To the Graduate Council:

I am submitting herewith a thesis written by Michael P. Muir entitled "Case Study: Statistical Testing of the Material Safety Datasheet System". I have examined the final copy of this thesis for form and content and recommend that it be accepted in partial fullfillment of the requirements for the degree of Master of Science, with a major in Computer Science.

Jesse H. Poore, Major Professor

We have read this thesis
And recommend its acceptance:

Accepted for the Council:

Associate Vice Chancellor and
Dean of The Graduate School

CASE STUDY: STATISTICAL TESTING OF THE MATERIAL SAFETY DATASHEET SYSTEM

A Thesis

Presented for the

Master of Science Degree

The University of Tennessee, Knoxville

Michael Patrick Muir

December 1997

Abstract

This thesis is a case study of statistical usage testing. Usage testing is a method in which the software usage is modeled and test cases are generated based on this model. As demonstrated by the Adams [1][4] data, and Mills' analysis of this data [2], usage testing has a 30 to 1 advantage over code-coverage testing in reducing mean time to failure. Model-based testing also produces many statistics based on the model and test results, one of which is an estimate of reliability, something impossible to achieve with code-coverage testing. Model-based testing relies on the development of a specification-based software usage model. Since this model is driven by the software specifications, it may be developed before or in parallel with the software. A usage model was developed for the Material Safety Datasheet System (MSDS) and analysis of the model led to change in the specification and test plan. Test cases were generated from the model and used to certify the system.

Table of Contents

1. Introduction

1.1 Material Safety Datasheet System (MSDS)

The Material Safety Datasheet System (MSDS) is an information management system designed to provide OSHA required information to all employees operating in the Department of Energy (DOE) Oak Ridge, Portsmouth, and Paducah facilities, as it pertains to the handling of hazardous material. The MSDS system is an interactive, windows based, multi-user, client/server application designed for the input, maintenance, and tracking of vendor furnished Material Safety Datasheets (MSDs). The MSDS was developed using Microsoft Visual Basic 4.0 and the Oracle relational database management system (RDBMS).

1.2 Statistical Testing of Software Based on a Usage Model

Usage testing refers to the practice of model-based testing, in which the software usage is modeled, and test cases are generated based on this model. Coverage testing refers to the concept of testing every line of code and branch paths. As demonstrated by the Adams data [1][4] and Mills' analysis of this data [2][4], usage testing has a 30 to 1 advantage over coverage testing. Model-based testing also produces many statistics based on the model and test results, one of which is a reliability estimate, something impossible to calculate from coverage testing.

Poore, Walton, Trammell [3], and Whittaker, Thomason [6] have described the theory and practices of treating software testing as a stochastic process and modeling usage by finite state, discrete parameter, time homogeneous, irreducible Markov chains. The Markov chain is used to model the software specifications and represents the complete capabilities of the system from the user's point of view. Probabilities assigned to each transition reflect the actual use of the system. Using this method, a model representing the usage environment of the system may be defined. From this model, a unique understanding of the complexity of the interface and of the testing requirements may be ascertained. Testing based on this model will enable software certification and the determination of a statistical reliability estimate.

As will be demonstrated by this case study, model-based testing is a feasible and cost effective process for information systems operated through a graphical user interface (GUI). The techniques described provide a software development team with a tremendous insight into the complexity of an application. Three items of information readily available from a model of an information system are:

- complexity of the interface
- relative importance (amount of use) of specific modules behind the interface
- test resource allocation

Model development, when performed as a disciplined engineering process, may be accomplished within the same resources as necessary for traditional test plan development. Since all the information necessary for the usage model is derived from the specifications, these benefits may be realized early in the development process, avoiding potential rework.

The Material Safety Datasheet (MSDS) system was developed using traditional development techniques. No specification or development engineering practices such as those defined in [4] were utilized for development.

1.3 Project Summary

The project history shows the value of model-based testing. In the MSDS case study, the initial estimate for the project based on current development techniques (no consideration was given to usage model testing during initial estimates) was 15 man-months, to be allocated and tracked as shown in Table 1.1.

Table 1.1: Estimated Effort Allocation

Phase	Estimated Effort	Actual Effort
Design	1	1
Code Development	12	11
Model Development	N/A	2
Developer Testing	1	.5
Systems Testing	1	2

Development of the usage model required 2 man-months of effort. This expenditure included the learning associated with model development. Model building revealed an overly complex design for the MSDS and led to

several changes of the initial specifications for the interface. These changes resulted in a simplification of the design by modifying the specifications to reflect a more modal system. The initial specification called for a mode-less multiple document interface (MDI) design. Upon analysis, it was determined that by adding some modality to the interface, the testing time and costs would be greatly reduced.

The time required for certification (system testing) was initially estimated as follows. We wanted to run 230 test cases without a failure in order to demonstrate a reliability of .99 with 90% confidence[5], as required by the binomial. Allowing for four failures at the worst possible times would require approximately 5x230=1150 test cases. Preliminary testing showed that approximately 50 test cases could be run in one day. Thus an estimate of 23 days for certification was made.

After an excessive number of failures (eight) were recorded, testing was suspended and the system was turned back over to the developers. A model arc coverage test set was generated and utilized by the development team. Certification testing was not to resume until this test set was completed successfully. Upon completion of the arc coverage test set, certification testing continued and the system was certified. Even though our actual time spent on testing was over the initial estimate, the reliability achieved and design changes made to the interface were worth the additional cost.

It should also be noted that the actual development effort completed 1 man-month (MM) under estimated effort. This is attributed to the changes made to the specifications to simplify the interface based upon the complexity analysis, made possible by the usage model.

The overall result was that the MSDS system was delivered on budget with only a slight delay in time. This initial release of the system (version 1) had no reported failures during its lifetime. The production period for version 1 was 3 months and consisted of a user base of 11 employees utilizing the software for daily operations.

A second version of the system was developed. For experimentation purposes it was decided that traditional testing practices would be utilized for this very minor release. The developers and environment for this version were identical to that of version 1. This release passed the traditional, user created, test plan with no failures. Version 2 was a very minor release and testing was completed in 2 days. This release was installed in the field and a failure was logged during the first week of use. As a follow up, the MSDS model was updated for version 2 and test cases were generated. It was determined that the 11^{th} test case generated would have revealed this design error. Using the data from the first version on time spent executing test cases, this error would have been found during the second hour of statistical testing.

1.4 Overview and Outline

The data maintenance function of the MSDS, described in the previous section, is an example of a Management Information Systems (MIS) application. Several applications of statistical testing have been documented for embedded real-time, graphical user interface (GUI), and application program interface (API) applications, but little work has been performed to date in applying these techniques to MIS data management applications. Embedded real-time and APIs, normally have much simpler interfaces with

few input types and relatively few transitions between states. MIS applications may require a complex interface with a larger number of input types. This thesis discusses development of a usage model for a GUI based MIS application, analysis of the model, and certification testing based on that model. The software product toolSET_Certify © from Software Engineering Technology, Inc. (SET) was utilized for model development, analysis, test case generation, and test monitoring.

Chapter 2 gives a brief overview of the application to be developed and the operating and development environments.

Chapter 3 presents a brief overview of the principles behind statistical testing. This chapter introduces data and analysis used to justify statistical testing. An overview of the technique for developing a usage model is presented and guidelines for testing discussed.

Chapter 4 details the actual design of the model for the MSDS system. This chapter briefly describes some early experiences during the initial efforts at modeling the system. It describes decisions that were made based upon those initial efforts, and describes the standards and conventions adopted. Guidelines used to assign probabilities are discussed, results of the model analysis are given, and test planning is presented.

Chapter 5 details the results of the testing activities. The test results are documented and analyzed. The concept of model coverage is discussed.

Chapter 6 summarizes the results and provides the conclusions and lessons learned from the study. An overall evaluation of the process is then given.

2. MSDS Application

2.1 System Definition

The Material Safety Datasheet System (MSDS) is an information management system designed to provide OSHA required information to all employees operating in the Department of Energy (DOE) Oak Ridge, Portsmouth, and Paducah facilities, as it pertains to the handling of hazardous material (MSDS is also the name of the managing office and program). The MSDS is composed of two major components: the MSDS Information Management Module (IMM), and the Electronic Access Module (EAM).

The MSDS IMM provides Hazardous Material Management Program (HMMP) personnel with a means of data management for all information necessary for the creation, maintenance, and tracking of hazardous materials and their manufacturer-supplied material safety datasheets, and for distribution of this information to all employees. The EAM is a distribution mechanism for this information and provides employees of Energy Systems electronic access to all hazardous material information and the corresponding datasheets. The IMM portion of this system is the module that was used for this study. The IMM function is supplied by a set of forms that provide a mechanism for input, maintenance, and querying of hazard

related data and the ability to index and scan vendor supplied material safety datasheets.

2.2 Objectives

The objectives of the MSDS IMM are to ensure that hazard information and the manufacturer's material safety datasheets are readily available to all Energy Systems personnel and contractors located on or near an Energy Systems managed site and that this data is efficiently and accurately maintained.

2.3 General Requirements

The system must provide a "user-friendly" on-line environment for authorized users to maintain and control the data through the use of menus, input screens, and help screens. The system utilizes current client/server technologies. The client portion of the system operates under the Microsoft Windows95 environment and conforms to the standard Windows95 style guidelines. The application was designed as a Windows Multiple Document Interface (MDI) type of application that will permit the user to display and manipulate information on multiple forms simultaneously. As will be discussed later, this gives the potential for a mode-less design which yields an extremely complex set of interface possibilities for testing.

2.4 Environment

The client portion of the IMM was developed using Microsoft Visual Basic and Oracle SQL*Net. The data is stored in an Oracle7 relational database management system (RDBMS) residing on a Digital Equipment Corporation (DEC) Alpha/2100. The operating system for the DEC 2100 is OpenVMS.

Real-time access to the data must be maintained between the IMM and EAM portions of the system. All development was in Visual Basic and C.

2.5 Size

The size of the system is as follows. The Visual Basic system is composed of 16 forms, 12 Oracle database tables, and 17,646 lines of code. The development effort was 2 developers, 6 months of effort. All members of the project team possessed a minimum of one year of experience with Visual Basic, and at least 3 months of experience with the existing business rules for handling the material safety data sheets.

3. Testing Software based on Usage Models

3.1 Failure rates of errors

Adams [1] developed a model to characterize the occurrence of software failures in the field based on a large collection of historical data. The basis of Adams' model is that design errors in software each possess a characteristic problem rate R. The occurrence of each design error is an event in a Poisson process in which errors occur at random times with an average rate R. The cumulative running time for all users (denoted in thousands of months) plays the role of time in the Poisson process. The error rate of the software product is the sum of the error rates associated with each design error. Adams analyzed data from 9 large software products and fitted these rate distributions to his model. Table 3.1 depicts these rate distributions.

Table 3.1: Error Rate Distributions for 9 Projects

	Mean Time to Failure (MTTF) in K months							
Product	60	19	6	1.9	0.6	0.19	0.06	0.019
1	34.2	28.8	17.8	10.3	5.0	2.1	1.2	0.7
2	34.2	28.0	18.2	9.7	4.5	3.2	1.5	0.7
3	33.7	28.5	18.0	8.7	6.5	2.8	1.4	0.4
4	34.2	28.5	18.7	11.9	4.4	2.0	0.3	0.1
5	34.2	28.5	18.4	9.4	4.4	2.9	1.4	0.7
6	32.0	28.2	20.1	11.5	5.0	2.1	0.8	0.3
7	34.0	28.5	18.5	9.9	4.5	2.7	1.4	0.6
8	31.9	27.1	18.4	11.1	6.5	2.7	1.4	1.1
9	31.2	27.6	20.4	12.8	5.6	1.9	0.5	0.0
Mean	33.2	28.2	18.7	10.6	5.2	2.5	1.1	0.5

The error rate distributions, shown in line 1 of Table 3.1, are very similar from product to product and reflect how error frequencies clump together. Also, most of the design errors present have mean times to discovery of thousands of months and a few of the errors have a short mean time to failure (MTTF). It is easy to see that with such ranges in failure rates, coverage testing will find the very low failure rate errors 1/3 of the time with very little effect on the MTTF. Usage testing should more accurately reflect field use and find many more of the high failure rate errors with much greater effect on MTTF.

Mills' analysis of the previous data is as follows. Table 3.2 depicts error densities and failure densities in the MTTF classes of Adams' data.

Table 3.2: Error and Failure Densities in MTTF Classes

1	M	60	19	6	1.9	.6	.19	.06	.019
2	ED	33.2	28.2	18.7	10.6	5.2	2.5	1.1	.5
3	ED/M	.6	1.5	3.1	5.6	8.7	13.2	18.3	26.3
4	FD	.8	2.0	3.9	7.3	11.1	17.1	23.6	34.2
5	FD/M	0	0	1	4	18	90	393	1800

Line 1 denotes the MTTF (**M**) of the class as discovered by Adams. Line 2 denotes the error density (**ED**) as defined by the average error rate distributions of the nine products. If **1/M** = failure rate of the class and **ED** = the probability that a member of this class will be found next using code-coverage testing, line 3 then becomes the contribution of each class, on average, in reducing the failure rate by fixing the next error found by code-coverage testing. Normalize line 3 and you obtain the failure density (**FD**) (line 4). Line 5 then becomes the contribution of each class by fixing the

next error found by usage testing. Mills discovered that the sums of the two lines, ED/M and FD/M turn out to be proportional to the decrease in failure rate from the fixes of errors found by coverage testing and usage testing, respectively [2]. Their sums are 77.3 and 2306, with a ratio of about 30 between them. The major finding of this data is that locating and correcting errors by usage testing increases MTTF by a factor of 30 over code-coverage testing.

