DISTRIBUTION STATEMENT A
Approved for public release
Distribution Unlimited

UNIVERSITY OF CENTRAL FLORIDA
THESIS APPROVAL

Date: March 8, 1996

I hereby recommend that the thesis prepared and successfully defended under my supervision entitled: Interim Experiment Design Model for Verification and Validation of Training Simulators

by Christopher M. Hill

be accepted in partial fulfillment of the requirements for the degree of

Master of Science

from the Department of Industrial Engineering and Management Systems, College of Engineering.

Linda C. Malone, Chair

Robert L. Armacost

Michael D. Proctor

Ralph V. Rogers
Interim Department Chair

Jose A. Sepulveda
Director of Graduate Affairs

Martin P. Wanielista
Dean

FINAL EXAMINATION COMMITTEE

Diane M. Jacobs
Vice President for Research and Graduate Studies

The material presented within this report does not necessarily reflect the opinion of the committee, the College, or the University of Central Florida.

DTIC QUALITY INSPECTED 1

INTERIM EXPERIMENT DESIGN MODEL FOR VERIFICATION AND VALIDATION OF TRAINING SIMULATORS

by

CHRISTOPHER M. HILL
B.S. Auburn University, 1986

A thesis submitted in partial fulfillment of the requirements
for the degree of Master of Science
in the Department of Industrial Engineering and Management Systems
in the College of Engineering
at the University of Central Florida
Orlando, Florida

Spring Term
1996

19960502 093

ABSTRACT

This study centers on the use of interim experimentation for verification and validation of training simulators. The first portion of the study focuses on the development of a structured approach for interim experimentation for verification and validation of training simulators. The result is an Interim Experiment Design Model. The second portion applies the Interim Experiment Design Model to a training simulator currently under development.

The model is developed from current verification and validation theory, systems engineering principles, quality assurance theory, and experimental design techniques. This development results in a model with three steps. The user must initially determine the number of levels to include in interim experimentation. The next step requires development of the sample used in the experiment. The final portion provides progression rules for moving from one level to another. The model is illustrated by applying it to the Advanced Gunnery Training System.

The interim experiment design model provides a management tool that will improve the organization of interim experimentation and verification and validation efforts. This improved organization allows for detection of errors early in the product life cycle.

The efficiency of subsequent experiments will be enhanced as a result. The early detection of errors coupled with the improved efficiency will provide verification and validation at reduced cost.

Recommendations are made to continue research to find application methods for other tools available in acceptance sampling and experimental design theory.

To my girls, Barbara, Elizabeth, and Samantha, for their steadfast love and support.

ACKNOWLEDGMENTS

I would like to thank Dr. Robert Armacost and Dr. Michael Proctor for their direction and encouragement.

A special thanks goes to Dr. Linda Malone for her support, patience and motivation. Her influence and wisdom made the thesis process both educational and enjoyable.

TABLE OF CONTENTS

LIST OF TABLES viii

LIST OF FIGURES ix

CHAPTER 1 - INTRODUCTION 1

CHAPTER 2 - PROBLEM IDENTIFICATION 6

CHAPTER 3 - LITERATURE REVIEW 12
- Systems Engineering 13
- Verification and Validation 15
- Experimental Design 21
- Quality Assurance 24

CHAPTER 4 - MODEL DEVELOPMENT 30
- Determine Levels 32
  - Component Level 36
  - Integration Level 37
  - Systems Level 37
  - Acceptance Level 38
- Determine Sample Size 38
  - Balanced Complete Block Designs 40
  - Balanced Incomplete Block Designs 41
  - Partially Balanced Incomplete Block Designs 42
- Determine Progression Rules 43
  - Single Sample Plan 44
  - Double Sample Plans and Multiple Sample Plans 45

CHAPTER 5 - GENERAL MODEL DESCRIPTION 49
- Determine the Number of Experiments 53
  - Component Level 53
  - Integration Level 55

Systems Level 56
Acceptance Testing 57
Determine Sample Size 58
Balanced Complete Block Designs 59
Balanced Incomplete Block Designs 59
Partially Balanced Incomplete Block Designs 61
Determine Progression Rules 62

