might be possible for ordinary statistical testing. This is because it will normally be possible to automatically record and compare coverage, even for compound coverage models. In statistical testing, corresponding extreme levels will only be possible if system output can be and as easily and inexpensively automatically validated.

The statistical coverage approach that has been described here is a paper approach in the sense that it has not been used on a real system. The basic theory has been worked out, and some of the results such as the adaptive confidence and parallelization theorems are solutions to important problems in both statistical testing and statistical coverage. But it would be reassuring to see at least some evidence of the idea's practicality. The following example reviews a testing project that was described in [18]. It can be interpreted as an informal application of the method, and an existence proof of its applicability.

4.1 Rule based systems example

The example involves a rule based system for monitoring system states. In this example it was necessary to test a system that could be in an astronomically large number of possible states. The states correspond to combinations of different rules that could occur when the rule oriented system is executed, and can be viewed as abstract compound coverage states. The goal was to cover these states in some way. It was impossible to consider a traditional coverage approach in which all of the states, or even some fraction such as .9, were covered.

The approach used in the monitor testing project was to first run the system and observe the set Y of actual states that occurred over a relatively long period of time, i.e. over a large set of tests T. After this was done, a set of potential tests for which correct behavior was known was examined to find a minimal covering subset, i.e. a minimal set of tests that covered all of the states in Y. This minimal set was then used for actual verification testing. The approach made it possible to informally confirm the reliability of the system, with respect to rule combination state coverage, using a relatively small number of output validations.

The first phase in the monitor example, the generation of Y, corresponds to the phase of statistical coverage where random testing is carried out until it is unlikely that any more coverage items (i.e. states) will be seen. In this example, there is no initial set X. We informally observe that after a certain point, new coverage items appear infrequently. The second phase in the monitor example corresponds to the phase of statistical coverage where we validate output for a test subset T' that covers all the coverage items that have been seen.

The monitor example describes an informal approach to reliability estimation. The confidence based coverage methods described in this paper could be used to compute formal coverage based reliability figures for this kind of application.

5. Functional and non-functional programs and systems

In the above discussion, both for statistical testing and statistical coverage, systems and programs were viewed as input/output functions where we choose random input cases, and then look at output or other kinds of behavior. For some kinds of systems this is inappropriate. These will be referred to as non-functional systems, in which persistent or static data is retained from one use of the system to the next, and becomes part of its input. In this case, the system uses both the persistent data and "new" input data for each computation. These will be referred to as the internal and the external input data.

Several possible approaches to systems with internal and external data can be suggested. One possible approach involves the use of the corollaries to Theorem 6, domain decomposition. Suppose that E is the set of all external inputs to a system, and that I is the set of possible internal data sets, so that the total input space for each use of a system is formed from the cross product D = IxE. In these corollaries we assume that we can decompose I into subsets $I_{i,}$ $1 \leq i \leq K_,$ for which we know the distribution of internal and external data. In one case (Corollary 6.3) we assume that we know a joint distribution for internal and external data for each individual subset. In another (Corollary 6.4) we assume that there is an independent distribution for the external data, and we can decompose the internal data into subsets over which internal data cases are uniformly distributed. Other corrolaries can be developed for alternative models of the relationship between internal and external input data.

6. Related work

The author previously presented a less formal, preliminary version of some of this work that contained the basic idea of termination of testing when new coverage is statistically unlikely, in [19]. Once the approach had been developed to its current point, several connections with other work, including the above rule based example could be seen. Additional theory and examples can be found in an expanded version of the paper [20].

Other related work includes the testing and reliability results in [21], which use the concept of "useful testing effort". In their paper the authors define useful effort as that which increases coverage with respect to some coverage measure. This can be compared to the idea in statistical coverage that we discontinue testing when new coverage is unlikely.

Related work also includes attempts to characterize the effectiveness of testing and analysis methods, and to develop a framework for predicting the reliability of a program based on the effectiveness of the methods used to evaluate it. One approach involves estimates of the *detectabilty* of different methods. Detectability is an empirical estimate of the probability that a method will detect a fault of some kind, given that the fault is present. Suppose we have a collection of methods M_i, $1 \leq i \leq m$, a collection of fault classes F_j, $1 \leq j \leq n$, estimates of the probability f_j that a program will contain faults from class F_j, and estimates of the detectability $D_{i,j}$ for methods M_i and fault classes F_j. Then we can construct a simple min-max formula that gives the probability P of there being a residual fault in a system from one of these fault classes, after the set of methods M_i has been applied, if such a fault were present. The complement 1-P of this number might be used as a measure of confidence in the absence of such faults, and is similar to the definition of software *trustability* used in [22].

There is also a large body of related work associated with reliability growth modeling [e.g. 23-25]. This paper has been restricted to a discussion of confidence based statistical methods. Some of the issues that have been studied in reliability growth modeling that are relevant including the problem of modeling a system's operational distribution. Additional related work on operational distributions is described in [8], [26] and [27]. Different kinds of operational distribution models for complex systems are closely related to the models used to describe complex input domains like those in the corollaries to Theorem 6.

7. Summary and conclusions

The basic theoretical results for confidence based statistical testing and test data coverage were reviewed and several additional results presented including the adaptive sampling theorem (3), the parallelization theorems (4,5), and the corollaries to Theorem 6 that describe possible methods for applying the approach to non-functional systems containing internal input data.

It was pointed out that if output validation is expensive there is a potential practical problem for the application of statistical testing. A suggested approach is to use a new method called statistical coverage measurement. In this approach, testing is halted when it is unlikely that no new coverage elements will be seen. When this strategy is followed, it is only necessary to validate output for tests that cover all of the coverage elements for a coverage model. Reliability can be established while running tests for which the only requirement is that we note if a new coverage item has occurred. The use of coverage methods was equated with the use of a program fault model, and reliability was interpreted with respect to such a model.