3.2 Usage Model

Statistical testing of software based on a software usage model ensures that the failures that will occur most frequently in operational use will be found early in the testing cycle. Development of the model is based on the software specification permitting model development to begin before code development. This permits independence between the software development and usage modeling effort. Because the information learned from the analysis of the model is available early in the development cycle, it may be utilized to assist in resource planning, importance ranking, reducing interface complexity, and test planning. The steps for developing a usage model as described in [3] and [4] are:

- Review and clarify the software specifications
- Identify expected users and uses of the software
- Define a stratification of user and environment parameters
- Determine the desired levels of usage model granularity
- Develop and verify usage model structure

- Develop and verify probability distributions

A Markov chain usage model was developed for the MSDS. Using the Markov model, one may determine:

- the expected test case length (may be used to help estimate resources required for testing)
- the expected number of test cases required to cover each state (may be used to estimate cost to cover each state)
- the expected proportion of time spent in each state (which may be used to identify which sections of the model will get the most use and may deserve the most effort)
- the expected number of test cases to first occurrence of each state (may be used to identify scarcely used functions)
- source entropy (used to compare model complexity and give an estimate of the number of test cases that are worth executing)

A Markov chain usage model has a unique start state which we will refer to as invocation, a unique final state we will refer to as termination, a set of intermediate usage states, and transition arcs between them. The Markov property (probability law) requires that the next state is independent of all past states given the present state. Every state must have a next state and all transition probabilities must sum to 1. The Markov usage model of n states may be represented as a nxn stochastic matrix. Table 3.3 depicts a possible

Markov usage model of a trivial system as defined by the following specifications:

Upon invocation, a box is displayed containing the current time, the user may press button Refresh which will refresh the display, or button Exit which will terminate the application. Initially we have no usage information available for this system, therefore, on our first pass we simply mark the possible transitions.

Table 3.3: Markov Transition Matrix

	Inv.	DisplayTime	ButtonRefresh	ButtonExit	Term.
Inv.		X			
DisplayTime			X	X	
ButtonRefresh		X			
ButtonExit					X
Term.	X				

In order to make the chain recurrent, a transition from Termination to Invocation is required. After all the states and transitions have been identified, the transition probabilities are assigned. Depending on information available regarding use of the software, probabilities may be assigned:

- based on field data
- based on informed assumptions of expected use
- uniformly

In the example, suppose there is field data indicating that a user normally verifies the time several times before exit, then the assignment of .2 to exit and .8 to refresh might accurately reflect use. Table 3.4 represents the transition matrix with probabilities assigned.

Table 3.4: Example Matrix with Probabilities Assigned

	Inv.	DisplayTime	ButtonRefresh	ButtonExit	Term.
Inv.		1			
DisplayTime			.8	.2	
ButtonRefresh		1			
ButtonExit					1
Term.	1				

Now suppose that we have actual field data which shows that the user refreshes the time display 80% of the time after the initial invocation but exits 80% of the time after that. To accurately model this environment, it is necessary to know how many times the display has been refreshed. Since the Markov property requires that the next state is independent of all past states given the present state, we must split the DisplayTime state into two separate states, DisplayF to represent the first time the display is activated and DisplayN to represent all future displays. Using this additional state our new matrix is displayed in Table 3.5.

Table 3.5: Example Matrix with New State

	Inv.	DisplayF	DisplayN	ButtonRefresh	ButtonExit	Term.
Inv.		1				
DisplayF				.8	.2	
DisplayN				.2	.8	
ButtonRefresh			1			
ButtonExit						1
Term.	1					

One can see from this example how the handling of state data in software specifications can have an impact on the size and complexity of a model. This is one of the key issues in determining the level at which to model. There are tradeoffs between the level of detail to model and controlling model size and complexity. Managing the usage state space is one of the most critical elements in model development, especially when developing MIS type, data-form-screen, GUI applications. The trends in these types of applications has been towards modeless display and manipulation of multiple screens of information simultaneously. Model size and complexity can grow rapidly when specifications require interaction and dependency among multiple data-forms. This is a true reflection of the size of the testing problem such systems create. This was a major issue in actual development of a model for the MSDS application.

4. MSDS Model Design

4.1 Specifications Overview

To assist in clarification and understanding of the usage model developed for the MSDS, a brief overview and description of the specifications for the system are given.

Upon invocation of the software, the user is required to log into the system using a userid, password, and data source definition. After the user has been authenticated, the user is presented with an MDI frame window that contains a button bar menu and windows pull-down menu. Upon initial invocation, the query form is automatically displayed permitting the user to enter search criteria or select another option from the menu or button bar. From the menu or button bar, options are available to create new material records, update existing records, copy records, delete records, maintain supporting data, print material information, and exit the system. There are 12 primary data-forms utilized by the system to maintain the data. Most of these forms display data based upon the currently selected record on the query form. As typical of MDI applications, none of the forms are modal. The user may switch between forms at any point and have any combination of forms active on the desktop at any given time. They may update data at any given time during the process on a form by form basis. The only restriction is that only data for a single material record may be displayed at a

time. If the user switches records in the query screen, all the currently active screens will change to display the data for the new record. The exceptions to this process are the New and Copy functions. Since these processes create new material records, they will not be present in the query screen. The forms for this process are modal and block access to the remainder of the system during processing.

4.2 Model Development

In development of the Markov usage model for the MSDS application, the approach defined in [3] was used. An iterative approach as outlined was taken and produced startling results after just a few initial iterations through the specifications.

4.2.1 Review and clarify the software specifications

It was determined that for the initial iteration of the modeling process, a model would be developed for logging into the system, performing a query, and maintaining the "Standard Table" information for record returned from the query. Figure 4.1 depicts the MSDS login screen as described by the specifications.

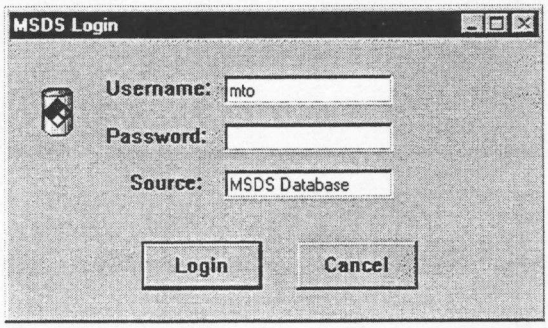

Figure 4.1: MSDS Login Screen

Upon successful login, the main MDI frame window will be displayed and the query form activated. Figure 4.2 depicts the MDI frame with the menu, button bar and query screen as depicted by our first iteration through the specifications. The user may enter query criteria and perform a search or exit the application. After a query has been executed, there are three forms that may be brought up by selecting their names on the pull-down menu. These forms are the three pages of the MSDS Standard Sheet and are also shown in Figure 4.2.

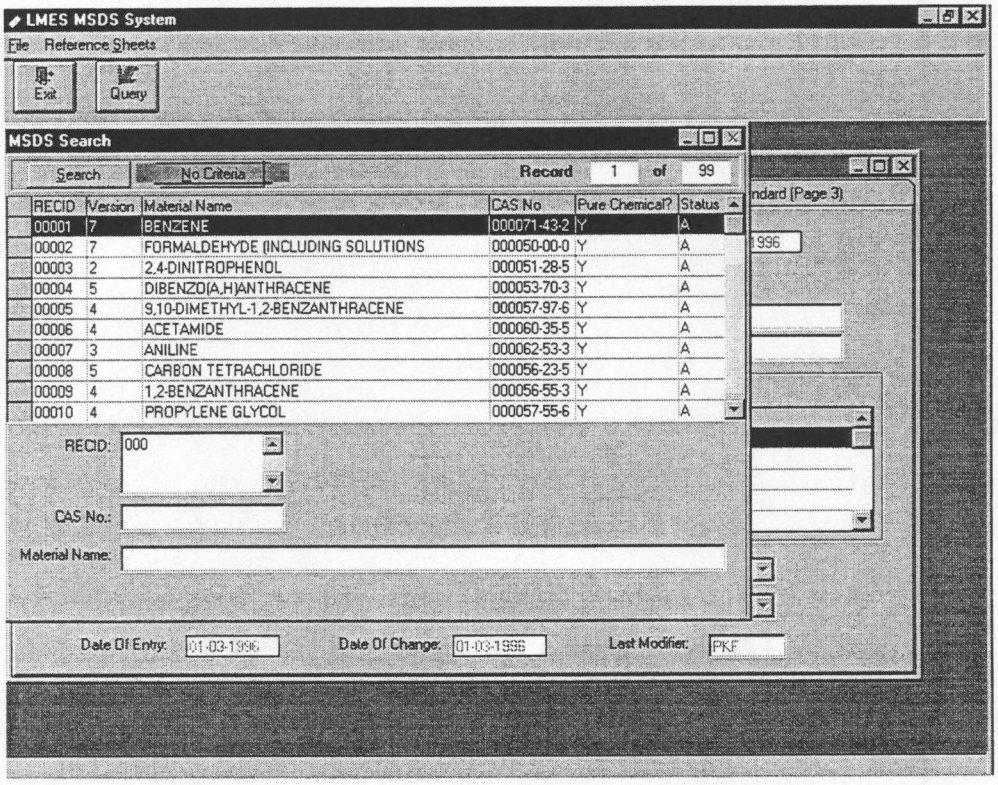

Figure 4.2: MSDS Frame Window With Query Screen

4.2.2 Identify expected users and uses of the software

Since only the data maintenance portion of the system is being modeled (field users access the information via a separate interface) the MSDS has a relatively small user base, consisting of 5 users, all performing the same function. Each user has the same system abilities and performs similar duties. This greatly simplifies model development because we do not have to worry about modeling different types of system use. All users have access to all the functions and perform like duties.

4.2.3 Define a stratification of user and environment parameters

The system was to utilize a dedicated server machine therefore minimizing any concerns of anomalies with the environment. Desktop configurations were identical for all users and were to be used exclusively for the system. Therefore, only a single stratification exists to be modeled.

4.2.4 Determine the desired levels of usage model granularity

The goal was to develop as detailed a model as possible. Achieving the appropriate level of abstraction turned out to be the most important aspect of model development. It was determined that to develop a model at the keystroke level would be unmanageable due the complexity and nature of the Windows operating system and the controls used. It was decided that use of the software would be modeled at the control level. Trust in the controls provided by the underlying operating system is required if modeling is to be performed at this level. Since we are using only the standard controls provided with the operating system, it was decided to trust the reliability of these controls. By the same reasoning, it was decided that for controls with multiple means of activation, such as a drop-down list box

that permits selection of an entry with the mouse click, use of arrow keys, or direct text typing, we would only refer to the selection or activation of an entry in that list, not attempt to model all the different mechanisms the operating system provides for activation. This applies also to the selection of an item in a grid control and the activation of a form itself. We did not attempt to model all the complexities built into the controls; we modeled only the functionality we were creating in our application.

As an aid to simplification, a state called Activate was added for each child form and the main MDI frame. This permitted a single entry point into the form. This was necessary due to the modelessness of the application. Since in the MDI model, the user is permitted to jump from any field on any form to any field on any other form, our model would have yielded transitions between nearly all available states. This was deemed too large to maintain early on. To further reduce the number of necessary arcs, all transitions to child forms were via the Activate state for the main MDI parent frame. This method reduced the number of transitions in the model significantly without any loss in granularity. This was due to the independence of the various MDI child forms. The specifications did not dictate any specific sequence through these forms nor did the usage data on which the probabilities were based.

As an example, examine the login form depicted in Figure 4.1. There are six controls involved in the use of this form: frmLogin.EnterUsername, frmLogin.EnterPassword, frmLogin.EnterDatasource, frmLogin.ClickOK (the default action is the OK button, in this case Login), frmLogin.ClickCancel, and the actual form itself frmLogin.Activate.

Modeling the software at the control level would yield the state list in Table 4.1

Since there are additional requirements dictated by the specifications for this form, namely, that the userid/password entered by the user be valid, we must introduce some additional states to handle this. The specifications call for validation of the userid, password, and datasource in response to the user pressing the OK button. We have one state, frmLogin.ClickOK to signify this event. We are not concerned with how this button was "clicked". The operating system provides the mechanism for this function via the mouse and keyboard and since we are trusting the OS and the control, we focus our attention on the fact that the button was in fact "clicked".

Table 4.1: State List For Login Screen

Invocation
frmLogin.Activate
frmLogin.EnterUsername
frmLogin.EnterPassword
frmLogin.EnterDatasource
frmLogin.ClickOK
frmLogin.ClickCancel
Termination

To handle the data validation states, one to three additional states may be added, once again depending on the level of granularity of the model. If in the software specification all login failures are treated the same, we could simply add a state frmLogin.InvalidLoginAttempt to handle all these attempts. If the requirements are to handle the errors in a different fashion, states could be defined to handle all the errors possible. Tables 4.2 and 4.3 depict the new list of states and new transition matrix with the introduction

of a single error handling capability. This demonstrates how granularity will effect the model. This modal screen, with 3 input fields and 2 buttons, contains 9 states and 21 arcs when using a high level of abstraction.