CHAPTER 6 - INTRODUCTION TO ADVANCED GUNNERY TRAINING SIMULATOR 68

CHAPTER 7 - MODEL IMPLEMENTATION 73
Determine the Number of Experiments 75
Component Level 75
Integration Level 76
Systems Level 77
Acceptance Test Level 78
Determine Sample Size 78
Balanced Complete Block Design 79
Balanced Incomplete Block Design 87
Partially Balanced Incomplete Block Design 90
Determine Progression Rules 94

CHAPTER 8 - CONCLUSION 98

APPENDIXES 101
A. Random Permutations 101
B. Index To Balanced Incomplete Block Plans 104
C. Table Of Partially Balanced Incomplete Block Designs 106
D. Normal Inspection Double Sampling Table 108
E. IAGTS Crew Training Matrix 110

LIST OF REFERENCES 112

LIST OF TABLES

1. Multiple Sampling Plan 64

2. Balanced Complete Block Design Example 83

3. Training Exercise To Factor Level Assignment Example 86

4. Partially Balanced Incomplete Block Design Example 92

5. Acceptance and Rejection Numbers Example 95

LIST OF FIGURES

1. IEDM Flow Chart Steps 1 & 2 50

2. IEDM Flow Chart Steps 3 & 4 51

3. AGTS Crew Level Training Matrix 71

4. Definition of BCBD Parameters 80

5. Selection of Random Permutation 82

6a. Permutation for Random Assignment of Exercise to Treatment 84

6b. Permutation for Random Assignment of Exercise to Treatment 85

7. IAGTS Matrix With BIBD and PBIBD Parameters 88

8. Balanced Incomplete Block Design Example 89

9. Selecting Partially Balanced Incomplete Block Design From Table 91

10. Comparison of Designs 93

CHAPTER 1

INTRODUCTION

Verification and validation is one of the most resource intense phases in the development of a training simulation system. It is also the most crucial to the success of a program. The purpose of this thesis is to develop a model that assists in the process of verification and validation. Verification and validation efforts are a part of the simulator's product development cycle that occur between the development of a prototype system and the system's production. Verification and validation efforts can be extended to a system after it is fielded. However, the approach used in this paper will strive for identification and resolution of issues prior to production. The verification and validation will be conducted through the use of interim experimentation.

Before the opportunities available for interim tests can be presented, an interim experiment must be defined in terms of its purpose, participants, and objectives. An interim experiment is an exercise involving both the producer and the user of a training simulator. It is conducted throughout the product development cycle, with a purpose of ensuring that the system under development is meeting some agreed upon

specifications or requirements. The exercises, or testing, should begin early in the product life cycle and should be conducted at stages throughout the product development cycle. The producer is the organization contractually responsible for the production of the device against detailed requirements or specifications. The user is the agency that needs the system under development to fill a specific training requirement. The user is the originator of the statement of requirements or specifications documents. There may be several organizations and players structured within the user agency. There may be a separate group detailed with the acquisition of the training system. There will be a specific training audience within the user group. There will also be several different types of subject matter experts within the user organization. These experts will be called upon at different intervals to ensure the training simulator is behaving as the replicated system does. Some of these experts may be experts in training devices or simulators. Some may be experts on a specific real system for which the training simulator is being developed. All of these participants will serve various important roles in the interim experiment process. The interim experiment can be conducted at different levels for different purposes. It can be a demonstration of the progress of the current development of a system, or it can be a concurrent engineering effort attempting to curtail changes late in the product life cycle. It can be a one time exercise for a prototype system that is used as a production milestone or it can be a sequential test following the test, fix, test methodology. The interim experiment can be

many things to many products. There is no exact solution for organization of every system.

According to Department of the Army Pamphlet 5-11 (1993) the purpose of verification of the system is to determine that the system functions as it was originally conceived, specified and designed, and that it meets the needs of the user as specified in the requirements documents. Department of the Army Pamphlet 5-11 (1993) states the purpose of validation of the simulator is to determine the extent to which the system accurately represents the intended real world phenomenon from the perspective of the intended user of the system. The interim tests should be used as a vehicle for verification and validation of the system under development. One of the greatest benefits to this interim testing organization is that the verification and validation will have the input and approval of both the user and the producer. Verification and validation theory will be discussed in greater detail in a following section.

Training simulators have a unique experimentation environment that is very controlled. An approach is needed to take advantage of the controlled environment and the benefits of interim experimentation. This structure will be developed into a model called the Interim Experiment Design Model (IEDM).

