Implementing Operational Profiles to Measure System Reliability

B.D. Juhlin
AT&T Bell Laboratories
11900 North Pecos Street
Denver, Colorado 80234

Abstract

This paper presents a method devised for creating Operational Profiles (OP's) by "decomposing" an OP into two components: a Configuration Profile (CP) and a Usage Profile (UP). This decomposition is necessary in order to reduce the broad variation in customer usage into a set of test environments (OP's) that can be implemented as part of a commercial product development/test process. These test environments must, of course, emulate actual customers closely enough so that the data gathered from the testing can be used to predict the reliability observed in the field. This paper describes the decomposition of an Operational Profile into its components and provides examples from the project in which this method was first used (an AT&T International Private Branch Exchange development).

Keywords: software system reliability, operational profiles, software test

1. Introduction

Measuring the reliability of a software-based system requires:

- metrics - measures of reliability that correlate with customer satisfaction
- models - mathematical/statistical methods for metric prediction and data smoothing
- operational profiles - descriptions of expected product usage
- lab methods - techniques for executing the Operational Profile and collecting the data needed by the models

Though metric definition and model fitting is very important, the foundation of Software Reliability Engineering (SRE) is Operational Profiles. No matter how sophisticated the metric and modeling techniques used, they're only as good as the data that serves as input to them. That data, in turn, is only as good as the methods used to generate it. For SRE, those are the methods that define and implement the Operational Profile in a development test lab.

An "ideal" operational profile method would be to:

- Set up EVERY customer's system in the test lab (all at once),
- use each system exactly like those customers,
- count failures and track usage,
- and compute and model the resulting metrics.

Continued testing of this type (with fault correction) would eventually lead to very high reliability as seen by the product's customers. The obvious weakness in this approach is simply that it is not practical for two basic reasons:

1. It is difficult (expensive? impossible?) to know that much about every present and future customer.

2. Even if the information were available, it would generally be much too expensive to implement such an approach (especially if the number of customers is large and/or their product usage is complex).

The goal then, is to find a practical way to assess reliability for present and future customers without implementing every individual customer's environment. In other words, the goal is to emulate h customers' usage (where h may be large) with o test environments (where o is generally much smaller than h due to test resource constraints). Each of these o emulations (a test environment or Operational Profile) represents many of the h customers. All o Operational Profiles taken together would represent all h customers.

Reducing the h customers' environments to o Operational Profiles means that completely accurate portrayals of every customer's environment are sacrificed to provide for practical test environments. The "trick" in this reduction is to characterize and implement the Operational Profiles (the "reduced" environments) with only the most important or significant parameters. The
"trick" in that characterization is to select only those parameters that significantly affect the product's reliability (i.e., its failure rate).

This paper will describe a process to implement operational profiles. The process was first used in the development of an AT&T Private Branch Exchange (PBX) in 1989 and has been continually improved since that time. Though designed for an AT&T PBX product, the process can be generalized to other products. This type of approach is most useful when the product's customers, usage, and configurations are varied and complex, and setting up a single, representative test environment is impractical.

2. Decomposing an Operational Profile

For AT&T PBX's, we have found that customers' use of the product varies across two basic dimensions:

1. How they set up their system in terms of hardware and connectivity (the configuration)
2. How they use the product's features and capabilities (the usage)

It is also apparent that our customers exhibit the same usage across different configurations, so that configurations and usage are "mixed and matched". Therefore, we decompose an Operational Profile into two components: a Configuration Profile (CP) and a Usage Profile (UP). These two components can then be combined as desired to create an Operational Profile (OP).

2.1 Configuration Profile

The way in which customers set up their system is a Configuration Profile (CP):

A Configuration Profile defines the physical and logical configuration that makes up an Operational Profile. This is the environment in which one or more Usage Profiles may take place.

Figure 1 illustrates that h customer configurations (CUSTh) are reduced to m Configuration Profiles (CPm). The subscript h represents the current or expected number of customers in the field and the subscript m is the number of CP's to be implemented in the test lab. For practical reasons, such as limited lab and staff resources, m is much smaller than h.

Each CPm in Figure 1 may also have multiple deltas (Di). Deltas represent modular changes to the CPm that are "slight departures" from the basic CPm. Deltas are loosely defined to be: "enough of a change to the CP to effectively represent a different OP, yet not a large enough change to be treated as an entirely separate CP.

Figure 1. Customer Configurations Reduced To Configuration Profiles

The difference between a CP and a delta is mostly related to the magnitude of the effort required to implement them. A Configuration Profile then, is either a basic CPm alone, or a CPm in combination with one or more Di.