Table 4.2: Revised State List For Login Screen

Invocation
frmLogin.Activate
frmLogin.EnterUsername
frmLogin.EnterPassword
frmLogin.EnterDatasource
frmLogin.ClickOK
frmLogin.ClickCancel
frmLogin.InvalidLoginAttempt
Termination

Table 4.3: Transition Matrix For Login Screen

FROM STATE	TO STATE	PROBABILITY
Invocation	frmLogin.Activate	1.0
frmLogin.Activate	frmLogin.EnterPassword	1.0
frmLogin.EnterUsername	frmLogin.EnterDatasource	0.1
frmLogin.EnterUsername	frmLogin.EnterPassword	0.7
frmLogin.EnterUsername	frmLogin.ClickOK	0.1
frmLogin.EnterUsername	frmLogin.ClickCancel	0.1
frmLogin.EnterPassword	frmLogin.ClickOK	0.7
frmLogin.EnterPassword	frmLogin.ClickCancel	0.1
frmLogin.EnterPassword	frmLogin.EnterUsername	0.1
frmLogin.EnterPassword	frmLogin.EnterDatasource	0.1
frmLogin.EnterDatasource	frmLogin.ClickOK	0.7
frmLogin.EnterDatasource	frmLogin.ClickCancel	0.1
frmLogin.EnterDatasource	frmLogin.EnterUsername	0.1
frmLogin.EnterDatasource	frmLogin.EnterDatasource	0.1
frmLogin.InvalidLoginAttem	frmLogin.EnterUsername	0.1
frmLogin.InvalidLoginAttem	frmLogin.Password	0.7
frmLogin.InvalidLoginAttem	frmLogin.ClickCancel	0.1
frmLogin.InvalidLoginAttem	frmLogin.DataSource	0.1
frmLogin.ClickOK	frmLogin.InvalidLoginAtte	0.2
frmLogin.ClickOK	Terminate	0.8
frmLogin.ClickCancel	Terminate	1.0
Termination	Invoke	1.0

Some additional model features were also added at this point. Using the certification software, scripts (comments) were added to each of the transitions (arcs) describing the purpose of the transition so that the tester would know exactly what to do. This was documented in our test procedure to assist the tester in data input selections, e.g. specific types of invalid input or combinations of data.

The MSDS application utilizes list boxes for all input fields that are restricted, therefore it is not possible for the user to enter invalid data. In the limited cases where bad input is possible, one example being invalid login attempts as described earlier, it was left up to the tester to supply appropriate data to meet the needs of the test script. The procedure as documented was for the tester to examine the current test case and determine where valid and invalid data was to be entered. A list of all available data in the testing environment was also made available to the tester, to be utilized for this purpose. The data utilized for the test was then documented along with each action performed. Selection of the data was based upon the input specifications in the specifications document.

Testing was performed on a system specifically designated as a QA environment and all testing began with the system in a known and documented state. For invalid entries, all possible categories for input being rejected were documented. If a test case called for invalid input, the tester was to select the first failure condition and use that to build the input data set. That condition was then moved to the end of the list. The largest failure condition list contained 3 items. All of these rejection conditions were met in the course of random testing.

4.2.5 Develop/Verify usage model structure

After the initial granularity was defined and our standards for states documented, the next step was to define the initial list of states, based on forms access. A top down approach in defining the states was taken and the initial list was the list of all the screens as defined by the specifications and shown in Table 4.4.

Table 4.4: Initial States List

Invocation
frmLogin.Activate
frmMain.Activate
frmStandard1.Activate
frmStandard2.Activate
frmStandard3.Activate
frmListing.Activate
frmTargetOrgans.Activate
frmIngredients.Activate
frmOptions.Activate
frmAvid.Activate
frmStores.Activate
frmQuery.Activate
frmHistory.Activate
frmNew.Activate
frmDelete.Activate
frmCopy.Activate
Termination

From this initial list, each Activate state was expanded until each form was individually modeled. Using the approach detailed earlier where each form was given a unique entry and exit point, Activate and frmMain.Activate respectively, it was a simple task to model each form individually and then add to the whole. Adding the frmLogin information from Table 4.2, we are given the state list in Table 4.5.

Table 4.5: Second Iteration of States List

Invocation
frmLogin.Activate
frmLogin.EnterUsername
frmLogin.EnterpPassword
frmLogin.EnterDatasource
frmLogin.ClickOK
frmLogin.ClickCancel
frmLogin.InvalidLoginAttempt
frmMain.Activate
frmStandard1.Activate
frmStandard2.Activate
frmStandard3.Activate
frmListing.Activate
frmTargetOrgans.Activate
frmIngredients.Activate
frmOptions.Activate
frmAvid.Activate
frmStores.Activate
frmQuery.Activate
frmHistory.Activate
frmNew.Activate
frmDelete.Activate
frmCopy.Activate
Termination

After expanding the state for the frmLogin thru frmStandard3, it was soon discovered that we had a problem with model complexity. After implementing only 30% of the applications functionality, our model contained 265 states and 1132 arcs. We had a limited testing budget and it was obvious at this point that our system was not testable. This was primarily due to the modelessness of the system. At this point, a review of the specifications was called for. Instead of allowing for completely modeless operation among the forms, it was decided that forcing modifications to the data to be performed in a modal fashion would reduce the size of the testing problem. An impact analysis of this change was done and it was learned that this change would not only simplify the testing

problem, but the developers claimed that it would also reduce the complexity of the code. This correlation between complexity of usage model and complexity of code is very interesting. It stands to reason that there would be a direct relationship in MIS applications. The logic of the average MIS application is generally straightforward to code once the business rules have been defined. They generally contain standard Create, Read, Update, and Delete (CRUD) capabilities. The complexity of these MIS applications often lies in the interfaces. In a GUI (event driven) environment, most of the business logic is handled via event sequences and often there exists a tight coupling of information among various screens. We found that by adding modal constraints to the specifications, not only was the model complexity reduced, we were able to remove almost 15% of the code from the system. This 15% reduction reflects removal of code already developed (since model development was occurring simultaneously with software development) then removed by the developers when the specification change was approved. An additional modification made to the specifications was to reduce the number of forms in the applications. The Standard Table information was combined into one form. The resultant state list is described in Table 4.6. Some additional characteristics that were modeled during this iteration were the automatic invocation of the query screen after a successful login and the restriction of certain menu options to require data from the query process. The important change to note during this iteration was the addition of the states to differentiate between updating the data and viewing the data. No transitions from ActivateUpdateMode states are permitted to another Activate state until the update is saved or aborted.

Table 4.6: MSDS State List After Specification Modifications

Invocation
frmLogin.Activate
frmLogin.EnterUsername
frmLogin.EnterpPassword
frmLogin.EnterDatasource
frmLogin.ClickOK
frmLogin.ClickCancel
frmLogin.InvalidLoginAttempt
frmMain.ActivateFirstTime
frmMain.ActivateNoQueryRecords
frmMain.ActivateQueryRecordAvail
frmStandard.ActivateViewMode
frmStandard.ActivateUpdateMode
frmListing.ActivateViewMode
frmListing.ActivateUpdateMode
frmTargetOrgans.ActivateViewMode
frmTargetOrgans.ActivateUpdateMode
frmIngredients.ActivateViewMode
frmIngredients.ActivateUpdateMode
frmOptions.ActivateViewMode
frmOptions.ActivateUpdateMode
frmAvid.ActivateViewMode
frmAvid.ActivateUpdateMode
frmStores.ActivateViewMode
frmStores.ActivateUpdateMode
frmQuery.Activate
frmHistory.Activate
frmNew.Activate
frmDelete.Activate
frmCopy.Activate
Termination

Using this iterative technique, the final state list was completed very quickly from this point. No additional specification changes were required. The resulting model had 130 states and 565 arcs.

4.2.6 Develop/Verify probability distributions

The next step in model development was to make the probability assignments. The methods described earlier are assigned:

- based on field data
- based on informed assumptions of expected use
- uniformly

All three approaches were used. Data was available regarding frequency of update to material information. Interviews were performed with the end users to see if a typical pattern existed during daily activities. It was determined that at least ½ of a typical day was spent modifying existing data while only a small portion was spent adding new material. Using this information, the probabilities were defined as in Table 4.7. All remaining probabilities were assigned a uniform distribution, because there was no information to justify any other values.

At this point, we began analysis of the model. The mean number of occurrences was examined along with mean first passage and long run

Table 4.7: Probabilities Based On End-User Interviews

FROM	TO	PROBABILITY
frmMain.ActivateQueryRecordsAvailable	frmQuery.Activate	.6
frmMain.ActivateQueryRecordsAvailable	frmMain.New	.3
frmMain.ActivateQueryRecordsAvailable	frmMain.Exit	.1

probabilities. These were utilized to detect "hot spots" in the system and assist in verification. The user interviews revealed the "most common" functions as defined by the users. The hot spots identified by model analysis did agree with the most common functions identified by the users.

Attempts were made to assign these "hot spots" to the most experienced software developer. Some example test cases were generated and examined to verify that they depicted proper use of the system. Based on these results, some slight modifications to the probabilities were made. The majority of these modifications involved the elevation of the probabilities of saving modified information. To prevent test cases from spending considerable time repeatedly modifying data within the same form, a test management issue, the probabilities for transitions to the Save states were increased.

4.3 Model Validation

Test cases were generated and analyzed. Routine Markov model calculations provided additional information.

- the expected test case length

The expected script length for the final model was 69 steps. Using data obtained from our earlier experiments, we estimated the average expected duration of a test script to be 10 minutes.

- the expected number of test cases required to cover each state

The least likely state coverage was expected to occur with 124 test cases.

- the expected proportion of time spent in each state

This was used to identify the key sections of the system. In our model, this turned out to be the Login, Query, and Standard table portions. As a result of this, these portions were assigned to the most experienced developer.

- the expected number of test cases to first occurrence of each state

This information revealed low user functions in the system. This information was compared with our information regarding system use and was used as a means of model validation.

4.4 Test Planning

4.4.1 Stopping Criteria

Based upon the analysis from the model, our stopping criteria was defined as the following:

- System attains .99 reliability with a confidence of 90%, which requires 230 random test cases without a failure

- Complete state coverage is achieved by random testing

4.4.2 Resource Planning

It was determined that approximately 50 test cases could be executed per day. Using this number and the estimated number of tests to achieve coverage of the least likely state, we estimated completion of the test to be not less than 4 days but not more that 4 weeks (assuming 4 failures). This upper bound was based on 1 day to find a problem and seeing no more than 4 problems. This calculation was performed during every iteration of our model as the determining factor in the affordable level of granularity possible within our budget. We started with the universal model: Software Invocation -> Process -> Software Termination and built up to our current model of 130 states. Since this model yielded analysis results within our parameters, we decided that further iterations were unnecessary.

5. Test Results

5.1 Testing Results

The system was submitted for testing and execution of the test scripts commenced. To the disappointment of the team, we had our first failure on the very first random test case. Table B.1 in appendix B details the system failures. Eight failures were encountered during testing of the system.

At this point, the testing process was halted. Eight errors were considered excessive and it was decided the software was not ready for testing.

5.2 Minimum-Cost Model-Coverage Test Set

The minimum-cost model-coverage test set[1] (MCMCTS) was generated from the graph of the Markov Model. This test scenario guaranteed arc coverage. It was decided that once the software was able to complete this test, random testing would once again commence. The MCMCTS consisted of 4 test cases with approximately 1200 transitions (combined). This test was completed in approximately one week's time.

Upon successful completion of this process, certification testing was resumed. The system reached our goal of complete state coverage after completing 337 additional test cases. The previous error (number 8) having occurring at test case 124. Using the zero failures table described in [5], to

[1] This test set might have been run first, but the author did not learn of it until this point in the project

achieve $r = .99$ with 90% confidence, it is necessary to execute 230 test cases without failure. Our state coverage was achieved after executing 337 cases without failure yielding $r \approx .997$ with 90% confidence.

After completing the certification process a new series of test cases was generated and executed. The MCMCTS would have found all eight errors. The system would have certified after executing 230 random test cases, about half the number used in testing. Using this approach, we would have cut our testing time in half.

5.3 Failure Analysis

An analysis of the failures discovered in the software was compared with the user acceptance test plan generated by the customer. Each of the errors located during certification mapped to an appropriate test scenario in the user's acceptance test. This analysis revealed that two of the design errors, initially present in the software but found and removed by random testing, would have been accepted by the customer.

5.4 States and Arcs not Covered

The certification test tool also reveals a list of states and arcs that are not covered by the test scenarios generated thus far. This information is useful and necessary for determining whether your test plan contains adequate model coverage. In our test scheme, it was determined that when the target reliability was achieved and complete state coverage obtained, we would halt testing. State coverage was obtained, during random testing, but not arc coverage. Since a MCMCTS was performed, this was not a concern, but the list of arcs not covered was still analyzed. It was determined that all the arcs

not covered were for very low and unlikely transitions between non-related fields within a form.

5.5 Reliability Graph

Figure C.1 in appendix C displays a graph of reliability, **R**, and **d(U,T)**, a stochastic measure of the difference between testing experience and field use as modeled, [6] (on the y-axis), versus the number of test cases executed (on the x-axis). This information is useful in determining whether benefit would be realized by executing additional test cases. One can see from the chart, that each curve has become asymptotic. This informs the tester that little benefit would be achieved from additional testing.

6. Conclusion

Usage testing is emerging as an effective and economical testing methodology. This study indicates that usage testing is a viable solution for GUI based MIS applications. It provides a wealth of information about the complexity of the interface and provides a better approach to allocating the testing budget over code-coverage testing. As demonstrated in this case study, three of the items of information readily available are:

- complexity of interface
- relative use of specific modules
- test resource allocation

Model development, when performed as a disciplined engineering process, may be accomplished with the same resources as necessary for traditional test plan development. Since all the information necessary for the usage model is derived from the specifications, all these benefits may be realized early in the development process, potentially avoiding rework.

The techniques utilized in this paper will function at many levels. They scale to the most complex application. One simply models to the level of granularity that the process and budget warrants. This ability to optimize the testing process is as important as the test plan and test results themselves

for large scale MIS applications. Clearly, mission-critical and safety-critical systems should be modeled to the lowest level possible, but for typical Management Information Systems (MIS), risk-reward tradeoffs may be made to achieve the highest possible reliability within your budget.