The paper begins with a description of interim experiments and a look at some of the deficiencies of current verification and validation efforts. These shortcomings are based largely on problems associated with traditional verification and validation techniques. This section will conclude with a look at recent interim experiment efforts.

A review of current literature will comprise the next section. The literature will be reviewed with the intent of finding areas that address the current verification and validation deficiencies. The literature will also provide motivation for different possible tools to use in the development of a structured approach.

Next, the tools and development theory that capitalize on the nature of interim experimentation and the training simulator's controlled environment will be developed into a concrete model, the IEDM. This section will be followed by a detailed description of the IEDM model. Throughout the paper several examples are provided with tank simulators. Tank simulators are a good example because of their complexity and differing levels of training application.

The true benefit of such a model is manifested in its application. The IEDM will be applied to the Advanced Gunnery Training Simulator (AGTS) program to show a real application example. The AGTS is a precision gunnery training simulator for main battle tanks and infantry fighting vehicles. This portion of the paper will begin with a brief description of AGTS and the difficulties in verifying and validating the simulator. The paper will also include the output of IEDM as applied to AGTS. The data concerning AGTS was gathered through personal observation of the program and its related testing, and through the review of documents associated with the simulator. The data generation was based on gathering information for an example program for IEDM, not in an official capacity reviewing capabilities of AGTS. Although the model illustrates a military application, it is not limited to the military environment. This

approach to verification and validation is applicable to all training simulator environments.

The contribution of this work lies in the development of a structured approach for progressive evaluation of a training simulator.. This contribution is an innovative application of existing tools to the verification and validation issue. Benefits of this work are derived from those associated with interim experimentation and those from a structured approach. Interim experimentation allows early involvement in the development cycle, brings the simulator user into the loop early, and helps reduce cost. IEDM is a flexible approach that integrates into existing acquisition models, organizes verification and validation efforts, reduces experiments' size and scope, and reduces cost. The structure of the model also allows for detailed numerical analysis. The paper concludes with recommendations for further development of the model and continued research.

CHAPTER 2

PROBLEM IDENTIFICATION

The original direction of this research was verification and validation of a complex training simulator, the AGTS. The work with AGTS and preliminary research efforts uncovered several problems with existing verification and validation techniques, at least in this particular case. The work also identified the need for a structured approach to verification and validation.

The first major problem with current verification and validation is a lack of organization of effort, or a guiding theory. This results in experiments that are poorly designed and cannot provide the desired quantified data, that answer the wrong questions, or that duplicate efforts. Pace (1995) suggests that many present verification and validation efforts become mainly judgment based. This condition results in many efforts for verification and validation containing unwanted variability. Consequently, simulation credibility is much less than desired (Pace, 1995). A structured approach is needed to ensure experiments are designed and organized correctly to achieve quantified results.

A second problem associated with traditional verification and validation efforts is that they are manpower and resource intensive. The result of this condition is that the verification and validation tests are often limited in scope because of cost. It is not unusual in some systems to find that a thorough verification and validation effort could exceed the cost of the product development (Gledhill, 1994). The cost of experimentation and of correction of errors is magnified the later in the product life cycle the experimentation is conducted. A structure guiding experimentation for verification and validation that starts early in the product life cycle and is used at discrete intervals will help alleviate the manpower requirements and the cost of experimentation and/or error correction. Several smaller scale tests conducted through the life cycle will also help develop a better product than one large test at the end of development.

Another general problem is that today's training simulators are built upon very large training matrices that are quite complex. As technology continues to move forward, the enumeration of all possible training scenarios may no longer be feasible because of length of time required and cost. If an experimenter tries to enumerate all of the possibilities, a hit or miss procedure will result, and the tester will not obtain a good representation of all training conditions that exist in the simulator's training matrix. Verification and validation tasks, like algorithm analysis that were once considered optional, are now mandatory to verify that applications are mapped properly to hardware. The result of more complex systems is that verification and validation

should be used earlier and throughout the entire life cycle (Dunham, 1989). A structure is needed to help obtain a representative group of exercises in experiments.

