Robust Testing of AT&T PMX/StarMAIL Using OATS

Robert Brownlie
James Prowse
Mahdav S. Phadke

Robust Testing is a systematic test method that uses the OATS (Orthogonal Array Testing System) tool to convert information from product requirements or specifications into a concise set of tests. We found Robust Testing more than twice as productive as conventional testing practices for testing PMX/StarMAIL Release 2.2, a complex private mail exchange (PMX) local area network (LAN)-based electronic mail product that works with several types and versions of LAN software, operating systems, and personal computers (PCs). Robust Testing required less time, enabling us to complete testing on schedule. It found more potentially user-affecting faults, and inspired greater confidence in the quality of the delivered product.

Introduction

Testing takes up a significant proportion of development resources. But faults are still discovered in the field, often causing customer dissatisfaction, increased field repair costs, and a perception of poor product quality. Thus, the need to improve testing effectiveness is clear.

Robust Testing, adapted from a broader engineering design method called Robust Design, is a systematic, analytical approach that saves time and improves fault detection. Originated by Genichi Taguchi in Japan, the Robust Design method has been used extensively to improve manufacturing process and hardware product design.[1-2]

Robust Testing uses the special properties of orthogonal arrays (OA) to design efficient experiments. When OA are used in Robust Testing, as proposed by Mandl[3] and Tatsumi,[4] those properties efficiently cover the test domain. This paper's purpose is to describe the practical results and benefits of Robust Testing. Panel 2 presents details of OA properties, and offers a simple application example.

Robust Testing requires a thorough understanding of the tested product's architecture and design requirements, as well as the conditions under which the product will be used. These must be analyzed to define the test domain in terms of test parameters (i.e., attributes of the tested product) and the possible levels of each parameter. The OATS (Orthogonal Array Testing System) software tool—developed by A. P. Chintapalli, S. S. Hegde, and M. S. Phadke—uses those parameters and levels to select an appropriate OA and automatically generate the test cases for the product. P. E. Brown developed an algorithm that allows the large orthogonal arrays needed in some special cases to be generated.

- It reduces testing time because fewer test cases are needed.
- It improves the ability to find faults because the resulting test cases exhibit superior functional and feature coverage.

This paper describes a case study of PMX/StarMAIL system testing. System Test is the last formal test step before the software is released. It ensures that the software conforms to the formal design requirements, is defect-free, and is functional when used in customer environments and configurations. Clearly, the thoroughness of testing directly affects customer perceptions of product quality. The fewer problems customers have, the greater their satisfaction with the product.

Testing Needs

This section describes the details of the testing assignment based on product requirements and the market deadline realities of product development. We demonstrate

Panel 1. Terms and Acronyms in This Paper

CPU	central processor unit
LAN	local area network
OA	orthogonal array
OATS	Orthogonal Array Testing System
PC	personal computer

that the testing assignment has grown too large for the available testing resources. Note that the testing needs refer to the hardware and software products in existence at the time of testing, March 1990.

Product Requirements. Figure 1 represents the PMX/StarMAIL product. Our System Test was performed to certify changes made to the PMX/StarMAIL product since the previous release. Changes included in PMX/StarMAIL Release 2.2 focused on supporting additional configurations of AT&T's StarGROUP LAN software. As Panel 3 shows, the combinations of supported client and LAN server types requiring testing increased a factor of nine. In addition, PMX/StarMAIL had to be tested on the three main Intel processors, the 8086, 80286, and 80386 because MS-DOS® version 4.01 being released after the earlier release of PMX/StarMAIL. (MS-DOS is a registered trademark of Microsoft Corporation.) Compatibility of the new StarGROUP client types with the prior release of PMX/StarMAIL (applicable only to the prior StarGROUP server release known as MSNET) also required testing. Testing PMX/StarMAIL with every combination of these factors would have required testing 72 configurations.