REFERENCES

References

[1] E.N. Adams, "Optimizing Preventative Service of Software Products", *IBM Journal of Research and Development*, Vol 28, No 1, pp 2-14, 1984

[2] H.D. Mills, M. Dyer, R.C. Linger, "Cleanroom Software Engineering", *IEEE Software*, pp 19-24, September 1987

[3] J.H. Poore, G.H. Walton, C.J. Trammell, "Statistical Testing of Software Based on a Usage Model", *Software-Practice and Experience*, Vol 25, No 1, pp 97-108, January 1995

[4] J.H. Poore, C.J. Trammell, "Cleanroom Software Engineering: A Reader", Blackwell Publishers Inc., Cambridge, MA, pp 65-67, 83-110, 287-344, 1996

[5] J.H. Poore, H.D. Mills, D. Mutchler, "Planning and Certifying Software System Reliability", *IEEE Software.*, pp 88-99, January 1993

[6] J.A. Whittaker, M.G. Thomason, "A Markov Chain Model for Statistical Software Testing", *IEEE Software*, pp 812-824, October 1994

APPENDICES

Appendix A

Appendix A details the final version of the MSDS Usage Model. The format for the model is as follows. The *from state* is defined and all transitions from that state are listed. For each *to state* listed, the transition probability along with the way that probability was determined is given.

In the source column, exit arcs are p1 ... pn and Δ means the remaining probabilities are distributed uniformly.

From State: Software Not Invoked

To State	Source	Prob	Description
frmLogin.Activate	NA	1.0	Invoke application and display login screen

From State: frmLogin.Activate

To State	Source	Prob	Description
frmLogin.EnterUsername	p1=1/2(p2)	.3	Enter a username in the username text box
frmLogin.EnterPassword	p2=.6	.6	Enter a password in the password text box
frmLogin.EnterDataSource	Δ	.03	Enter a datasource in the datasource text box
frmLogin.Close	Δ	.03	Press CLOSE button
frmLogin.Login	Δ	.03	Press Login button

From State: frmLogin.EnterUsername

To State	Source	Prob	Description
frmLogin.EnterUsername	Δ	.1	Enter a username in the username text box
frmLogin.EnterPassword	p2=.6	.6	Enter a password in the password text box
frmLogin.EnterDataSource	Δ	.1	Enter a datasource in the datasource

			text box
frmLogin.Close	Δ	.1	Press CLOSE button
frmLogin.Login	Δ	.1	Press Login button

From State: frmLogin.EnterPassword

To State	Source	Prob	Description
frmLogin.EnterUsername	Δ	.025	Enter a username in the username text box
frmLogin.EnterPassword	Δ	.025	Enter a password in the password text box
frmLogin.EnterDataSource	Δ	.025	Enter a datasource in the datasource text box
frmLogin.Close	Δ	.025	Press CLOSE button
frmLogin.Login	p5=.8	.8	Press Login button

From State: frmLogin.EnterDatasource

To State	Source	Prob	Description
frmLogin.EnterUsername	Δ	.05	Enter a username in the username text box
frmLogin.EnterPassword	Δ	.05	Enter a password in the password text box
frmLogin.EnterDataSource	Δ	.05	Enter a datasource in the datasource text box
frmLogin.Close	Δ	.05	Press CLOSE button
frmLogin.Login	p5=.8	.8	Press Login button

From State: frmLogin.Close

To State	Source	Prob	Description
Software Terminated	NA	1	Close application

From State: frmLogin.Login

To State	Source	Prob	Description
frmMain.ActivateFirst Time	p1=.95	.95	Successful login. Bring up main MDI form
frmLogin.BadLogin	Δ	.05	Invalid login attempt. Retry.

From State: frmLogin.BadLogin

To State	Source	Prob	Description
frmLogin.Activate	NA	1	Re-enter login information and try again.

From State: frmMain.ActivateFirstTime

To State	Source	Prob	Description
frmQuery.Activate	NA	1	First time application is executed, the query screen

			should be brought up upon successful login.

From State: frmQuery.Activate

To State	Source	Prob	Description
frmQuery.PerformRECID Search	p1=p2=p3	.3	Enter a recid and press search button
frmQuery.PerformCAS Search	p1=p2=p3	.3	Enter a CAS number and press search button
frmQuery.PerformMaterialN ameSearch	p1=p2=p3	.3	Enter a material name or synonym and press search button
frmQuery.Cancel	Δ	.1	Do not perform a query

From State: frmQuery.PerformRECIDSearch

To State	Source	Prob	Description
frmQuery.ValidSearch	p1=.9	.9	Search criteria entered returns a results list
frmQuery.InvalidSearch	Δ	.1	Search criteria entered returns no records

From State: frmQuery.PerformCASSsarch

To State	Source	Prob	Description
frmQuery.ValidSearch	p1=.9	.9	Search criteria entered returns a results list
frmQuery.InvalidSearch	Δ	.1	Search criteria entered returns no records

From State: frmQuery.PerformMaterialNameSearch

To State	Source	Prob	Description
frmQuery.ValidSearch	p1=.7	.7	Search criteria entered returns a results list
frmQuery.InvalidSearch	Δ	.3	Search criteria entered returns no records

From State: frmQuery.ValidSearch

To State	Source	Prob	Description
frmQuery.SelectItem InGrid	p1=.75	.75	Examine list of query results and select the appropriate material from the grid.
frmQuery.Activate	p2=.2	.2	Perform a different query
frmQuery.Cancel	Δ	.05	Close query window

			and remove list

From State: frmQuery.InvalidSearch

To State	Source	Prob	Description
frmQuery.Activate	p1=.95	.95	Perform a different query (Try again)
frmQuery.Cancel	Δ	.05	Close query window

From State: frmQuery.SelectItemInGrid

To State	Source	Prob	Description
frmMain.ActivateQuery RecordsAvailable	NA	1	A Record from the grid has been selected. All menu options assocated with material should now be activated.

From State: frmQuery.Cancel

To State	Source	Prob	Description
frmMain.ActivateNoQuery RecordsAvailable	NA	1	Close query window removing list. The only options available here are Exit, New, and Query

From State: frmMain.ActivateNoQueryRecords

To State	Source	Prob	Description
frmQuery.Activate	p1=.6	.6	Perform a different query
frmMain.New	p2=1/2(p1)	.3	Create a new Material Record
frmMain.Exit	Δ	.1	Exit the application

From State: frmMain.ActivateQueryRecordsAvailable

To State	Source	Prob	Description
frmQuery.SelectItemIn Grid	Δ	.071	Select a different (or the same again) material from the query reults list
frmMain.New	Δ	.071	Create a new Material Record
frmMain.Copy	Δ	.071	Create a new Material Record by copying the currently selected to a new RECID
frmMain.Delete	Δ	.071	Archive the currently selected material
frmMain.PrintMSRS	Δ	.071	Print a material safety reference

			sheet for the currently selected material
frmStandard.ActivateViewMode	Δ	.071	Bring up the standard table form in read-only mode
frmIng.ActivateViewMode	Δ	.071	Bring up the ingredients table form in read-only mode
frmListing.ActivateViewMode	Δ	.071	Bring up the listing table form in read-only mode
frmTagetOrgans.ActivateViewMode	Δ	.071	Bring up the target organ table form in read-only mode
frmAvid.ActivateViewMode	Δ	.071	Bring up the avid reference table form in read-only mode
frmStores.ActivateViewMode	Δ	.071	Bring up the stores reference table form in read-only mode
frmHistory.ActivateViewMode	Δ	.071	Bring up the history table form in read-only mode
frmOptions.Activate	Δ	.071	Bring up the user options form
frmMain.Exit	Δ	.071	Exit the application

From State: frmMain.New

To State	Source	Prob	Description
frmNew.MaterialName	p1=.9	.9000	Enter the name of the new material
frmNew.Cancel	Δ	.1000	Cancel the new record creation

From State: frmMain.Delete

To State	Source	Prob	Description
frmDelete.Verify.Yes	p1=.9	.9000	Yes, archive the record
frmDelete.Verify.No	v	.1000	No, cancel archival request

From State: frmMain.Copy

To State	Source	Prob	Description
frmCopy.NewMaterialName	p1=.5	.5000	Modify the material name of the copied record
frmCopy.NewNameSort	Δ	.0677	Modify the sort name of the copied record
frmCopy.NewType	Δ	.0677	Modify the

			material type of the copied record
frmCopy.Save	p4=.3	.3000	Save the new material and close modal copy window
frmCopy.Cancel	Δ	.0677	Abort copy operation

From State: frmMain.Exit

To State	Source	Prob	Description
Software Terminated	NA	1.0000	Exit the application

From State: frmMain.PrintMSRS

To State	Source	Prob	Description
frmMain.ActivateQuery RecordsAvailable	NA	1.0000	Activate Main Menu with options records available options

From State: frmMain.Exit

To State	Source	Prob	Description
Software Terminated	NA	1.0000	Exit system

From State: frmStandard.ActivateViewMode

To State	Source	Prob	Description
frmMain.ActivateQuery RecordsAvailable	Δ	0.1666	Activate Main Window
frmStandard.SynSelect	Δ	0.1666	Select a synonym from list
frmStandard.UpdateSyn Button	Δ	0.1666	Update current synonym in list
frmStandard.AddSynButton	Δ	0.1666	Add A Synonym to list
frmStandard.Cancel	Δ	0.1666	Close standard Form
frmStandard.Modify	Δ	0.1666	: Press modify button to lock record and enable modification

From State: frmStandard.ActivateUpdateMode

To State	Source	Prob	Description
frmStandard.CASNo	Δ	0.0554	Edit Material CAS Number
frmStandard.PureChemical	p2=.01	0.0100	Edit Pure Chemical Flag
frmStandard.MaterialName	Δ	0.0554	Edit Material Name
frmStandard.NameSort	Δ	0.0554	Edit Sorting Name
frmStandard.SynSelect	Δ	0.0554	Select synonym from list
frmStandard.EPAID	Δ	0.0554	Edit EPA Id
frmStandard.RTECSNo	Δ	0.0554	Edit RTECS Number
frmStandard.MSDSType	p8=.01	0.0100	Edit MSDS Type
frmStandard.MSDSStatus	p9=.01	0.0100	Edit MSDS Status
frmStandard.Req	p10=.01	0.0100	Edit Requests

Outstanding			Outstanding
frmStandard.MSDS Available	Δ	0.0554	Edit MSDS Available Flag
frmStandard.Cancel	p12=.05	0.0500	Cancel Modify Mode
frmStandard.Save	p13≥.3	0.3544	Save Modifications
frmStandard.HealthHazardRating	Δ	0.0554	Edit Health Hazard Rating
frmStandard.FireHazard Rating	Δ	0.0554	Edit Fire Hazard Rating
frmStandard.Reactivity HazardRating	Δ	0.0554	Edit Reactivity Rating

From State: frmStandard.CASNo

To State	Source	Prob	Description
frmStandard.CASNo	Δ	0.0554	Edit Material CAS Number
frmStandard.PureChemical	p2=.01	0.0100	Edit Pure Chemical Flag
frmStandard.MaterialName	Δ	0.0554	Edit Material Name
frmStandard.NameSort	Δ	0.0554	Edit Sorting Name
frmStandard.SynSelect	Δ	0.0554	Select synonym from list
frmStandard.EPAID	Δ	0.0554	Edit EPA Id
frmStandard.RTECSNo	Δ	0.0554	Edit RTECS Number
frmStandard.MSDSType	p8=.01	0.0100	Edit MSDS Type
frmStandard.MSDSStatus	p9=.01	0.0100	Edit MSDS Status
frmStandard.Req Outstanding	p10=.01	0.0100	Edit Requests Outstanding
frmStandard.MSDS Available	Δ	0.0554	Edit MSDS Available Flag
frmStandard.Cancel	p12=.05	0.0500	Cancel Modify Mode
frmStandard.Save	p13≥.3	0.3554	Save Modifications
frmStandard.HealthHazardRating	Δ	0.0554	Edit Health Hazard Rating
frmStandard.FireHazard Rating	Δ	0.0554	Edit Fire Hazard Rating
frmStandard.Reactivity HazardRating	Δ	0.0554	Edit Reactivity Rating

From State: frmStandard.PureChemical

To State	Source	Prob	Description
frmStandard.CASNo	Δ	0.0554	Edit Material CAS Number
frmStandard.PureChemical	p2=.01	0.0100	Edit Pure Chemical Flag
frmStandard.MaterialName	Δ	0.0554	Edit Material Name
frmStandard.NameSort	Δ	0.0554	Edit Sorting Name
frmStandard.SynSelect	Δ	0.0554	Select synonym from list
frmStandard.EPAID	Δ	0.0554	Edit EPA Id
frmStandard.RTECSNo	Δ	0.0554	Edit RTECS Number
frmStandard.MSDSType	p8=.01	0.0100	Edit MSDS Type
frmStandard.MSDSStatus	p9=.01	0.0100	Edit MSDS Status
frmStandard.Req Outstanding	p10=.01	0.0100	Edit Requests Outstanding

frmStandard.MSDS Available	Δ	0.0554	Edit MSDS Available Flag
frmStandard.Cancel	p12=.05	0.0500	Cancel Modify Mode
frmStandard.Save	p13≥.3	0.3554	Save Modifications
frmStandard.HealthHazardRating	Δ	0.0554	Edit Health Hazard Rating
frmStandard.FireHazard Rating	Δ	0.0554	Edit Fire Hazard Rating
frmStandard.Reactivity HazardRating	Δ	0.0554	Edit Reactivity Rating