The lack of a standard guiding theory oriented specifically towards training simulators is another basic problem associated with verification and validation. There is a great deal of individual information available for verification and validation of software, for verification and validation of simulation models, and for system testing for specific production items. Training simulators involve a combination of each of these three fields, so when testing, theory from each field is applicable. A set of rules or a structure for combining the theory would be very helpful, because when each are integrated separately, problems can easily be created, or critical items may be overlooked.

An example of a recent application of interim exercises in a system is the Interim Advanced Gunnery Training System (IAGTS). This application is presented because it shows results of interim testing without a guiding theory, and provides the opportunity to show what results are possible with a testing model. This test will be used to show what information was captured in the exercises versus what information could have been obtained. The purpose of the interim experiment for the IAGTS was two-fold: final verification and validation of three simulators that were to be shipped to the user early for use as interim training devices and a progress check by the user on development of the overall AGTS system. The test was orchestrated to last over a two week period. The user brought in an independent agent to conduct the tests, and

soldiers from line armor units were brought in to man the simulators. The tests were organized into two phases. The first phase was designed to test each individual exercise to ensure it contained the correct target presentations, the correct target placement, and to check that each scenario was executed according to design. Testing of each exercise was conducted by enumeration. The second phase of the experiment was to check the system progressions rules and system management. In this scenario, each crew was started at the same point early in the training matrix, and the computer was allowed to move the crew through training sessions by matching their performance on exercises against rules of the simulator. The results of the IAGTS interim experiment are a good example of what can happen when testing is deferred to one big interim or final verification and validation test. There were four major issues discovered during this testing

The first problem was that the simulators being tested were not ready for extensive testing at the level desired by the test planners. Problems with the system surfaced as early as the first day because there was no methodology of a progressive check or test to ensure the system was ready for detailed testing. Examples of this type problem were those found in the warm-up and test procedures of the simulators, and a stack dump problem that would throw the crew completely out of the simulators training management records for no apparent reason, at largely undetectable intervals. A guiding structure could have required successful lower level testing prior to such a high level detailed experiment. The next problem was that too much of the testing was

deferred to the final test rather than gradual testing over the life cycle of the product. As a result, too many items were crammed into the final test for checking. This resulted in numerous problems with the system that impeded progress of the test. These problems were of a simple nature and should have been discovered prior to involvement of all of the participants of the experiment. Examples include faulty hit/kill templates that would not allow targets to be killed and machine gun failure on tanks during machine gun engagements. These errors did not allow completion of exercises. Again, a structure requiring lower level testing before advanced testing is needed. The next major problem was using the wrong level of experts for the experiment. The crews used in the test were naive and had little experience on the actual system. The result was crew members arguing with each other on whether the real system performed in a manner under question, and if the simulator was correct, or the tank itself behaved differently. More importantly, the crews did not have the skill required to manipulate the simulator at the high end of the training matrix. As a result, this section of the simulator was not really tested. A heuristic for test level selection and selection of participants could avoid this problem. Perhaps the most significant problem surfaced in the IAGTS test was the lack of an error correction system. The test was scheduled so close to the ship date of the simulators that the corrections to some of the identified problems were not verified effectively. The test should have been organized with the intent of allowing time for error repair verification. Progression rules should be a key ingredient for any standard experimentation theory. As alluded to earlier, all portions

of the training matrix were not successfully tested. This was due to accumulation of system error and lack of crew ability to manipulate the simulator. A guiding structure that provides a deliberate, balanced testing of selected portions of the matrix would have provided a picture of how well the entire matrix functioned.

The end result of the tests was that significant problems were uncovered by the testing, but the problem corrections were not verified prior to shipment of the simulators. Additionally, corrections made could have introduced errors down the line that did not originally exist. All original aspects of the test were not able to be conducted because of problems encountered early in the tests. As a result, a great deal of time and money was spent on a test that did not achieve its full potential. The credibility of the simulator was based largely on judgments, and those conclusions were obtained at a high cost.

The early research and experience with IAGTS have identified the need for a structure to guide systems through interim experimentation and verification and validation efforts. The structure should provide a guide for determining the numbers of interim experiments to conduct at specified levels. It should also provide a means to determine which parts of the training matrix to test in order to achieve a good representation at a reduced cost. A method of determining when testing at a particular level is acceptable, and when efforts should move to higher levels is also needed. The following chapter will search current literature to find areas that address these needs.