The use of the parallelization theorems, along with statistical coverage modeling, raises the possibility of establishing stronger reliability results for a program. Parallelization allows the results of separate reliability experiments to be combined. Statistical coverage eliminates the need for output validation during the statistical phase of a testing process. It also has the advantage that it allows the easy incorporation of non-statistical testing efforts into the reliability computation effort.

The compound coverage models that were discussed indicate that statistical coverage will be useful at the systems level where we want to cover program functionality associated with interactions between different parts of a system, such as communication/state interaction in distributed programs, object property interactions in abstract state systems, or rule combination in rule oriented systems.

The proponents of growth modeling sometimes criticize confidence based approaches because they base reliability on the probability of a program failing, or a new coverage item occurring, on the next execution rather than predicting some future period of expected system behavior. But confidence based reliability for a program continues to hold for the second and subsequent executions of a program as long as we continue to see no failures or new coverage items. In fact, confidence will slowly increase with the number of successful runs. If a failure or new coverage item occurs, then we need to use the results of Theorem 2 or Theorem 3 to establish a new confidence level.

Although the work in this paper is primarily of a theoretical nature, its results, such as adaptive sampling and parallelization, at the very least help to characterize the underlying principals of statistical testing and test coverage, and may also help to facilitate its use in appropriate situations. The monitor rules testing application from [16], for example, indicates the potential usefulness of the confidence based model to coverage based testing.

References

[1] Duran, J. W. and Ntafos, S.C., An evaluation of random testing, *IEEE Transactions on Software Engineering*, 10-4, July, 1984.

[2]Hamlet. R. and Taylor, R. Partition analysis does not inspire confidence, *IEEE Transactions on Software Engineering*, SE-16, 12, December, 1990.

[3]Howden, W.E., *Functional Program Testing and Analysis*, McGraw-Hill, 1987.

[4] Poore, J.H., Mills, H.D., Mutchler, D., Planning and certifying software system reliability, *IEEE Software*, January, 1993.

[5] Thayer, T.A., Lipow, M., and Nelson, E., *Software Reliability*, North Holland, 1978.

[6] Tsoukalas, M.Z., Duran, J.W. and Ntafos, S.C., On some reliability estimation problems in random and partition testing, *IEEE Transactions on Software Engineering*, 19-7, July, 1993.

[7] Voas, J.M. and Miller, K., Improving the software development process using testability research, *Proceedings, Third ISSRE*, October, 1992.

[8] Voit D.M, Estimating software reliability with hypothesis testing, *Proceedings, Fifth ISSRE*, Monterey, IEEE, 1994.

[9] Howden, W.E., Software trustability, theory and practice, CSE Technical report, UCSD, 1995.

[10] Hamlet, R., Foundations of software testing: dependability theory, *Proceedings SIGSOFT '94*, ACM, New Orleans, 1994.

[11] Myers, G.J., *The Art of Software Testing*, Wiley, New York, 1979.

[12] Ostrand, T.J., and Balcer, M.J., The category partition method for specifying and generating functional tests, *CACM*, 31-6, June 1988.

[13] Marick, B. *The Craft of Software Testing*, Prentice Hall, New Jersey, 1995.

[14] Frankl P.G. and Weyuker, E.J., An applicable family of data flow testing criteria, *IEEE Transactions on Software Engineering*, 14-10, October, 1988.

[16] Taylor, R. N., Levine, and D.L. Kelly, C.D., Structural testing of concurrent programs, *IEEE Transactions on Software Engineering*, 18-3, March, 1992.

[15] Woodward M.R., Hedley, D. and Hennell, M.A., Experience with path analysis and testing of programs, *IEEE Transactions on Software Engineering*, 6, May, 1980.

[17] Koppol, P.V., Tai K.C., An incremental approach to structural testing of concurrent software, *Proceedings 1996 ISSTA*, ACM, San Diego, 1996.

[18] Avritzer, A. Ros, J.P and Weyuker, E.J., Reliability of rule-based systems, *IEEE Software*, 13-5, September, 1996.

[19] Howden, W.E., Auditing the use of informal testing and analysis methods, *Proceedings, Asia Pacific Conference on Software Engineering*, IEEE, Seoul, 1996.

[20] Howden. W.E., Confidence measurement and its application to statistical testing and test coverage, CSE Technical Report, UCSD, La Jolla, CA, 1997.

[21] Horgan, J.R. and Mathur, A.P., Software testing and reliability, in ed. Lyu, M.R., *Software Reliability Engineering*, McGraw Hill, 1996.

[22] Howden. W.E. and Huang, Yudong, Software trustability, *ACM Transactions on Software Engineering and Methodology*, January, 1995.

[23] Goel, A. and Okumoto, K., Time dependent error detection rate model for software reliability and other performance measures, *IEEE Transactions on Reliability*, R-28(3), 1979.

[24] LaPrie, J.C., Kanoun, K., Béounes, C. and Kaâniche, M. The KAT (Knowledge-Action-Transformation) approach to modeling and evaluation of reliability and availability growth, *IEEE Transactions on Software Engineering*, 17-4, April 1991.

[25] Musa, John, Iannino, A. Okumoto, K, *Software Reliability*, McGraw Hill, New York, 1990.

[26] Whittaker, J. and Thomason, M., A Markov chain model for statistical software testing, *IEEE Transactions in Software Engineering*, 20-10, October, 1994.

[27] Whittaker J, and Poore, J.H., Markov analysis of software specifications, *ACM Transactions on Software Engineering and Methodology*, 2-1, Jan. 1993.

0-8186-8120-9/97 $10.00 © 1997 IEEE