From State: frmStandard.MaterialName

To State	Source	Prob	Description
frmStandard.CASNo	Δ	0.0554	Edit Material CAS Number
frmStandard.PureChemical	p2=.01	0.0100	Edit Pure Chemical Flag
frmStandard.MaterialName	Δ	0.0554	Edit Material Name
frmStandard.NameSort	Δ	0.0554	Edit Sorting Name
frmStandard.SynSelect	Δ	0.0554	Select synonym from list
frmStandard.EPAID	Δ	0.0554	Edit EPA Id
frmStandard.RTECSNo	Δ	0.0554	Edit RTECS Number
frmStandard.MSDSType	p8=.01	0.0100	Edit MSDS Type
frmStandard.MSDSStatus	p9=.01	0.0100	Edit MSDS Status
frmStandard.Req Outstanding	p10=.01	0.0100	Edit Requests Outstanding
frmStandard.MSDS Available	Δ	0.0554	Edit MSDS Available Flag
frmStandard.Cancel	p12=.05	0.0500	Cancel Modify Mode
frmStandard.Save	p13≥.3	0.3554	Save Modifications
frmStandard.HealthHazardRating	Δ	0.0554	Edit Health Hazard Rating
frmStandard.FireHazard Rating	Δ	0.0554	Edit Fire Hazard Rating
frmStandard.Reactivity HazardRating	Δ	0.0554	Edit Reactivity Rating

From State: frmStandard.NameSort

To State	Source	Prob	Description
frmStandard.CASNo	Δ	0.0554	Edit Material CAS Number
frmStandard.PureChemical	p2=.01	0.0100	Edit Pure Chemical Flag
frmStandard.MaterialName	Δ	0.0554	Edit Material Name
frmStandard.NameSort	Δ	0.0554	Edit Sorting Name
frmStandard.SynSelect	Δ	0.0554	Select synonym from list
frmStandard.EPAID	Δ	0.0554	Edit EPA Id
frmStandard.RTECSNo	Δ	0.0554	Edit RTECS Number
frmStandard.MSDSType	p8=.01	0.0100	Edit MSDS Type
frmStandard.MSDSStatus	p9=.01	0.0100	Edit MSDS Status
frmStandard.Req Outstanding	p10=.01	0.0100	Edit Requests Outstanding
frmStandard.MSDS	Δ	0.0554	Edit MSDS

Available			Available Flag
frmStandard.Cancel	p12=.05	0.0500	Cancel Modify Mode
frmStandard.Save	p13≥.3	0.3554	Save Modifications
frmStandard.HealthHazardRating	Δ	0.0554	Edit Health Hazard Rating
frmStandard.FireHazard Rating	Δ	0.0554	Edit Fire Hazard Rating
frmStandard.Reactivity HazardRating	Δ	0.0554	Edit Reactivity Rating

From State: frmStandard.AddSynButton

To State	Source	Source	Description
frmStandard.SynText	NA	1.0000	Enter synonym text

From State: frmStandard.UpdateSynButton

To State	Source	Prob	Description
frmStandard.SynText	NA	1.0000	Modify Synonym Text

From State: frmStandard.OKSynButton

To State	Source	Prob	Description
frmStandard.ActivateViewMode	NA	1.0000	Perform other standard table operation

From State: frmStandard.DeleteSynButton

To State	Source	Prob	Description
frmStandard.DeleteSynButton.Yes	p1=4(p2)	0.8000	Confirm Delete [Yes]
frmStandard.DeleteSynButton.No	p2=1/4(p1)	0.2000	Cancel Delete [No]

From State: frmStandard.DeleteSynButton.Yes

To State	Source	Prob	Description
frmStandard.ActivateViewMode	NA	1.0000	Activate Standard Window

From State: frmStandard.DeleteSynButton.No

To State	Source	Prob	Description
frmStandard.ActivateViewMode	NA	1.0000	Activate Standard Window

From State: frmStandard.CancelSynButton

To State	Source	Prob	Description
frmStandard.ActivateViewMode	NA	1.0000	Perform other standard table operation

From State: frmStandard.SynText

To State	Source	Prob	Description
frmStandard.CancelSynButton	p1=1/4(p2)	0.2000	Abort Add
frmStandard.OKSynButton	p2=4(p1)	0.8000	Save new synonym

From State: frmStandard.SynSelect

To State	Source	Prob	Description

frmStandard.UpdateSyn Button	p1=.5	0.5000	Update currently selected synonym
frmStandard.DeleteSynButt on	c	0.2500	Delete synonym from list
frmStandard.ActivateViewM ode	v	0.2500	Perform other standard action

From State: frmStandard.EPAID

To State	Source	Prob	Description
frmStandard.CASNo	Δ	0.0554	Edit Material CAS Number
frmStandard.PureChemical	p2=.01	0.0100	Edit Pure Chemical Flag
frmStandard.MaterialName	Δ	0.0554	Edit Material Name
frmStandard.NameSort	Δ	0.0554	Edit Sorting Name
frmStandard.SynSelect	Δ	0.0554	Select synonym from list
frmStandard.EPAID	Δ	0.0554	Edit EPA Id
frmStandard.RTECSNo	Δ	0.0554	Edit RTECS Number
frmStandard.MSDSType	p8=.01	0.0100	Edit MSDS Type
frmStandard.MSDSStatus	p9=.01	0.0100	Edit MSDS Status
frmStandard.Req Outstanding	p10=.01	0.0100	Edit Requests Outstanding
frmStandard.MSDS Available	Δ	0.0554	Edit MSDS Available Flag
frmStandard.Cancel	p12=.05	0.0500	Cancel Modify Mode
frmStandard.Save	p13≥.3	0.3554	Save Modifications
frmStandard.HealthHazardR ating	Δ	0.0554	Edit Health Hazard Rating
frmStandard.FireHazard Rating	Δ	0.0554	Edit Fire Hazard Rating
frmStandard.Reactivity HazardRating	Δ	0.0554	Edit Reactivity Rating

From State: frmStandard.RTECSNo

To State	Source	Prob	Description
frmStandard.CASNo	Δ	0.0554	Edit Material CAS Number
frmStandard.PureChemical	p2=.01	0.0100	Edit Pure Chemical Flag
frmStandard.MaterialName	Δ	0.0554	Edit Material Name
frmStandard.NameSort	Δ	0.0554	Edit Sorting Name
frmStandard.SynSelect	Δ	0.0554	Select synonym from list
frmStandard.EPAID	Δ	0.0554	Edit EPA Id
frmStandard.RTECSNo	Δ	0.0554	Edit RTECS Number
frmStandard.MSDSType	p8=.01	0.0100	Edit MSDS Type
frmStandard.MSDSStatus	p9=.01	0.0100	Edit MSDS Status
frmStandard.Req Outstanding	p10=.01	0.0100	Edit Requests Outstanding
frmStandard.MSDS Available	Δ	0.0554	Edit MSDS Available Flag
frmStandard.Cancel	p12=.05	0.0500	Cancel Modify Mode
frmStandard.Save	p13≥.3	0.3554	Save Modifications
frmStandard.HealthHazardR	Δ	0.0554	Edit Health Hazard

ating			Rating
frmStandard.FireHazard Rating	Δ	0.0554	Edit Fire Hazard Rating
frmStandard.ReactivityHaz ardRating	Δ	0.0554	Edit Reactivity Rating

From State: frmStandard.MSDSType

To State	Source	Prob	Description
frmStandard.CASNo	Δ	0.0554	Edit Material CAS Number
frmStandard.PureChemical	p2=.01	0.0100	Edit Pure Chemical Flag
frmStandard.MaterialName	Δ	0.0554	Edit Material Name
frmStandard.NameSort	Δ	0.0554	Edit Sorting Name
frmStandard.SynSelect	Δ	0.0554	Select synonym from list
frmStandard.EPAID	Δ	0.0554	Edit EPA Id
frmStandard.RTECSNo	Δ	0.0554	Edit RTECS Number
frmStandard.MSDSType	p8=.01	0.0100	Edit MSDS Type
frmStandard.MSDSStatus	p9=.01	0.0100	Edit MSDS Status
frmStandard.Req Outstanding	p10=.01	0.0100	Edit Requests Outstanding
frmStandard.MSDS Available	Δ	0.0554	Edit MSDS Available Flag
frmStandard.Cancel	p12=.05	0.0500	Cancel Modify Mode
frmStandard.Save	p13≥.3	0.3554	Save Modifications
frmStandard.HealthHazardR ating	Δ	0.0554	Edit Health Hazard Rating
frmStandard.FireHazard Rating	Δ	0.0554	Edit Fire Hazard Rating
frmStandard.Reactivity HazardRating	Δ	0.0554	Edit Reactivity Rating

From State: frmStandard.MSDSStatus

To State	Source	Prob	Description
frmStandard.CASNo	Δ	0.0554	Edit Material CAS Number
frmStandard.PureChemical	p2=.01	0.0100	Edit Pure Chemical Flag
frmStandard.MaterialName	Δ	0.0554	Edit Material Name
frmStandard.NameSort	Δ	0.0554	Edit Sorting Name
frmStandard.SynSelect	Δ	0.0554	Select synonym from list
frmStandard.EPAID	Δ	0.0554	Edit EPA Id
frmStandard.RTECSNo	Δ	0.0554	Edit RTECS Number
frmStandard.MSDSType	p8=.01	0.0100	Edit MSDS Type
frmStandard.MSDSStatus	p9=.01	0.0100	Edit MSDS Status
frmStandard.Req Outstanding	p10=.01	0.0100	Edit Requests Outstanding
frmStandard.MSDS Available	Δ	0.0554	Edit MSDS Available Flag
frmStandard.Cancel	p12=.05	0.0500	Cancel Modify Mode
frmStandard.Save	p13≥.3	0.3554	Save Modifications
frmStandard.HealthHazardR ating	Δ	0.0554	Edit Health Hazard Rating

frmStandard.FireHazardRating	Δ	0.0554	Edit Fire Hazard Rating
frmStandard.ReactivityHazardRating	Δ	0.0554	Edit Reactivity Rating

From State: frmStandard.HealthHazardRating

To State	Source	Prob	Description
frmStandard.CASNo	Δ	0.0554	Edit Material CAS Number
frmStandard.PureChemical	p2=.01	0.0100	Edit Pure Chemical Flag
frmStandard.MaterialName	Δ	0.0554	Edit Material Name
frmStandard.NameSort	Δ	0.0554	Edit Sorting Name
frmStandard.SynSelect	Δ	0.0554	Select synonym from list
frmStandard.EPAID	Δ	0.0554	Edit EPA Id
frmStandard.RTECSNo	Δ	0.0554	Edit RTECS Number
frmStandard.MSDSType	p8=.01	0.0100	Edit MSDS Type
frmStandard.MSDSStatus	p9=.01	0.0100	Edit MSDS Status
frmStandard.Req Outstanding	p10=.01	0.0100	Edit Requests Outstanding
frmStandard.MSDS Available	Δ	0.0554	Edit MSDS Available Flag
frmStandard.Cancel	p12=.05	0.0500	Cancel Modify Mode
frmStandard.Save	p13≥.3	0.3554	Save Modifications
frmStandard.HealthHazardRating	Δ	0.0554	Edit Health Hazard Rating
frmStandard.FireHazard Rating	Δ	0.0554	Edit Fire Hazard Rating
frmStandard.Reactivity HazardRating	Δ	0.0554	Edit Reactivity Rating

From State: frmStandard.FireHazardRating

To State	Source	Prob	Description
frmStandard.CASNo	Δ	0.0554	Edit Material CAS Number
frmStandard.PureChemical	p2=.01	0.0100	Edit Pure Chemical Flag
frmStandard.MaterialName	Δ	0.0554	Edit Material Name
frmStandard.NameSort	Δ	0.0554	Edit Sorting Name
frmStandard.SynSelect	Δ	0.0554	Select synonym from list
frmStandard.EPAID	Δ	0.0554	Edit EPA Id
frmStandard.RTECSNo	Δ	0.0554	Edit RTECS Number
frmStandard.MSDSType	p8=.01	0.0100	Edit MSDS Type
frmStandard.MSDSStatus	p9=.01	0.0100	Edit MSDS Status
frmStandard.ReqOutstanding	p10=.01	0.0100	Edit Requests Outstanding
frmStandard.MSDS Available	Δ	0.0554	Edit MSDS Available Flag
frmStandard.Cancel	p12=.05	0.0500	Cancel Modify Mode
frmStandard.Save	p13≥.3	0.3554	Save Modifications
frmStandard.HealthHazardRating	Δ	0.0554	Edit Health Hazard Rating
frmStandard.FireHazard	Δ	0.0554	Edit Fire Hazard

Rating			Rating
frmStandard.Reactivity HazardRating	Δ	0.0554	Edit Reactivity Rating

From State: frmStandard.ReactivityHazardRating

To State	Source	Prob	Description
frmStandard.CASNo	Δ	0.0554	Edit Material CAS Number
frmStandard.PureChemical	p2=.01	0.0100	Edit Pure Chemical Flag
frmStandard.MaterialName	Δ	0.0554	Edit Material Name
frmStandard.NameSort	Δ	0.0554	Edit Sorting Name
frmStandard.SynSelect	Δ	0.0554	Select synonym from list
frmStandard.EPAID	Δ	0.0554	Edit EPA Id
frmStandard.RTECSNo	Δ	0.0554	Edit RTECS Number
frmStandard.MSDSType	p8=.01	0.0100	Edit MSDS Type
frmStandard.MSDSStatus	p9=.01	0.0100	Edit MSDS Status
frmStandard.Req Outstanding	p10=.01	0.0100	Edit Requests Outstanding
frmStandard.MSDS Available	Δ	0.0554	Edit MSDS Available Flag
frmStandard.Cancel	p12=.05	0.0500	Cancel Modify Mode
frmStandard.Save	p13≥.3	0.3554	Save Modifications
frmStandard.HealthHazardRating	Δ	0.0554	Edit Health Hazard Rating
frmStandard.FireHazard Rating	Δ	0.0554	Edit Fire Hazard Rating
frmStandard.Reactivity HazardRating	Δ	0.0554	Edit Reactivity Rating