To support the new configurations of the AT&T StarGROUP LAN Manager software, modifications had to be made to both the UNIX® system-based PMX/StarMAIL software and the MS-DOS client-based user applications. (UNIX is a registered trademark of UNIX System Laboratories, Inc.) No enhancements or changes were made to the PMX/StarMAIL software to add new functions. Thus, the programs would appear unchanged from the user's perspective, and tests of the product's functions could depend on the test cases used to test the previous release.

Realities of the Test Cycle. Several factors affected the PMX/StarMAIL test cycle.

- PMX/StarMAIL was a critical link in a tightly scheduled multiple product development effort where only eight weeks were scheduled for testing.

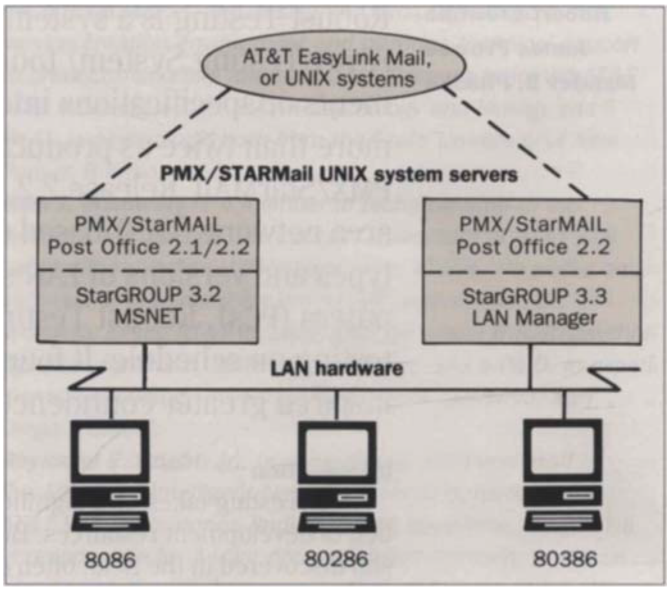
AT&T EasyLink Mail, or UNIX systems
PMX/STARMail UNIX system servers
PMX/StarMAIL Post Office 2.1/2.2
StarGROUP 3.2 MSNET
PMX/StarMAIL Post Office 2.2
StarGROUP 3.3 LAN Manager
LAN hardware
8086
80286
80386

Figure 1. MS-DOS PCs running PMX/StarMAIL user interface. AT&T PMX/StarMAIL is an electronic mail system for users of MS-DOS-based PCs connected via a Local Area Networks (LAN). Specifically, AT&T StarLAN hardware and AT&T StarGROUP software is represented here. The PC-based applications provide the user a screen based interface (either Microsoft Windows™-based or character screen based) for creating, editing, reading and delivering mail messages. The UNIX system-based applications provide intermediate storage and mail messages transport between users. Mail can be sent and received between users on the LAN and users on other UNIX host systems, as well as AT&T EasyLink Services.

- Only one staff person was assigned to testing.
- Other complex products were built on PMX/StarMAIL, and required a stable basis for development and testing.

The challenge was to use available resources to test PMX/StarMAIL to ensure stable performance in all configurations while meeting the deadlines.

The following sections will describe and compare two test plans that were considered for testing PMX/StarMAIL.

1. A test plan, based on the conventional approach, to test the product functions in select configurations.
2. The test plan, based on tests defined by OATS, that was used.

Panel 2. Orthogonal Array-based Test Cases

Consider a function to be tested with four parameters: A, B, C, and D. Suppose each parameter has three possible levels as given in Table I. This parameter-level table specifies the test domain consisting of the 81 possible combinations of the test parameter levels. Similarly, Table III provides an example of parameters and levels used for defining PMX/StarMAIL tests.