PBX Configuration Profile parameters include such things as:
- Public Network Connectivity (e.g., Central Office interfaces)
- Private Network Connectivity (e.g., Network Type, Number and Type of Nodes)
- Duplication (e.g., Processor Duplication)
- Adjuncts (e.g., voice mail)
- Dial Plan and Routing Patterns

CPm for PBX's include such configurations as a standalone system, a main/tandem network, or a distributed systems cluster (see the example in Figure 2). Each CPm is assigned a probability that reflects its market/field proportion (the CPm/Di combinations may also be assigned probabilities).

2.2 Usage Profile

The way in which customers use their system is a Usage Profile:

A Usage Profile defines the uses, tasks, and activities of the product by its users including the rate, volume, and/or frequency of that usage.

A Usage Profile (UP) is described by:
- A list of users that together define that Usage Profile (a list defines a customer type).
- The respective proportions of each user in a customer type (each user belongs to a user type and their proportions form a user mix).
An example of a simplified usage profile is shown in Table 1.

<table>
<thead>
<tr>
<th>User Name</th>
<th>Generic User Type</th>
<th>User Percentage</th>
<th>Call Rate</th>
</tr>
</thead>
<tbody>
<tr>
<td>secretary</td>
<td>secretary</td>
<td>20</td>
<td>10</td>
</tr>
<tr>
<td>attendant</td>
<td>attendant</td>
<td>1</td>
<td>24</td>
</tr>
<tr>
<td>agent</td>
<td>acd-agent</td>
<td>1</td>
<td>15</td>
</tr>
<tr>
<td>sysadm</td>
<td>acd-supervisor</td>
<td>1</td>
<td>3</td>
</tr>
<tr>
<td>executive</td>
<td>manager</td>
<td>3</td>
<td>3</td>
</tr>
<tr>
<td>office workers</td>
<td>worker</td>
<td>68</td>
<td>4</td>
</tr>
<tr>
<td>in counselors</td>
<td>administrators</td>
<td>6</td>
<td>5</td>
</tr>
<tr>
<td>system administrator</td>
<td>sysadm</td>
<td>-</td>
<td>-</td>
</tr>
</tbody>
</table>

**User Descriptions:**
- Secretary:
  - Multi-Function Digital Set with Display
  - Primary interactions are with Manager user types
  - Extended feature set (Bridging, consulting, intercom)
  - Call coverage goes to other secretaries, then the attendant
- Attendant:
  -

**TABLE 1. Simplified Example Of A Usage Profile**

Other UPm for PBX's include factories and hotels with users such as factory workers and hotel front desk personnel. Each UPm (like that in Table 1) is assigned a probability that reflects its market/field proportion.

2.3 Operational Profiles

An Operational Profile (OP), then, is:

A description of product usage created by combining customer configurations (CP's) and customer usage (UP's).

The combination of a CP with a compatible UP creates an OP. That is, a CP in combination with a UP creates a description of some real, though "generic", customer's environment (see Figure 4). Each CP typically has multiple UP's associated with it. [It may be the case that not all CP's are compatible with all UP's. This could occur for functional product reasons (e.g., a certain CP does not support a certain UP) or because that combination is known to not occur in the field].

Each of the o OP's account for some proportion of the customer population (h). This proportion could be viewed as the probability of that OP occurring.

Each actual customer (represented by the CUSTh in Figure 4) uniquely links (by definition) a configuration than h.
with a usage one for one. When the OP decomposition step is complete, the $m$ CP's and the $n$ UP's are no longer linked one for one. In its place are a set of CP's, UP's, and the potential to combine them in many ways to create OP's.

3. Acquiring Operational Profile Information

Operational profile based testing depends on having appropriate and accurate information to define the profiles. At AT&T, various groups, including systems engineering, field support, and system test, gather this information through such mechanisms as:

- Customer Modeling
- Field Studies
- Market Studies

This information is summarized into several frequency/probability distributions as illustrated in Figure 5.

Each distribution consists of the item, descriptive information about the item, and its probability of occurrence:

- **Configuration Distribution** -- a distribution of the system configurations. Our approach has been to keep this list quite short (i.e., keep $m$ small) and use deltas ($D_i$) as needed to reflect differences seen in the field among the "basic" configurations. Additional descriptive information here includes such things as trunk types, network connectivity, and processor configurations (as illustrated in Figure 2).

