**3 USAGE-BASED STATISTICAL TESTING WITH MUSA’S OPERATIONAL**

**PROFILES**

One important testing technique, the usage-based statistical testing with Musa’s operational profiles (OPs) (Musa, 1993; Musa, 1998), shares the basic model with partition testing techniques, and enhances it to include probabilistic usage information.

**3.1 The cases for usage-based statistical testing**

The many sub-domains for large software systems may include many different operations for each sub-domain. In such situations, the equivalence relation as represented by partition testing represents a uniform sampling of one test point from each sub-domain. However, if operations associated with one particular sub-domain are used more often than others, each underlying defect related to this sub-domain is also more likely to cause more problems to users than problems associated with other sub-domains.

This likelihood for problems to customers, or related system failures defined accordingly, is captured in software product ***reliability. R***eliability is defined to be the probability of failure-free operations for a specific time period or a specific input set (Musa et al., 1987; Lyu, 1995a; Tian, 1998).

The best way to assess and ensure product reliability during testing is to test the software product as if it is used by customers through the following steps:

* The information related to usage scenarios, patterns, and related usage frequency by target customers and users needs to be collected.
* The information collected above needs to be analyzed and organized into some models- what we call operational profiles (OPs) - for use in testing. Testing needs to be performed in accordance with the OPs.
* Testing results can be analyzed to assess product reliability and provide feedback to the testing and the general software development process.

Therefore, we concentrated on the information collection, OP construction, and it usage in testing.

Like most test activities, the actual testing is typically performed late in the overall product development process, and the model construction could be and should be started much earlier. Usage-based statistical testing actually pushes both these activities to the extremes at both ends as compared with most other testing techniques. On the one hand, the operational profiles (OPs) need to be constructed with customer and user input. It makes more sense to start them right at the requirement analysis phase, or even earlier, in the product planning and market assessment phase. On the other hand, testing according to customer usage scenarios and frequencies captured in OPs cannot be performed until most of the product components have been implemented. Therefore, such OP-based testing could only be performed in the very late sub-phases of testing, such as in the integration, system, or acceptance testing sub-phases.

**3.2 Musa OP: Basic ideas**

According to Musa (Musa, 1993; Musa, 1998), an operational profile, or an OP for short, is a list of disjoint set of operations and their associated probabilities of occurrence. It is a quantitative characterization of the way a software system is or will be used. As a simple example, consider the usage of **www.seas.smu.edu,** the official web site for the School of Engineering (which used to be called School of Engineering and Applied Science, or SEAS) of Southern Methodist University (SMU/SEAS). **Table1** gives the OP for this site, or the number of requests for different types of files by web users over 26 days and the related probabilities.

**Table1** Usage frequencies (hits) and probabilities (% of total) for different file types for SMUlSEAS

****

The “operations” represented in the operational profiles are usually associated with multiple possible test cases or multiple runs. Therefore, we typically assume that each “operation” in an OP can be tested through multiple runs without repeating the exact execution under exactly the same environment. In a sense, each operation corresponds to an individual sub-domain in domain partitions, thus representing a whole equivalence class.

In this example, each item in the table, or each operation, represents a type of file requested by a web user, instead of individual web pages. Of course, we could represent each web page as an operation, but it would be at a much finer granularity. When the granularity is too fine, the statistical testing ideas may not be as applicable, because repeated testing may end up repeating a lot of the same test runs, which adds little to test effectiveness. In addition, such fine-granularity OPs would be too large to be practical. For example, the number of individual web pages on an average web site would be more than tens of thousands, while the number of file types is usually limited to a hundred or so, including many variations of the same type, such as HTML files with extensions of “.HTML”, “.html”, “.htm”, etc. There are more than 11,000 web pages but only about a hundred file types for SMWSEAS. There are also several other key points worth noting about such OPs, including:

* It is customary to sort the operations by descending probabilities of usage and present the results in that order.
* It is common to have quite uneven distribution of usage probabilities, with a few frequently used ones account for most of the usage frequencies. For example, the top **13** out of a hundred or so files types account for more than 98% of the web hits or individual requests for SMU/SEAS.
* Related to the uneven distribution of usage probabilities is the probability threshold for individual operations. The basis of statistical testing is to perform more testing for those operations that are used more by the customers. Therefore, if some operations have very low probability of usage, we could omit them in the OP. This probability threshold plays an important role in limiting the numbers of operations to represent in the OP, especially when there are a large number of possible operations.
* The representation in forms similar to **Table1** is called tabular representation in literature, which can often be represented visually as a histogram, such as in **Figure1**. The use of such histograms has several advantages, primary in the direct and intuitive information about the relative value and magnitude of the different usage probabilities associated with different operations.