From State: frmStandard.ReqOutstanding

To State	Source	Prob	Description
frmStandard.CASNo	Δ	0.0554	Edit Material CAS Number
frmStandard.PureChemical	p2=.01	0.0100	Edit Pure Chemical Flag
frmStandard.MaterialName	Δ	0.0554	Edit Material Name
frmStandard.NameSort	Δ	0.0554	Edit Sorting Name
frmStandard.SynSelect	Δ	0.0554	Select synonym from list
frmStandard.EPAID	Δ	0.0554	Edit EPA Id
frmStandard.RTECSNo	Δ	0.0554	Edit RTECS Number
frmStandard.MSDSType	p8=.01	0.0100	Edit MSDS Type
frmStandard.MSDSStatus	p9=.01	0.0100	Edit MSDS Status
frmStandard.Req Outstanding	p10=.01	0.0100	Edit Requests Outstanding
frmStandard.MSDS Available	Δ	0.0554	Edit MSDS Available Flag
frmStandard.Cancel	p12=.05	0.0500	Cancel Modify Mode
frmStandard.Save	p13≥.3	0.3554	Save Modifications
frmStandard.HealthHazardRating	Δ	0.0554	Edit Health Hazard Rating
frmStandard.FireHazard Rating	Δ	0.0554	Edit Fire Hazard Rating

frmStandard.Reactivity Hazard Rating	Δ	0.0554	Edit Reactivity Rating

From State: frmStandard.MSDSAvailable

To State	Source	Prob	Description
frmStandard.CASNo	Δ	0.0554	Edit Material CAS Number
frmStandard.PureChemical	p2=.01	0.0100	Edit Pure Chemical Flag
frmStandard.MaterialName	Δ	0.0554	Edit Material Name
frmStandard.NameSort	Δ	0.0554	Edit Sorting Name
frmStandard.SynSelect	Δ	0.0554	Select synonym from list
frmStandard.EPAID	Δ	0.0554	Edit EPA Id
frmStandard.RTECSNo	Δ	0.0554	Edit RTECS Number
frmStandard.MSDSType	p8=.01	0.0100	Edit MSDS Type
frmStandard.MSDSStatus	p9=.01	0.0100	Edit MSDS Status
frmStandard.Req Outstanding	p10=.01	0.0100	Edit Requests Outstanding
frmStandard.MSDS Available	Δ	0.0554	Edit MSDS Available Flag
frmStandard.Cancel	p12=.05	0.0500	Cancel Modify Mode
frmStandard.Save	p13≥.3	0.3000	Save Modifications
frmStandard.HealthHazardRating	Δ	0.0554	Edit Health Hazard Rating
frmStandard.FireHazard Rating	Δ	0.0554	Edit Fire Hazard Rating
frmStandard.Reactivity HazardRating	Δ	0.0554	Edit Reactivity Rating

From State: frmStandard.Cancel

To State	Source	Prob	Description
frmMain.ActivateQuery RecordsAvailable	NA	1.0000	Cancel changes and perform another function

From State: frmStandard.Modify

To State	Source	Prob	Description
frmStandard.ActivateUpdateMode	NA	1.0000	Modify Standard Table Information

From State: frmStandard.Save

To State	Source	Prob	Description
frmStandard.Save.Yes	p1=4(p2)	0.8000	Save changes [Yes]
frmStandard.Save.No	p2=1/4(p1)	0.2000	Cancel Save [No]

From State: frmStandard.Save.Yes

To State	Source	Prob	Description
frmStandard.ActivateViewM	NA	1.0000	Activate Standard Window

From State: frmStandard.Save.No

To State	Source	Prob	Description
frmStandard.ActivateViewM	NA	1.0000	Activate Standard

ode			Window

From State: frmNew.MaterialName

To State	Source	Prob	Description
frmNew.NameSort	p1=.9	0.9000	Enter Material Name
frmNew.Cancel	p2=.1	0.1000	Cancel add new

From State: frmNew.NameSort

To State	Source	Prob	Description
frmNew.MaterialType	p1=.9	0.9000	Enter Material Type
frmNew.Cancel	p2=.1	0.1000	Cancel add new

From State: frmNew.MaterialType

To State	Source	Prob	Description
frmNew.Save	p1=.9	0.9000	Save new RECID
frmNew.Cancel	p2=.1	0.1000	Cancel add new

From State: frmNew.Cancel

To State	Source	Prob	Description
frmMain.ActivateNoQuery Records	NA	1.0000	Process RECIDS

From State: frmNew.Save

To State	Source	Prob	Description
frmNew.Save.Yes	p1=4(p2)	0.8000	Save changes [Yes]
frmNew.Save.No	p2=1/4(p1)	0.2000	Cancel Save [No]

From State: frmNew.Save.Yes

To State	Source	Prob	Description
frmMain.ActivateQuery RecordsAvailable	NA	1.0000	Activate Main Window

From State: frmNew.Save.No

To State	Source	Prob	Description
frmMain.ActivateQuery RecordsAvailable	NA	1.0000	Activate Main Window

From State: frmCopy.NewMaterialName

To State	Source	Prob	Description
frmCopy.NewMaterialName	Δ	0.0666	Modify Material Name
frmCopy.NewNameSort	p2=.5	0.5000	Modify Sort Name
frmCopy.NewType	Δ	0.0666	Modify MSDS Type
frmCopy.Save	p4=.3	0.3000	Save New RECID
frmCopy.Cancel	Δ	0.0666	Cancel Operation

From State: frmCopy.NewNameSort

To State	Source	Prob	Description
frmCopy.NewMaterialName	Δ	0.1750	Modify Material Name
frmCopy.NewNameSort	Δ	0.1750	Modify Sort Name
frmCopy.NewType	Δ	0.1750	Modify MSDS Type
frmCopy.Save	p1=.3	0.3000	Save New RECID

frmCopy.Cancel	Δ	0.1750	Cancel Operation

From State: frmCopy.NewType

To State	Source	Prob	Description
frmCopy.NewMaterialName	Δ	0.1750	Modify Material Name
frmCopy.NewNameSort	Δ	0.1750	Modify Sort Name
frmCopy.NewType	Δ	0.1750	Modify MSDS Type
frmCopy.Save	p4=.3	0.3000	Save New RECID
frmCopy.Cancel	Δ	0.1750	Cancel Operation

From State: frmCopy.Save

To State	Source	Prob	Description
frmCopy.Save.Yes	p1=4(p2)	0.8000	Save changes [Yes]
frmCopy.Save.No	p2=1/4(p1)	0.2000	Cancel Save [No]

From State: frmCopy.Save.Yes

To State	Source	Prob	Description
frmMain.ActivateQuery RecordsAvailable	NA	1.0000	Activate Main Window

From State: frmCopy.Save.No

To State	Source	Prob	Description
frmMain.ActivateQuery RecordsAvailable	NA	1.0000	Activate Main Window

From State: frmCopy.Cancel

To State	Source	Prob	Description
frmMain.ActivateQuery RecordsAvailable	NA	1.0000	Activate Main Window

From State: frmDelete.Verify.Yes

To State	Source	Prob	Description
frmMain.ActivateQuery RecordsAvailable	NA	1.0000	Create new recid and add to list

From State: frmDelete.Verify.No

To State	Source	Prob	Description
frmMain.ActivateQuery RecordsAvailable	NA	1.0000	Archive record and remove from list

From State: frmIng.ActivateViewMode

To State	Source	Prob	Description
frmMain.ActivateQuery RecordsAvailable	Δ	0.1400	Activate Main Window
frmIng.Cancel	Δ	0.1400	Close Ingredient Form
frmIng.SelectIngredient	p3=.3	0.3000	Select An Ingredient From List
frmIng.Modify	Δ	0.1400	Press Modify button and lock ingredient for update
frmIng.Delete	Δ	0.1400	Delete ingredient

			from list
frmIng.Add	Δ	0.1400	Add a new ingredient to list

From State: frmIng.ActivateUpdateMode

To State	Source	Prob	Description
frmIng.Save	p1=.3	0.3000	Save Changes
frmIng.Cancel	p2≤.1	0.0500	Cancel Changes and activate main form
frmIng.IngredientName	Δ	0.1083	Enter Ingredient Name
frmIng.IngredientCAS	Δ	0.1083	Enter Ingredient CAS
frmIng.RECID	Δ	0.1083	Enter Ingredient RECID
frmIng.Percent	Δ	0.1083	Enter Percentage of ingredient
frmIng.Comments	Δ	0.1083	Enter Ingredient Comments
frmIng.Evaluation	Δ	0.1083	Enter Hazard Evaluation information performed

From State: frmIng.Modify

To State	Source	Prob	Description
frmIng.ActivateUpdate Mode	NA	1.0000	Modify Record

From State: frmIng.Save

To State	Source	Prob	Description
frmIng.Save.Yes	p1=4(p2)	0.8000	Save changes [Yes]
frmIng.Save.No	p2=1/4(p1)	0.2000	Cancel Save [No]

From State: frmIng.Save.Yes

To State	Source	Prob	Description
frmIng.ActivateViewMode	NA	1.0000	Activate Main Window

From State: frmIng.Save.No

To State	Source	Prob	Description
frmIng.ActivateViewMode	NA	1.0000	Activate Main Window

From State: frmIng.Cancel

To State	Source	Prob	Description
frmMain.ActivateQuery RecordsAvailable	NA	1.0000	Cancel action and activate main window

From State: frmIng.Delete

To State	Source	Prob	Description
frmIng.Delete.Yes	p1=4(p2)	0.8000	Confirm Delete [Yes]
frmIng.Delete.No	p2=1/4(p1)	0.2000	Cancel Delete [No]

From State: frmIng.Delete.Yes

To State	Source	Prob	Description
frmIng.ActivateViewMode	NA	1.0000	Activate Ingredient Window

From State: frmIng.Delete.No

To State	Source	Prob	Description
frmIng.ActivateViewMode	NA	1.0000	Activate Ingredient Window

From State: frmIng.Add

To State	Source	Prob	Description
frmIng.Cancel	Δ	0.1000	Cancel New Ingredient
frmIng.Save	p2=.9	0.9000	Enter ingredient recid and save it

From State: frmIng.SelectIngredient

To State	Source	Prob	Description
frmIng.ActivateViewMode	NA	1.0000	Process Ingredient

From State: frmIng.IngredientName

To State	Source	Prob	Description
frmIng.Save	p1=.3	0.3000	Save Changes
frmIng.Cancel	p2≤.1	0.0500	Cancel Changes and activate main form
frmIng.IngredientName	Δ	0.1083	Enter Ingredient Name
frmIng.IngredientCAS	Δ	0.1083	Enter Ingredient CAS
frmIng.RECID	Δ	0.1083	Enter Ingredient RECID
frmIng.Percent	Δ	0.1083	Enter Percentage of ingredient
frmIng.Comments	Δ	0.1083	Enter Ingredient Comments
frmIng.Evaluation	Δ	0.1083	Enter Hazard Evaluation information performed

From State: frmIng.IngredientCAS

To State	Source	Prob	Description
frmIng.Save	p1=.3	0.3000	Save Changes
frmIng.Cancel	p2≤.1	0.0500	Cancel Changes and activate main form
frmIng.IngredientName	Δ	0.1083	Enter Ingredient Name
frmIng.IngredientCAS	Δ	0.1083	Enter Ingredient CAS
frmIng.RECID	Δ	0.1083	Enter Ingredient RECID
frmIng.Percent	Δ	0.1083	Enter Percentage of ingredient
frmIng.Comments	Δ	0.1083	Enter Ingredient

			Comments
frmIng.Evaluation	Δ	0.1083	Enter Hazard Evaluation information performed

From State: frmIng.RECID

To State	Source	Prob	Description
frmIng.Save	p1=.3	0.3000	Save Changes
frmIng.Cancel	p2≤.1	0.0500	Cancel Changes and activate main form
frmIng.IngredientName	Δ	0.1083	Enter Ingredient Name
frmIng.IngredientCAS	Δ	0.1083	Enter Ingredient CAS
frmIng.RECID	Δ	0.1083	Enter Ingredient RECID
frmIng.Percent	Δ	0.1083	Enter Percentage of ingredient
frmIng.Comments	Δ	0.1083	Enter Ingredient Comments
frmIng.Evaluation	Δ	0.1083	Enter Hazard Evaluation information performed

From State: frmIng.Percent

To State	Source	Prob	Description
frmIng.Save	p1=.3	0.3000	Save Changes
frmIng.Cancel	p2≤.1	0.0500	Cancel Changes and activate main form
frmIng.IngredientName	Δ	0.1083	Enter Ingredient Name
frmIng.IngredientCAS	Δ	0.1083	Enter Ingredient CAS
frmIng.RECID	Δ	0.1083	Enter Ingredient RECID
frmIng.Percent	Δ	0.1083	Enter Percentage of ingredient
frmIng.Comments	Δ	0.1083	Enter Ingredient Comments
frmIng.Evaluation	Δ	0.1083	Enter Hazard Evaluation information performed