- **Customer Type Distribution** -- a distribution of "generic" customer types such as banks, hospitals, and the like. Descriptive information includes such things as types of features used, traffic rates and traffic mix (see Table 1 for an example).

- **User Mix Distribution** -- for any given customer type, many different users and their relative usage proportions (i.e., their mix) are possible. This distribution indicates which are most/least likely. Additional information includes user descriptions and, for each user, usage rates and typical features used (see Table 1 for an example). Also, some user's activities are described by call trees as outlined in Section 4.3.

- **CP to UP Distribution** -- this information lists the allowable combinations of CP's and UP's that can form OP's. Each combination has a probability and all combinations together form a distribution of OP's.

4. Creating Operational Profiles

Once the CP's and UP's have been defined, the process for creating and executing a specific OP is outlined in Figure 6 and defined in the following subsections. This process may be followed once, or many times, for a single product. The choice is based on whether it is possible/practical to define and implement a single all-encompassing Operational Profile. If not, multiple OP's are necessary, each representing some fraction of the overall OP (i.e., the overall customer base).

Factors that indicate the need for multiple Operational Profiles include:

- Widely varying customer configurations -- no single [feasible] configuration would accurately emulate all customers' configurations.
Highly varied customer usage patterns -- in which a single OP would "wash out" any actual customer's usage and thus not be truly representative.

Limited test equipment/resources -- so that it's not possible to set up one "grand" OP.

When multiple OP's are required and cannot all be set up simultaneously, OP evaluation (testing) must take place sequentially in time. This implies of course, that as fault correction takes place, the software is changing and thus requires some degree of regression testing. Clearly then, the cost of setting up a "grand OP" must be balanced against the cost of multiple OP's and regression testing. We have chosen the latter approach and have developed strategies for overcoming the regression testing costs. These strategies include defining "generic" UP's from "generic" user descriptions in order to provide a broad range of usage in each UP. (The downside of the "generic" approach is that it implies a sacrifice of some amount of realism to attain broader coverage). Another important strategy is the heavy emphasis at AT&T placed on test automation. This allows regression test sets to be implemented repeatedly at [relatively] low cost.

The process outlined in Figure 6 and described in the following subsections is based on a situation in which multiple Operational Profiles are required. Figure 6 describes the process followed for each OP implemented.

4.1 Selecting Customer Type, Configuration, and Users

The process of creating a specific Operational Profile begins with selecting from the configuration, customer type, and user type distributions described in Section 3. The configuration and customer type choices may be dependent or independent (as indicated by the dotted line in Figure 6). For example, it may be the case that certain customer types tend to use only certain specific configurations. In our experience though, these selections have been independent since [nearly] all combinations exist in our customer base (or at least possible).

There are several strategies that can be followed when making the selections, including:

1. Randomly select from the usage probability distributions -- This is the best strategy from a statistical sampling point of view. It yields the most accurate reliability metrics and models and thus provides the best assessment of the overall reliability (quality) of the product. However, it may not be optimal for quality improvement in a commercial product development environment when other factors like knowledge of product failure behavior, first customer's needs, and (especially) limited time and testing resources are considered.

2. Deliberately select the highest probability items from the probability distributions -- This strategy deliberately (not randomly) selects items to test in order (highest probability items first) according to their usage probabilities. This is a good strategy from the test resource perspective. By following this strategy, the largest proportion of customers are covered with the fewest selections. When no (or little) knowledge of the product's failure behavior is available, this strategy is an optimal way of improving software quality since it examines the items most often used by the customer. This approach also may allow early controlled release of the product to the "most common" customers, especially if they represent the simpler configurations and usages.

3. Deliberately select the "high probability" and "most likely to fail" items -- this strategy extends strategy 2 by adding in knowledge of product failure behavior. (Though perhaps difficult to quantify, knowledge of product failure behavior is often quite available during the development
process. It includes such things as what code has changed, what circuit packs are new, what functional interactions exist, what the "traditional" trouble areas are, and so on. This knowledge can guide the selection strategy.] Strategy 2 can be used initially to gather the product failure behavior information needed to implement this more "directed" strategy. This strategy tends to optimize software quality improvement by creating a test strategy that selects items based on the combination of usage and failure likelihood. It can also be implemented by making random selections from a probability distribution constructed by combining usage and failure probabilities.

4. Deliberately select the "most likely to fail" items from the probability distributions -- This has often been a "classic" approach to software product testing (i.e., the "try to break it" approach). Though it does indeed optimize fault exposure (on which many traditional fault-based metrics depend), that is not the same as improving software quality since it does not account for customer usage (i.e., it does not necessarily find the problems that are most important to the customer). This strategy will not give an accurate assessment of the product's reliability and hence makes it difficult to determine when the product is ready for release.