Table II shows an orthogonal array (OA) called L9. It has nine rows and four columns. (The OA name refers to the number of rows.) The rows correspond to test cases; the columns correspond to the test parameters. Thus, Test Case 1 consists of level 1 for each parameter: i.e., it represents the combination (A1, B1, C1, D1). Case 2 consists of the combination A1, B2, C2, D2, etc. An orthogonal array has the balancing property that, for each pair of columns, all parameter-level combinations occur consistently. In the orthogonal array L9, there are nine parameter-level combinations for each pair of columns, and each combination occurs once. For example, in the columns of parameters C and D, the combinations (1,1), (1,2), (1,3), (2,1), (2,2), (2,3), (3,1), (3,2), and (3,3) occur in test numbers 1, 6, 8, 9, 2, 4, 5, 7, and 3 respectively. Table IV presents the identical pattern for examples of PMX/StarMAIL tests generated by OATS. It would be easy to verify for these orthogonal arrays that for each pair of columns all combinations occur exactly once.

By conducting the nine tests indicated by L9, we can detect:

- Any consistent problem with any level of any single parameter.
- Any consistent problem with pairwise compatibility of parameters.

Orthogonal array-based test cases cannot prove that a product will work for all possible parameter-level combinations (e.g., three or more parameter-level combinations). But by virtue of their pairwise balancing property, they give excellent coverage of the entire test domain defined by the test parameters.

The example L9 (Table IV) array provided for PMX/StarMAIL's *copy* function is a simplification of the tests. PMX/StarMAIL's copy function allows the user to copy mail messages between separate folders (MS-DOS directories) to save them for future reference. The *function scope* parameter refers to the three copy options available: copy only a single (current) message; copy messages previously marked, or copy *all* messages contained within a folder. The *Target Folder Content* parameter defines a range of possible conditions for the content of the folder to which messages are copied. An empty folder and a full folder define limits or boundaries of the range, while a partial folder represents any number of messages between these boundaries.

The StarGROUP client and server parameters are explained in Figure 1. Tests defined for the copy function combined the remaining four configuration parameters (described in the subsection "Defining Test Parameters and Levels") to generate an OA of 18 rows (L18). OATS produced five L16, two L18, and seven L27 arrays for testing 13 additional functions.

The OATS tool allows automatic selection and fitting of an orthogonal array to suit a wide variety of applications. See Taguchi[5] and Phadke[1] for a discussion of orthogonal array fitting methods.

The Conventional Test Plan

The first attempt to define a test plan focused on testing product functions in a representative set of hardware and software configurations. This approach has its merits, because when a software function is broken, it usually will not function, whatever its configuration. It is imperative to detect and correct faults of this type before software is released to customers. This reasoning was followed to test prior releases of PMX/StarMAIL. Hereafter, we refer to this as the *conventional* test plan.

Because PMX/StarMAIL function tests were available from prior testing, the next step would be to identify the appropriate configurations. The combinations of StarGROUP clients and servers (Panel 3) gave the basis for selection. The shaded areas represents the client and server combinations that would have been used to test the product functions in the hypothetical conventional scenario. These combinations allow testing of every client and server type, and satisfy the testing requirements defined by the modifications to PMX/StarMAIL Release 2.2. The additional requirements to test MS-DOS system versions and Intel central

Panel 3. Summary of AT&T StarGROUP client and server interoperability with PMX/StarMAIL Release 2.2

Clients	Servers		
		LAN Manager	
	MSNET	Share mode	User mode
MSNET	▨		not supported
Basic		▨	
Enhanced			▨

This panel summarizes PMX/StarMAIL's interoperability with the different client and server versions of AT&T StarGROUP software. For any combination of client and server version (any cell in the panel), PMX/StarMAIL should work or at least send an appropriate error message to the user if it can not perform a specific function. As an example, the combination of MSNET client and user mode LAN Manager server is not supported for any PMX/StarMAIL function, and should always send error messages.

As discussed in the text, the shaded area represents the combinations that would have been tested in detail using the conventional test plan.