**Figure1 An** operational profile (OP) of requested file types for the **SMUISEAS** web site

Consequently, an OP can be considered to be a checklist or a partition, but with frequency or probability of usage associated with each item in the list or with each sub-domain in the partition.

**3.3 Using OPs for statistical testing and other purposes**

Once an OP is constructed, it can be used to support statistical testing by some random sampling procedure to select test cases according to the probability distribution and execute them. Essentially, each operation in the OP corresponds to certain test cases specifically constructed or selected from existing ones to specifically test this system operation. The allocation of test cases is also affected by the associated probability. For legacy products, there may be more test cases than we can use for some operations. Therefore, the probabilities for individual operations can be used to select some existing test cases while screening out others. Consequently, these probabilities determine the minimal number of test cases that need to be constructed or selected for each operation.

The actual test runs are sampled from these test cases according to the probability of associated operations. Therefore, the number of test runs for each operation in the OP is proportional to its probability. Under most circumstances, these test cases and associated runs can be prepared ahead of time; so that some test procedure can be employed to sequence the multiple test runs according to various criteria. In some cases, truly dynamic random sampling can be used, to dynamically select test cases to run next.

However, such dynamic random sampling will slow down test execution and the system performance because of the overhead involved in managing the test case selection in addition to monitoring the test runs. On the other hand, much of testing is aimed at monitoring the system performance and possible problems under “normal” operational conditions, particularly for usage-based statistical testing of common usage scenarios carried out under the environment that resembles target customers’. We would like to reduce such overhead and get a truly representative setting of in-field operations. Therefore, unless absolutely necessary, we should prepare the test cases and test procedures ahead of time to reduce the impact of testing overhead on normal system operations.

In addition to or in place of proportional sampling, progressive testing is often used with the help of OPs. For example, at the beginning of testing, higher probability threshold can be used to select a few very important or highly used operations for testing. As testing progresses, the threshold can be lowered to allow testing of less frequently used operations so that a wide variety of different operations can be covered. In a sense, the use of OPs can help us prioritize and organize our testing effort so that important or highly used areas are tested first, and other areas are progressively tested to ensure good coverage.

The use of OPs for statistical testing enables us to make of quantitative reliability assessment and ensure product reliability that is meaningful to target customers. Failure and test execution data from OP-guided testing should resemble the data that would be expected by target customers. It is for this reason that data from such statistical testing can be analyzed using various reliability models to assess product reliability, to determine if the product is reliable enough for release. If the reliability goal has not been achieved yet, the analysis results can often help us predict time and resource to reach the reliability goal, and sometimes to identify problematic areas for focused improvement actions. Besides the above use of OPs to help us prioritize testing and assure product reliability, OPs can also be used in other situations for various purposes, including:

* ***Productivity improvement and schedule gains*** could be achieved because of the focus on high-leverage parts with the use of OPs. The use of OPs would reduce the over testing of lesser-used product components or functions: The same effort could be applied to most-used parts, with reduced effort for lesser-used parts. According to data from AT&T (Musa, 1993), a reduction of 56% system testing cost or 11.5% overall cost could be achieved.
* ***Introducing new products*** by implementing highly used features quickly to capture market share and then gradually adding lesser-used features in subsequent releases or product updates. OP-guided testing would make this approach work. This use of OP is similar to and can be combined with spiral development process (Boehm, 1988), or the use of software prototypes to resolve important product design questions before proceeding with other implementation activities.
* ***Better communications with customers and better customer relations:*** The use of OPs can foster better communications with target customers and help establish and maintain better customer relations. There are several reasons for these, including:

- The construction of operational profiles needs the direct or indirect involvement of customers. Customer’s perspectives of quality, product features, relative importance, and possible usage scenarios and associated problems are adopted in testing when OPs are used. Therefore, it’s more likely for the software development organization and the customers to appreciate each other’s views and efforts in such a collaborative instead of an adversarial environment.

- The use of OPs can help develop more precise product requirements and specifications, which are more likely to satisfy customers’ needs.

- Customer and user training can be better focused on those features that the customers use the most instead of esoteric ones more likely to be used by internal users.

* ***High return*** *on* ***investment:*** According to data from AT&T (Musa, 1993), the OP development cost for an “average” product is about one staff-month. The average product is one that contains 100 KLOC (thousand lines of code) and takes 10 developers 18 months to finish. There is also a sub-linear increase of OP development cost for larger products. The cost-benefit ratio is reported to be about 1:10

Because of these tangible and intangible benefits, OPs should always be a prime candidate for testing in large software system with many users or with diverse usage environments.