Our approach to selecting customer types, configurations, and users has been primarily that of strategy 3. We first select operational profiles of high usage probability (the ones most likely to occur in the field) and then select parameter values and deltas \( \Delta_i \) that have higher expected failure likelihoods. This choice of strategy is driven by the need to optimize testing resource use and to reduce test intervals.

4.2 Developing The Configuration Profile

Once a configuration is selected, the process continues with further Configuration Profile definition. It is typically necessary at this step to take the source information and add detail. This includes making lower-level configuration choices, translating the customer description into actual lab setups (based on equipment availability, etc.), defining the \( \Delta_i \) as appropriate, and documenting the detailed information needed to set up the lab.

In order to support the implementation of multiple CP's without having to completely rebuild the test lab each time, a Base Configuration (BC) is defined. The BC describes physical and related logical items that form a "base test bed." It is the "lowest common denominator" across all CP's. It reflects the intersection of creating "true" customer configurations with lab realities (limited equipment, staffing and time). The BC helps reduce the number and magnitude of test lab changes required to implement the set of OP's.

Also in this step, deltas \( \Delta_i \) for the CP must be defined. Recall that deltas represent modular changes to the CP that are "slight departures" from the basic CP. Deltas of interest might include such things as:

- common differences from the CP observed in the field
- "traditionally" error prone product areas related to the CP selected
- new product features related to the CP selected
- different sets of hardware (supporting, for example, different countries)

Note that defining and choosing the \( \Delta_i \) is one way to implement the focused selection strategy (strategy 3) described in Section 4.1

4.3 Creating The Usage Profile

In this step, descriptions of customer usage are developed and implemented. We define a unit of customer usage to be a "user scenario" which is simply a transaction oriented description of system usage. Scenarios for PBX's cover such users as system administrators, maintenance technicians, telecom managers, telephone users, and the like. Scenarios for all users except telephone users are created manually by system testers "acting out" the role of that user.

A more structured and automated approach has been developed for PBX telephone users (who make up the overwhelming majority of system usage). This approach begins by defining a group of "generic users" (like those listed in Table 1). Each "generic user" represents multiple, similar actual users.

Each generic telephone user's activity is then described by a call tree. For example, Figure 7 is a simplified version of a generic manager's call tree. The call tree shows external outgoing calls (extout), external incoming calls (extinc), outgoing networked calls (netout), incoming networked calls (netinc), internal outgoing calls (intout), and internal incoming calls (intinc). The call tree also shows typical features and operations such as abbreviated dialing (abbrev), standard dialing (std.d), answering a call (ans), putting someone on hold (hold), and conferencing a call (conf).

A set of tools are used to automatically create scenarios from the call trees. This method substantially automates the generation of the Usage Profile portion of an Operational Profile. An added benefit has been
greatly enhanced tester productivity since the vast majority of scenarios/test cases are automatically produced by this method. Figure 8 shows an example of two simplified trees yielding a simple scenario.

In addition to creating scenarios from the call trees, tools have been developed to create random samples of scenarios. Using the frequencies defined in the call trees, these tools generate samples that are proportioned according to customer usage (strategy 1 in Section 4.1). Consequently, the failure rate of the samples represents the failure rate users would expect to see as they use the system (see Section 4.5). The sampling schemes used for creating user scenario random samples follow the same basic strategies as those for selecting from the customer distributions (as outlined in Section 4.1).

4.4 Executing The Operational Profile

Executing the Operational Profile follows this basic process:

1. Set up the system into a customer-like configuration (the CP) using the same methods and instructions as the customer.
2. Execute the scenarios (the UP) on the configuration (the CP).
3. Implement any deltas as appropriate (again, in a "customer-like" way).
4. Execute additional scenarios on the configuration with the delta.

Though step 2 forms the majority of the test time, step 1 is essential for detecting failures in the setup/administration process. Setting up a CP means implementing the BC (once for all CP's) and adding the additional items that completely define a CP.

The scenarios generated from the previous step (described in Section 4.3) are implemented in two basic ways:

1. The simpler, high-volume scenarios are implemented via automated traffic generation tools.
2. More complex scenarios are executed by system testers.
As the OP is executed, a variety of reliability data is gathered including:

- CPU Occupancy
- Traffic loads
- Call failures from traffic loads
- Manual user scenario failures
- Serious system failures

Data is also gathered for quality areas other than reliability -- like maintainability and fault tolerance.