The AT&T StarGROUP LAN Manager for the UNIX system is developed from licensed source code developed by IBM for its OS/2 operating system. Like Microsoft's LAN Manager for OS/2, the UNIX system-based LAN Manager Server provides two separate and configurable operational environments, the *user* and *share* security modes. Each server security mode constitutes a distinctly different environment for administering and using the server. In other words, the facilities for file and printer sharing provided by each of these security modes are used differently by both users and application programs. In turn, both LAN Manager security modes are operationally different from the prior StarGROUP versions referred to as MSNET. Similarly, AT&T StarGROUP LAN Manager for UNIX offers two MS-DOS system client types: the basic client, with a restricted set of commands to allow server logon and access to file and print resources; and a special MS-DOS system version of the enhanced client that allows administration of the server. These two client types differ completely from the MSNET client provided in the earlier release of StarGROUP software.

processor unit (CPU) architectures would be examined by random assignment to the selected client types. Greater emphasis would be placed on previously untested factors, such as the then new MS-DOS 4.01 and the Intel 80386 processor-based systems. (Note: later tests have certified PMX/StarMAIL with MS-DOS 5.0, and with the Intel 80486 CPU). The remaining client-server combinations (the unshaded boxes in Panel 3)—and compatibility tests of the prior release of PMX/StarMAIL—would be tested with a limited subset of the functional tests to ensure that the major functions worked.

Based on testing experience with PMX/StarMAIL, 450 critical tests of the affected product functions were identified. It was estimated that testing in the three primary environments described above would require 18 staff-weeks of effort. Test schedule extensions were unacceptable. Even with an additional tester, the eight weeks scheduled for testing would allow only 1,000 tests to be performed, rather than the 1,500 that would have been required. Therefore, additional selection and reduction of the number of tests would be needed. The net result would have been an increased likelihood of missing latent software faults resulting in lower confidence in the thoroughness of the planned testing.

In light of these concerns, and the fact that the Robung method had just become available, the conventional test plan described above was never executed.

The OATS-Based Test Plan

This section describes the test plan that was put into use. Specifically, it presents the method by which test parameters and levels were defined, and then briefly describes how the Orthogonal Array Testing System generated test cases based on the defined test parameters.

Defining Test Parameters and Levels. Before we could use OATS to generate specific test cases, all product functions, test data, and hardware and software configurations for PMX/StarMAIL specified by our testing requirements had to be expressed in terms of parameters and levels.

Table I. Sample Set Of Parameters And Levels

Test Parameter	Levels 1	2	3
A	A1	A2	A3
B	B1	B2	B3
C	C1	C2	C3
D	D1	D2	D3

Table II. L9 Orthogonal Array

Test Case Number	Test Parameters A	B	C	D
1	1	1	1	1
2	1	2	2	2
3	1	3	3	3
4	2	1	2	3
5	2	2	3	1
6	2	3	1	2
7	3	1	3	2
8	3	2	1	3
9	3	3	2	1

Table III: Example of Parameters And Levels for PMX/StarMAIL Copy Function Testing

Test Parameter	Levels 1	2	3
StarGROUP Server	MSNET	User Mode	Share Mode
StarGROUP Client	MSNET	Basic	Enhanced
Function Scope	Copy Current	Copy Marked	Copy All
Target Folder Content	Empty	Partial	Full

First, the hardware and software configurations were considered. The specific parameters and parameter levels were easy to identify from our requirements. As discussed above, they consisted of:

- The StarGROUP server types (MSNET server, share mode server, and user mode server).
- The StarGROUP client types (MSNET client, basic client, and enhanced client).
- The PMX/StarMAIL release (Releases 2.2 and 2.1).
- The MS-DOS system version of the client PC (Release 4.01 and Release 3.3).
- The Intel CPU type (8086, 80286, and 80386).

Next, the testable functions of PMX/StarMAIL were identified and redefined in terms of parameters and levels. The starting point was the cases used for the prior release testing. The result was 18 tables of parameters and levels, including a definition of mail messages for test data.