From State: frmIng.Comments

To State	Source	Prob	Description
frmIng.Save	p1=.3	0.3000	Save Changes
frmIng.Cancel	p2≤.1	0.0500	Cancel Changes and activate main form
frmIng.IngredientName	Δ	0.1083	Enter Ingredient Name
frmIng.IngredientCAS	Δ	0.1083	Enter Ingredient CAS
frmIng.RECID	Δ	0.1083	Enter Ingredient RECID

frmIng.Percent	Δ	0.1083	Enter Percentage of ingredient
frmIng.Comments	Δ	0.1083	Enter Ingredient Comments
frmIng.Evaluation	Δ	0.1083	Enter Hazard Evaluation information performed

From State: frmIng.Evaluation

To State	Source	Prob	Description
frmIng.Save	p1=.3	0.3000	Save Changes
frmIng.Cancel	p2≤.1	0.0500	Cancel Changes and activate main form
frmIng.IngredientName	Δ	0.1083	Enter Ingredient Name
frmIng.IngredientCAS	Δ	0.1083	Enter Ingredient CAS
frmIng.RECID	Δ	0.1083	Enter Ingredient RECID
frmIng.Percent	Δ	0.1083	Enter Percentage of ingredient
frmIng.Comments	Δ	0.1083	Enter Ingredient Comments
frmIng.Evaluation	Δ	0.1083	Enter Hazard Evaluation information performed

From State: frmListing.ActivateViewMode

To State	Source	Prob	Description
frmMain.ActivateQuery RecordsAvailable	Δ	0.2000	Activate Main Window
frmListing.SelectMFR Listing	Δ	0.2000	Select a Listing from the available list
frmListing.Add	Δ	0.2000	Add a new Listing
frmListing.Delete	Δ	0.2000	Lock current record for modification
frmListing.Cancel	Δ	0.2000	Close Listing Window

From State: frmListing.ActivateAddMode

To State	Source	Prob	Description
frmListing.MFRNumber	p1=4(p2)	0.8000	Enter a manufacturers code
frmListing.Cancel	p2=1/4(p1)	0.2000	Cancel current changes

From State: frmListing.Delete

To State	Source	Prob	Description
frmListing.Delete.Yes	p1=4(p2)	0.8000	Confirm Delete [Yes]
frmListing.Delete.No	p2=1/4(p1)	0.2000	Cancel Delete [No]

From State: frmListing.Delete.Yes

To State	Source	Prob	Description
frmListing.ActivateView Mode	NA	1.0000	Activate Listing Window

From State: frmListing.Delete.No

To State	Source	Prob	Description
frmListing.ActivateView Mode	NA	1.0000	Activate Listing Window

From State: frmListing.Save

To State	Source	Prob	Description
frmListing.Save.Yes	p1=4(p2)	0.8000	Save changes [Yes]
frmListing.Save.No	p2=1/4(p1)	0.2000	Cancel Save [No]

From State: frmListing.Save.Yes

To State	Source	Prob	Description
frmListing.ActivateView Mode	NA	1.0000	Activate Listing Window

From State: frmListing.Save.No

To State	Source	Prob	Description
frmListing.ActivateView Mode	NA	1.0000	Activate Listing Window

From State: frmListing.Cancel

To State	Source	Prob	Description
frmMain.ActivateQuery RecordsAvailable	NA	1.0000	Cancel Listing window and activate main menu

From State: frmListing.Add

To State	Source	Prob	Description
frmListing.ActivateAdd Mode	NA	1.0000	Add a new Manufacturer for the product

From State: frmListing.SelectMFRListing

To State	Source	Prob	Description
frmListing.ActivateView Mode	NA	1.0000	View selected record

From State: frmListing.MFRNumber

To State	Source	Prob	Description
frmListing.MFRName	p1=4(p2)	0.8000	Enter a manufacturers name
frmListing.Cancel	p2=1/4(p2)	0.2000	Cancel current changes

From State: frmListing.MFRName

To State	Source	Prob	Description
frmListing.MFRPrepared [illegible]	p1=4(p2)	0.8000	Enter a manufacturers code
frmListing.Cancel	p2=1/4(p2)	0.2000	Cancel current changes

From State: frmListing.MFRPreparedDate

To State	Source	Prob	Description
frmListing.ReceivedDate	p1=4(p2)	0.8000	Enter date MSDS was received in office
frmListing.Cancel	p2=1/4(p2)	0.2000	Cancel current changes

From State: frmListing.ReceivedDate

To State	Source	Prob	Description
frmListing.Save	p1=4(p2)	0.8000	Save new listing entry
frmListing.Cancel	p2=1/4(p2)	0.2000	Cancel current changes

From State: frmTargetOrgans.ActivateViewMode

To State	Source	Prob	Description
frmMain.ActivateQuery RecordsAvailable	Δ	0.3333	Activate Main Window
frmTargetOrgans.Cancel	Δ	0.3333	Cancel Target Organs Window
frmTargetOrgans.Modify	Δ	0.3333	Select Modify button and enter update mode

From State: frmTargetOrgans.ActivateModifyMode

To State	Source	Prob	Description
frmTargetOrgans.Select Organ	Δ	0.1400	Select Target Organ From list
frmTargetOrgans.Toggle ANSI	Δ	0.1400	Toggle ANSI check
frmTargetOrgans.Toggle MFR	Δ	0.1400	Toggle MFR check
frmTargetOrgans.Comments	Δ	0.1400	Update target Organ Comments
frmTargetOrgans.Cancel	Δ	0.1400	Cancel changes
frmTargetOrgans.Save	p6=.3	0.3000	Save changes

From State: frmTargetOrgans.SelectOrgan

To State	Source	Prob	Description
frmTargetOrgans.Select Organ	Δ	0.1400	Select Target Organ From list
frmTargetOrgans.Toggle ANSI	Δ	0.1400	Toggle ANSI check
frmTargetOrgans.Toggle MFR	Δ	0.1400	Toggle MFR check
frmTargetOrgans.Comments	Δ	0.1400	Update target Organ Comments
frmTargetOrgans.Cancel	Δ	0.1400	Cancel changes
frmTargetOrgans.Save	p6=.3	0.3000	Save changes

From State: frmTargetOrgans.Modify

To State	Source	Prob	Description
frmTargetOrgans.ActivateM odifyMode	NA	1.0000	Select Modify Button to lock record for modify

From State: frmTargetOrgans.Save

To State	Source	Prob	Description
frmTargetOrgans.Save.Yes	p1=4(p2)	0.8000	Save changes [Yes]
frmTargetOrgans.Save.No	p2=1/4(p2)	0.2000	Cancel Save [No]

From State: frmTargetOrgans.Save.Yes

To State	Source	Prob	Description
frmTargetOrgans.ActivateViewMode	NA	1.0000	Activate Target Organs Window

From State: frmTargetOrgans.Save.No

To State	Source	Prob	Description
frmTargetOrgans.ActivateViewMode	NA	1.0000	Activate Target Organs Window

From State: frmTargetOrgans.Cancel

To State	Source	Prob	Description
frmLogin.Activate	Δ	0.5000	
frmMain.ActivateQueryRecordsAvailable	Δ	0.5000	Cancel changes and activate main menu

From State: frmTargetOrgans.Comments

To State	Source	Prob	Description
frmTargetOrgans.Select Organ	Δ	0.1400	Select Target Organ From list
frmTargetOrgans.Toggle ANSI	Δ	0.1400	Toggle ANSI check
frmTargetOrgans.Toggle MFR	Δ	0.1400	Toggle MFR check
frmTargetOrgans.Comments	Δ	0.1400	Update target Organ Comments
frmTargetOrgans.Cancel	Δ	0.1400	Cancel changes
frmTargetOrgans.Save	p6=.3	0.3000	Save changes

From State: frmTargetOrgans.ToggleANSI

To State	Source	Prob	Description
frmTargetOrgans.Select Organ	Δ	0.1400	Select Target Organ From list
frmTargetOrgans.Toggle ANSI	Δ	0.1400	Toggle ANSI check
frmTargetOrgans.Toggle MFR	Δ	0.1400	Toggle MFR check
frmTargetOrgans.Comments	Δ	0.1400	Update target Organ Comments
frmTargetOrgans.Cancel	Δ	0.1400	Cancel changes
frmTargetOrgans.Save	p6=.3	0.3000	Save changes

From State: frmTargetOrgans.ToggleMFR

To State	Source	Prob	Description
frmTargetOrgans.Select Organ	Δ	0.1400	Select Target Organ From list
frmTargetOrgans.Toggle ANSI	Δ	0.1400	Toggle ANSI check
frmTargetOrgans.Toggle MFR	Δ	0.1400	Toggle MFR check

frmTargetOrgans.Comments	Δ	0.1400	Update target Organ Comments
frmTargetOrgans.Cancel	Δ	0.1400	Cancel changes
frmTargetOrgans.Save	p6=.3	0.3000	Save changes

From State: frmAvid.ActivateViewMode

To State	Source	Prob	Description
frmMain.ActivateQuery RecordsAvailable	Δ	0.5000	Activate Main Window
frmAvid.Cancel	Δ	0.5000	Close Avid Part Number List

From State: frmAvid.Cancel

To State	Source	Prob	Description
frmMain.ActivateQuery RecordsAvailable	NA	1.0000	Activate Main Window

From State: frmStores.ActivateViewMode

To State	Source	Prob	Description
frmMain.ActivateQuery RecordsAvailable	Δ	0.5000	Activate Main Window
frmStores.Cancel	Δ	0.5000	Close Stores Window

From State: frmStores.Cancel

To State	Source	Prob	Description
frmMain.ActivateQuery RecordsAvailable	NA	1.0000	Activate Main Window

From State: frmHistory.ActivateViewMode

To State	Source	Prob	Description
frmMain.ActivateQuery RecordsAvailable	Δ	0.2500	Activate Main Window
frmHistory.Cancel	Δ	0.2500	Close history window
frmHistory.Print HistoricalMSRS	Δ	0.2500	Print Historical Reference Sheet
frmHistory.SelectRECID	Δ	0.2500	Select current record from list

From State: frmHistory.SelectRECID

To State	Source	Prob	Description
frmHistory.ActivateView Mode	NA	1.0000	Activate History Window

From State: frmHistory.Cancel

To State	Source	Prob	Description
frmMain.ActivateQuery RecordsAvailable	NA	1.0000	Activate Main Window

From State: frmHistory.PrintHistoricalMSRS

To State	Source	Prob	Description
frmHistory.ActivateView Mode	NA	1.0000	Activate History Window

From State: frmOptions.ActivateUpdateMode

To State	Source	Prob	Description
frmOptions.Cancel	Δ	0.1166	Cancel Update
frmOptions.Save	p2=.3	0.3000	Save Changes
frmOptions.Username	Δ	0.1166	Edit Username
frmOptions.DataSource	Δ	0.1166	Edit Datasource
frmOptions.Viewer	Δ	0.1166	Select Report Viewer
frmOptions.Printer	Δ	0.1166	Select Printer
frmOptions.Reportdir	Δ	0.1166	Define working directory

From State: frmOptions.Cancel

To State	Source	Prob	Description
frmMain.ActivateQuery RecordsAvailable	NA	1.0000	Activate Main Window

From State: frmOptions.Save

To State	Source	Prob	Description
frmMain.ActivateQuery RecordsAvailable	NA	1.0000	Save changes and close window

From State: frmOptions.Username

To State	Source	Prob	Description
frmOptions.Cancel	Δ	0.1166	Cancel Update
frmOptions.Save	p2=.3	0.3000	Save Changes
frmOptions.Username	Δ	0.1166	Edit Username
frmOptions.DataSource	Δ	0.1166	Edit Datasource
frmOptions.Viewer	Δ	0.1166	Select Report Viewer
frmOptions.Printer	Δ	0.1166	Select Printer
frmOptions.Reportdir	Δ	0.1166	Define working directory

From State: frmOptions.DataSource

To State	Source	Prob	Description
frmOptions.Cancel	Δ	0.1166	Cancel Update
frmOptions.Save	p2=.3	0.3000	Save Changes
frmOptions.Username	Δ	0.1166	Edit Username
frmOptions.DataSource	Δ	0.1166	Edit Datasource
frmOptions.Viewer	Δ	0.1166	Select Report Viewer
frmOptions.Printer	Δ	0.1166	Select Printer
frmOptions.Reportdir	Δ	0.1166	Define working directory

From State: frmOptions.Viewer

To State	Source	Prob	Description
frmOptions.Cancel	Δ	0.1166	Cancel Update
frmOptions.Save	p2=.3	0.3000	Save Changes
frmOptions.Username	Δ	0.1166	Edit Username
frmOptions.DataSource	Δ	0.1166	Edit Datasource
frmOptions.Viewer	Δ	0.1166	Select Report Viewer
frmOptions.Printer	Δ	0.1166	Select Printer

frmOptions.Reportdir	Δ	0.1166	Define working directory

From State: frmOptions.Printer

To State	Source	Prob	Description
frmOptions.Cancel	Δ	0.1166	Cancel Update
frmOptions.Save	p2=.3	0.3000	Save Changes
frmOptions.Username	Δ	0.1166	Edit Username
frmOptions.DataSource	Δ	0.1166	Edit Datasource
frmOptions.Viewer	Δ	0.1166	Select Report Viewer
frmOptions.Printer	Δ	0.1166	Select Printer
frmOptions.Reportdir	Δ	0.1166	Define working directory

From State: frmOptions.Reportdir

To State	Source	Prob	Description
frmOptions.Cancel	Δ	0.1166	Cancel Update
frmOptions.Save	p2=.3	0.3000	Save Changes
frmOptions.Username	Δ	0.1166	Edit Username
frmOptions.DataSource	Δ	0.1166	Edit Datasource
frmOptions.Viewer	Δ	0.1166	Select Report Viewer
frmOptions.Printer	Δ	0.1166	Select Printer
frmOptions.Reportdir	Δ	0.1166	Define working directory

From State: Software Terminated

To State	Source	Prob	Description
Software Invoked	NA	1	Invoke software

Appendix B

Table B. 1: Failure log for MSDS Usage test.