4.5 Computing the Reliability Metrics

From the data gathered during the execution step, we compute these reliability metrics:

- **Serious Failure Rate (SFR)** -- the number of serious failures (e.g., high percentage of stations out of service, system restarts, etc.) divided by the number of CPU hours executed for the Operational Profile. This rate is then converted to a customer-oriented rate using a CPU to Customer Hours conversion factor.

- **Call Failure Rate (CFR)** -- the number of calls that fail (e.g., no dial tone, no talk path, etc.) divided by the total number of calls generated.

- **User Failure Rate (UFR)** -- the failure rate of a sample of scenarios generated from the call tree. This metric represents the probability that users will experience a failed transaction.

5. Results

We began developing this technique about three years ago for use on an international release of an AT&T PBX. For that release, we created a complete Operational Profile for each set of customer information we acquired. The results across those 12 Operational Profiles are pictured in the reliability metric graphs below.

Each of the data points in Figures 10, 11, and 12 represents the results of testing an Operational Profile during one of three intervals:

- **PRE-ST** -- Operational Profile based product quality assessments are performed prior to the official start of System Test (i.e., while the product is still in development).

- **SYSTEM TEST** -- The official system test interval consists of continued Operational Profile based testing.

- **CI** -- The Controlled Introduction interval is the point at which the product begins to be released to the field.

All of the graphs represent quality improvement as a function of time during product development. The data displayed for the CI interval are from the system testing process, not from the field data collected during CI. The "Musa Basic Reliability Model" [1] was used for both Serious Failure Rate and User Failure Rate (log plot yields a straight line for UFR in Figure 12). Note that the predetermined CI goals for all three of these metrics were met before the CI was allowed to start, but because of the effectiveness of the OP testing these goals were met with a very short system test interval.

Figure 10 illustrates how the Serious Failure Rate (SFR) metric was tracked during system test and the CI interval. Each of the rectangular blocks represents the observed failure rate for an Operational Profile. Failure rates above the goal required immediate developer response to isolate and repair the defects that were causing the high failure rate. Note that the majority of defects were found and fixed during the "PRE-ST" interval (i.e., before the start of system test). Also note that the serious failure rate during customer operation has to date been zero as predicted.

As illustrated by this data, the Operational Profile method of reliability testing and assessment was quite successful (other factors also influenced these results -- see [2]). Despite its success, the method in its initial form proved to be somewhat cumbersome to implement (each new OP had to be developed "from scratch") and not nearly flexible enough for the dynamic global marketplace in which we are competing. These observations led to development of the OP decomposition technique described in this paper.

We began applying the OP decomposition technique late in the development/test interval of the product depicted in the reliability graphs. Since that time, we have implemented the technique more fully in subsequent product releases. We now have OP's derived from four basic CP's and 10 basic customer types that use 22 different generic user types. The combinations of these items (along with dozens of CP deltas) provide us with hundreds of different Operational Profiles from which to choose. In addition, the call tree scenario approach provides us with a population of millions of user scenarios from which to sample.
Figure 10. Serious Failure Rate

Figure 11. Call Failure Rate

Figure 12. User Failure Rate
6. Conclusion and Future Work

This paper has presented a technique for implementing Operational Profiles. The technique involves decomposing an operational profile into two basic components: a configuration profile and a usage profile. These components are then "re-combined" using appropriate selection and/or sampling strategies.

Though first used as part of the development process for an AT&T PBX, the Operational Profile decomposition approach can be generalized to any product. It is a "practical" approach to implementing an "ideal" test environment (the Operational Profile). It is most useful (and has the most benefit) when the product's customers, usage, and configurations are varied and complex, and setting up a single, representative test environment is impractical.

Future work on this approach is being done in the following areas:

- Automated execution of the Usage Profile.
- Customer data gathering -- including better ways of classifying and categorizing customer configurations and usage.
- CP/UP selection techniques -- especially sequential or focused techniques (ones that develop subsequent selections/samples based on the results of previous samples).
- User scenario sampling techniques.
- Metric and modeling strategies for dealing with biased selection/sampling schemes.

At AT&T, we are committed to customer satisfaction. Operational Profile based testing/reliability assessment is proving to be an important development technique for achieving the high levels of reliability demanded by our customers.

Acknowledgements

I am grateful to my supervisor, Bliss Jensen, whose vision has guided much of this work. I also wish to thank all of my colleagues in Bliss' group who have contributed so much to make this "theoretical approach" to testing real, and above all, successful in a commercial product development environment.

References