Table IV. L9 Orthogonal Array of PMX/StarMAIL Test Cases

Test Case Number	Test Parameters StarGROUP Server	StarGROUP Client	Function Scope	Target Content
1	MSNET	Basic	Copy Current	Empty Folder
2	MSNET	Enhanced	Copy Marked	Partial Folder
3	MSNET	MSNET	Copy All	Full Folder
4	Share Mode	Basic	Copy All	Partial Folder
5	Share Mode	Enhanced	Copy Current	Full Folder
6	Share Mode	MSNET	Copy Marked	Empty Folder
7	User Mode	Basic	Copy Marked	Full Folder
8	User Mode	Enhanced	Copy All	Empty Folder
9	User Mode	MSNET	Copy Current	Partial Folder

Using OATS to Generate Test Cases. Because PMX/StarMAIL Release 2.2 requirements focused on modified code to support new hardware and software, testing a product function had to produce the same results regardless of hardware and software configurations. Therefore, we reasoned that an orthogonal array could be constructed to combine parameters defined for the test configurations with parameters defined to test a specific function. The resulting array would define a set of tests that systematically evaluated the effects of all parameters relevant to the testing requirements. Panel 2 shows a simplified example to prove the validity of this approach.

Table V. Productivity Comparisons

Testing method	Proportion of testing faults found (P)	Testing effort expended (E)	Relative testing productivity (P/E)
Robust Testing	100%	8.0 staff-weeks	12.5
Conventional Testing	78%	16.0 staff-weeks	4.9
Productivity Ratios (Robust Testing/Conventional)	1.3	0.5	2.6

Of the 422 tests used to test PMX/StarMAIL Release 2.2, 354 (i.e., 84 percent) were generated by OATS. Another 68 test cases defined special conditions or tests not applicable to all configuration parameters, e.g., installing the software on the UNIX system. The 422 tests compared favorably with the estimated 1,000 tests for our conventional test plan. The conclusion was that less testing could be performed with more thorough coverage of test configurations and product functions.

Quantitative Results

This section presents a discussion of the results achieved by using OATS. They are presented in terms of timely completion and efficiency of fault detection, expressed as a productivity ratio of faults per unit of testing effort. In addition, we present some tentative conclusions about the effectiveness of PMX/StarMAIL in the field, based on our use of Robust Testing before the software was released.

Test Results. Using the Robust Testing strategy described above, the tester completed the 422 test cases on schedule, in 8 weeks, uncovering several serious software faults. After testing, and after extracting and analyzing the fault listings from the change control database, we determined that 12 percent of the faults found by Robust Testing were unlikely to have been found using conventional testing methods. For example, an unsupported client-server combination that could have been inadvertently configured by the customer would result in a misleading error message. Time pressures often limit conventional testing of such unsupported configurations in favor of more thorough testing of supported configurations.

By contrast, the conventional testing method, because of the limited testing interval, would have required two testers (or 16 staff-weeks). But as noted, even with the additional person, only 1,000 of the 1,500 planned conventional test cases could have been done in that time. We estimated conservatively that limited testing would have missed another 10 percent of the faults. Thus, 10 percent + 12 = 22 percent *fewer* faults would have been found (i.e., 78 percent of all test-detected faults) had we used a time-limited conventional test suite.

Productivity Comparisons. Testing productivity was defined as *the number of faults found per unit of testing effort expended* (i.e., faults per staff-week). Table V shows a comparison of Robust Testing productivity with that of the conventional testing method. The Robust Testing results provided the actual data for comparison, and included all faults found during the system testing interval.

The relative testing productivity ratio of 2.6 in the lower right-hand corner of Table V shows that Robust Testing was more than twice as productive as conventional testing for this application.

Additional benefits that were observed included:

- Test plan reviews are simpler because only parameters and levels must be considered.
- If parameters and levels change, test plans and test cases are easily revised.
- Requirements reviews can easily include test-case generation, which may reveal missing error conditions and messages.

Field Results. In the time customers have been using PMX/StarMAIL Release 2.2 (at this writing, about a year and a half), three additional faults have been found in the field, all of them outside the scope of the planned testing. One resulted from a defect in the underlying software upon which PMX/StarMAIL's functionality depends. The remaining two were caused by deficiencies in the product requirements. Such problems would be less likely if Robust Testing were routinely applied to *all* stages of a sequentially developed product, not only System Test.