Script #	Failure Description	Trans.
1	Delete Ingredients failed	92
5	Material Name Search Failed to match last word in query	10
6	Target Organs Window does not fit on screen	77
19	New RECID not defaulting flags to 'N'	50
31	CAS# Search on ingredients failed	7
39	Not displ RTECS in Material Printout	10
43	' in synonym name	50
124	Pure chemical input not searching for duplicate RTECS number	49

Table B. 2: Test Script 1

Step	State	Action
0	Software Not Invoked	Click MSDS Icon to start application
1	frmLogin.Activate	Bring up login screen
2	frmLogin.EnterPassword	Enter valid password for displayed userid password
3	frmLogin.Login	Press login button an log into database
4	frmMain.ActivateFirst Time	Display the main MDI frame and initial menu
5	frmQuery.Activate	Activate Query Screen
6	frmQuery.PerformRECID Search	Enter valid RECIDS to search on
7	frmQuery.ValidSearch	Display list of material
8	frmQuery.SelectItemIn Grid	Select a record from material list
9	frmMain.ActivateQuery	Menu options change to

	RecordsAvailable	reflect a record has been selected
10	frmStandard.ActivateView Mode	Activate Standard Table in View Mode
11	frmStandard.SynSelect	Select a synonym from the list on the standard table form
12	frmStandard.UpdateSyn Button	Press update synonym button. Synonym text box appears
13	frmStandard.SynText	Modify the synonym text
14	frmStandard.OKSynButton	Save the modification. Text field disappears and new synonym now in list
15	frmStandard.ActivateView Mode	Perform another operation
16	frmStandard.UpdateSyn Button	Press update synonym button. Synonym text box appears
17	frmStandard.SynText	Modify the synonym text
18	frmStandard.OKSynButton	Save the modification. Text field disappears and new synonym now in list
19	frmStandard.ActivateView Mode	Perform another operation
20	frmStandard.Update SynButton	Press update synonym button. Synonym text box appears
21	frmStandard.SynText	Modify the synonym text
22	frmStandard.OKSynButton	Save the modification. Text field disappears and new synonym now in list
23	frmStandard.ActivateView Mode	Perform another operation
24	frmStandard.Cancel	Close standard table form
25	frmMain.ActivateQuery RecordsAvailable	Perform another operation
26	frmStandard.activateView Mode	Open standard table for current record
27	frmStandard.AddSynButton	Add A Synonym to list
28	frmStandard.SynText	Enter synonym text
29	frmStandard.OKSynButton	Save new synonym
30	frmStandard.activateView Mode	Open standard table for current record
31	frmStandard.Cancel	Close standard table form
32	frmMain.ActivateQuery RecordsAvailable	Perform another operation
33	frmQuery.SelectItemIn Grid	Select another record from search list
34	frmMain.ActivateQuery RecordsAvailable	Perform another operation
35	frmStores.ActivateView Mode	Bring up stores table form for currently selected material
36	frmMain.ActivateQuery	Perform another operation

	RecordsAvailable	
37	frmStandard.activateView Mode	Open standard table for current record
38	frmMain.ActivateQuery RecordsAvailable	Perform another operation
39	frmQuery.SelectItemIn Grid	Select another record from search list, make sure open forms reflect change
37	frmStandard.activateView Mode	Open standard table for current record
38	frmMain.ActivateQuery RecordsAvailable	Activate Main Window
39	frmQuery.SelectItemIn Grid	Select a RECID From query list
40	frmMain.ActivateQuery RecordsAvailable	Process currently selected recid
41	frmListing.ActivateView Mode	Bring up MFR Listings window
42	frmListing.SelectMFR Listing	Select a Listing from the available list
43	frmListing.ActivateView Mode	View selected record
44	frmListing.Cancel	Close Listing Window
45	frmMain.ActivateQuery RecordsAvailable	Cancel Listing window and activate main menu
46	frmListing.ActivateView Mode	Bring up MFR Listings window
47	frmListing.SelectMFRList ing	Select a Listing from the available list
48	frmListing.ActivateView Mode	View selected record
49	frmMain.ActivateQuery RecordsAvailable	Activate Main Window
50	frmListing.ActivateView Mode	Bring up MFR Listings window
51	frmListing.SelectMFR Listing	Select a Listing from the available list
52	frmListing.ActivateView Mode	View selected record
53	frmListing.Delete	Lock current record for modification
54	frmListing.Delete.No	Cancel Delete [No]
55	frmListing.ActivateView Mode	Activate Listing Window
56	frmListing.Cancel	Close Listing Window
57	frmMain.ActivateQuery RecordsAvailable	Cancel Listing window and activate main menu
58	frmMain.New	Create New RECID
59	frmNew.MaterialName	Enter Material Name
60	frmNew.NameSort	Enter Material Name
61	frmNew.MaterialType	Enter Material Type
62	frmNew.Cancel	Cancel add new

63	frmMain.ActivateNoQuery Records	Process RECIDS
64	frmQuery.Activate	Perform Recid Query
65	frmQuery.PerformRECID Search	Enter RECIDS to search on
66	frmQuery.ValidSearch	Perform search with valid RECIDs
67	frmQuery.Activate	Perform a different search
68	frmQuery.PerformCAS Search	Enter CAS numbers to search on
69	frmQuery.ValidSearch	Perform Valid CAS number search and locate data
70	frmQuery.Activate	Perform a different search
71	frmQuery.PerformRECID Search	Enter RECIDS to search on
72	frmQuery.ValidSearch	Perform search with valid RECIDs
73	frmQuery.SelectItemIn Grid	Select a current record from hit list
74	frmMain.ActivateQuery RecordsAvailable	Process currently selected recid
75	frmHistory.ActivateView Mode	Activate History Window
76	frmHistory.SelectRECID	Select current record from list
77	frmHistory.ActivateView Mode	Activate History Window
78	frmMain.ActivateQuery RecordsAvailable	Activate Main Window
79	frmHistory.ActivateView Mode	Activate History Window
80	frmHistory.PrintHistoric alMSRS	Print Historical Reference Sheet
81	frmHistory.ActivateView Mode	Activate History Window
82	frmHistory.SelectRECID	Select current record from list
83	frmHistory.ActivateView Mode	Activate History Window
84	frmHistory.Cancel	Close history window
85	frmMain.ActivateQuery RecordsAvailable	Activate Main Window
86	frmQuery.SelectItemIn Grid	Select a RECID From query list
87	frmMain.ActivateQuery RecordsAvailable	Process currently selected recid
88	frmQuery.SelectItemIn Grid	Select a RECID From query list
89	frmMain.ActivateQuery RecordsAvailable	Process currently selected recid
90	frmIng.ActivateViewMode	Bring up ingredients screen
91	frmIng.Delete	Delete ingredient from list

92	frmIng.Delete.Yes	Confirm Delete [Yes]

Result: Failure
Failure ID: 1

Table B. 3: Test Script 2

Step	State	Action
	Software Not Invoked	Click MSDS Icon to start application
1	frmLogin.Activate	Bring up login screen
2	frmLogin.EnterPassword	Enter password
3	frmLogin.Close	Abort Application
4	Software Terminated	Close application

Result: Passed

Table B. 4: Test Script 3

Step	State	Action
	Software Not Invoked	Click MSDS Icon to start application
1	frmLogin.Activate	Bring up login screen
2	frmLogin.EnterPassword	Enter password
3	frmLogin.Login	Login to database
4	frmMain.ActivateFirst Time	Bring main MDI form
5	frmQuery.Activate	Activate Query Screen
6	frmQuery.PerformRECIDSe arch	Enter RECIDS to search on
7	frmQuery.ValidSearch	Perform search with valid RECIDs
8	frmQuery.SelectItem InGrid	Select a current record from hit list
9	frmMain.ActivateQuery RecordsAvailable	Process currently selected recid
10	frmMain.Copy	Copy existing material to a new RECID
11	frmCopy.NewMaterial Name	Modify Material Name
12	frmCopy.Save	Save New RECID
13	frmCopy.Save.Yes	Save changes [Yes]
14	frmMain.ActivateQuery RecordsAvailable	Activate Main Window
15	frmStandard.Activate	Activate Standard Table in

	ViewMode	View Mode
16	frmStandard.AddSyn Button	Add A Synonym to list
17	frmStandard.SynText	Enter synonym text
18	frmStandard.OKSyn Button	Save new synonym
19	frmStandard.Activate ViewMode	Perform other standard table operation
20	frmStandard.Modify	Press modify button to lock record and enable modification
21	frmStandard.Activate UpdateMode	Modify Standard Table Information
22	frmStandard.Health HazardRating	Edit Health Hazard Rating
23	frmStandard.Cancel	Cancel Modify Mode
24	frmMain.ActivateQuery RecordsAvailable	Cancel changes and perform another function
25	frmHistory.Activate ViewMode	Activate History Window
26	frmMain.ActivateQuery RecordsAvailable	Activate Main Window
27	frmAvid.ActivateView Mode	Activate AVID Window
28	frmMain.ActivateQuery RecordsAvailable	Activate Main Window
29	frmMain.PrintMSRS	Print Selected RECIDS
30	frmMain.ActivateQuery RecordsAvailable	Activate Main Menu
31	frmListing.Activate ViewMode	Bring up MFR Listings window
32	frmListing.SelectMFR Listing	Select a Listing from the available list
33	frmListing.Activate ViewMode	View selected record
34	frmListing.Cancel	Close Listing Window
35	frmMain.ActivateQuery RecordsAvailable	Cancel Listing window and activate main menu
36	frmMain.Delete	Archive existing record
37	frmDelete.Verify.Yes	Verify Delete [Yes]
38	frmMain.ActivateQuery RecordsAvailable	Create new recid and add to list
39	frmIng.ActivateView Mode	Bring up ingredients screen
40	frmIng.Cancel	Close Ingredient Form
41	frmMain.ActivateQuery RecordsAvailable	Cancel action and activate main window
42	frmAvid.ActivateView Mode	Activate AVID Window
43	frmMain.ActivateQuery RecordsAvailable	Activate Main Window
44	frmOptions.Activate	Activate Options Window

	UpdateMode	
45	frmOptions.Save	Save Changes
46	frmMain.ActivateQuery RecordsAvailable	Save changes and close window
47	frmListing.Activate ViewMode	Bring up MFR Listings window
48	frmListing.Delete	Lock current record for modification
49	frmListing.Delete.Yes	Confirm Delete [Yes]
50	frmListing.Activate ViewMode	Activate Listing Window
51	frmListing.Delete	Lock current record for modification
52	frmListing.Delete.Yes	Confirm Delete [Yes]
53	frmListing.Activate ViewMode	Activate Listing Window
54	frmMain.ActivateQuery RecordsAvailable	Activate Main Window
55	frmMain.Copy	Copy existing material to a new RECID
56	frmCopy.Save	Save New RECID
57	frmCopy.Save.Yes	Save changes [Yes]
58	frmMain.ActivateQuery RecordsAvailable	Activate Main Window
59	frmHistory.Activate ViewMode	Activate History Window
60	frmHistory.Cancel	Close history window
61	frmMain.ActivateQuery RecordsAvailable	Activate Main Window
62	frmMain.Exit	Exit System
63	Software Terminated	Exit system

Result: Passed

Appendix C

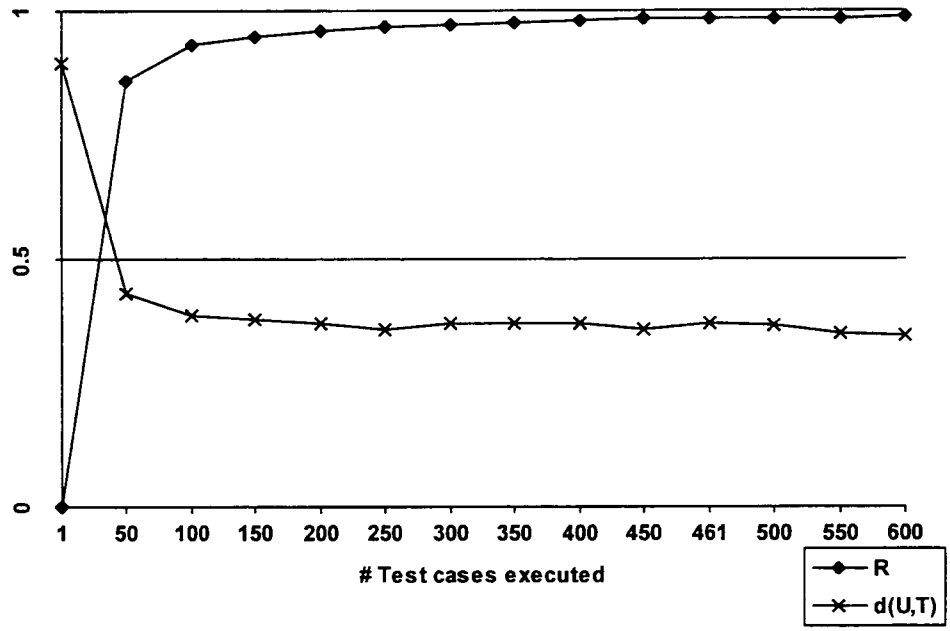

Figure C. 1: Reliability

VITA

Vita

Michael Muir was born in Corning, New York on March 17, 1965. He graduated from Horseheads High School in June of 1983. In 1987, he graduated from the Pennsylvania State University with a B.S. in the field of Computer Science. From 1987 to the present Mr. Muir has been employed by Lockheed Martin Energy Systems as a project manager and technical analyst. He graduated from the University of Tennessee with an M.S. in Computer Science in 1997.