Conclusions

We successfully used Robust Testing to system test PMX/StarMAIL Release 2.2 in an environment with tight schedules and many hardware and software combinations. We estimated results from the hypothetical conventional test scenario to permit a realistic comparison with the actual results of Robust Testing, enabling us to prove Robust Testing's productivity advantage over our conventional testing methods. The result was a shorter testing interval, lower staffing needs, improved quality, and higher confidence in the coverage of requirements.

To obtain the most concise set of tests, we learned to combine configuration and functional parameters in the same array. The properties of orthogonal arrays provide systematic testing of the product's functions relative to usable configurations, resulting in higher customer confidence. In cases where only one or two test parameters are defined, OATS suggests testing all combinations.

Robust Testing and OATS continue to be used to test PMX/StarMAIL and related products. Our experience suggests that Robust Testing should be used wherever there are many independent hardware, software, or functional parameters to test. Robust Testing would also benefit earlier development phases, such as requirements reviews, design reviews, unit testing and integration testing. We concluded that, for the best results, Robust Testing and OATS should be used to focus and leverage—but not *replace*—the judgment of an experienced, trained test engineer, working from stable requirements, who also is familiar with the product's applications.

As product complexity escalates and customers demand increased reliability, test engineers continuously seek to increase their productivity. In response to this challenge, Robust Testing techniques and OATS are becoming more widely accepted throughout AT&T Bell Laboratories, and should be added to every tester's toolkit.

References

1. Mahdav S. Phadke, *Quality Engineering Using Robust Design*, Prentice-Hall, Englewood Cliffs, New Jersey, 1989.
2. G. Taguchi, *Introduction to Quality Engineering: Designing Quality Into Products and Processes*, Asian Productivity Organization, Tokyo, 1986; American distribution by UNIPUB/Kraus International Publications, New York, 1986.
3. Robert Mandl, "Orthogonal Latin Squares: An Application of Experiment Design to Compiler Testing," *Communications of the ACM*, October 1985, Vol. 28, No. 10, pp. 1054-1058.
4. Keizo Tatsumi, *Test Case Design Support System*, ICQC 1987.
5. G. Taguchi, *Jikken Keikakuho*, 3rd Edition. Tokyo, Japan: Maruzen, Vol. 1 and 2, 1977 and 1978 (in Japanese). English translation: G. Taguchi, *System of Experimental Design*, ed. Don Clausing, Vols. 1 and 2, UNIPUB/Kraus International Publications, New York, 1987.

Robert Brownlie *is distinguished member of technical staff in the Software Process Architecture and Engineering Department of AT&T Bell Laboratories, Red Hill, New Jersey. He works in the Operations Systems Process Planning and Implementation Group on software quality assurance process engineering, process data collection, process measurement, and analysis systems. He joined AT&T in 1977 with a B.S.E.E. from Iowa State University, Ames, and an M.S.E.E. from Purdue University, West Lafayette, Indiana.*

James Prowse *is a member of technical staff at AT&T Bell Laboratories, Lincroft, New Jersey, where he works in the EasyLink Premises Products and Quality Assurance department. He is responsible for all planning, setup, test execution, and reporting results for system test of AT&T's PMX/StarMAIL system. He joined AT&T in 1986 with a B.S. in psychology from the State University of New York College at Cortland.*

Madhav S. Phadke *was a supervisor in AT&T Bell Laboratories from 1977 to 1991, where he pioneered the development and application of the Robust Design method in the U.S.A., and received the Taguchi Award in 1985. He is currently President of Phadke Associates, Inc., Tinton Falls, New Jersey, a company specializing in quality technology deployment training and consultation, especially in the Robust Design and Taguchi Method, economic production quality systems, and other methods of quality and productivity improvement for software and hardware projects. He wrote the first textbook in English on the Taguchi Method,* Quality Engineering Using Robust Design, *published in 1989 by Prentice-Hall. Mr. Phadke received a B.S. in mechanical engineering from the Indian Institute of Technology, Bombay; an M.S. in mechanical and aerospace sciences from the University of Rochester, New York; and an M.S. in statistics and Ph.D. in mechanical engineering from the University of Wisconsin, Madison.*

(Manuscript received October 21, 1991)