CHAPTER 3

LITERATURE REVIEW

The review of current literature will focus on four major areas: experimental design, quality assurance, systems engineering, and verification and validation. These areas were searched for two basic purposes. The first was to find an indicator of current and future verification and validation efforts. This part of the search will help reinforce the need for a structured verification and validation approach. The second purpose for the search was to find tools to use in the development and construction of IEDM. The quality assurance, experimental design, and systems engineering fields were explored for theory that could be developed to apply to the design and execution of interim experiments. All of the work explored in these areas was used to find new potential applications of existing theory. The verification and validation theory was searched to determine what significant contributions existed already and could be applied to the interim experiment model.

Systems Engineering

The initial focus of the literature search begins with the systems engineering field; it was used to help identify the need for interim experimentation. Blanchard's (1991) work on systems engineering provided a focus for testing throughout the life cycle. He defines systems engineering as developing and producing well integrated, cost effective, high quality systems with complete customer satisfaction as the objective. He provides a general framework to describe systems engineering in terms of objectives and their application to a given system. The author believes that one key to systems engineering is maintaining a "whole life cycle" perspective on any given system. Blanchard states that as system design and development progresses, there needs to be an ongoing measurement and evaluation process. Waiting until towards the end of the system development cycle, when the system is in a fully operational production model, will create great expense for any required change. Evaluation is a vehicle that provides high confidence that the system at hand performs ultimately as intended , and it should begin as early as possible. The beginning step of evaluation is the initial specification of system requirements. These requirements will develop into specific technical performance measures. The general steps involved in the evaluation effort are review of requirements, determination of methods for the evaluation effort and anticipated effectiveness of these efforts, and development of a comprehensive evaluation and test plan. The most important concept brought out in the text is the need for evaluation as early as possible to avoid the cost and difficulty associated with a drastic change late in

the production cycle. This work helps identify the need for a potential use of interim experimentation for validation and verification.

LaRocca (1965) provides a good framework on test planning and addressing objectives of the test. His work is a general reference for personnel concerned with the design and analysis of systems. The system testing section was based on the author's experience with a large military space system. Although his application seems specific, the application contains principles applicable to any system. The test procedure should be guided by a general test plan that provides direction for the testing effort. This will also ensure organization, allowing for results of early phases to be used to the maximum extent in the later phases of testing. The author calls for a test planning procedure that designates objectives for the tests, designates participating agencies and their responsibilities, provides for process and controls, and then creates a detailed test schedule. The objectives portion should be divided into primary and secondary levels. A significant contribution was a discussion of quality assurance and its application to the development process.

Blanchard's (1992) work on logistics engineering provides a reliability testing framework, and it also has a good section on test preparation and planning. The work is a unique combination of logistics and systems engineering principles that are oriented on a given system life cycle. According to Blanchard, system engineering begins as soon as a requirement is stated with the intention of designing a product to fulfill the requirement. Test and evaluation provide a means of determining how well the product

meets that requirement. Corrections are more economical and less difficult to implement when problems with meeting requirements are addressed as early as possible. Consistent with his other work, Blanchard stresses in several areas the importance of early assessment. The text provides the thought that test requirements be considered on an integrated basis to provide maximum benefit. One benefit of this work is the development of interim levels that ensure that all requirements should be tested together, if feasible. Another benefit of the work is a framework for test preparation and planning. Care should be exercised to ensure proper test conditions are met. These conditions include selection of the items to be tested, selection of personnel, preparation of facilities, and acquisition of proper support equipment and spare parts. It also emphasizes the importance of choosing personnel with an appropriate background and experience for the type of testing conducted. There must be great care exercised when choosing personnel to ensure they are qualified, but not to overkill for the type test being conducted. The section on reliability testing helps clarify the need for ensuring there is a clear understanding of what defines a successful system, and what defines a failure.

Verification and Validation

The next area in the literature search centered around review of current verification and validation principles. Dunham (1989) provides a look into the future of verification and validation. The author proposes, that to remain effective, verification and

validation technology must keep pace with the demands of the computer marketplace. Because of the complexity of the environment, it is imperative to incorporate verification and validation early in the life cycle. Verification and validation techniques like algorithm analysis that were once considered optional, are now mandatory to verify that applications are mapped properly to the hardware. In addition, there is a need to address new verification and validation tasks that address synchronization of this complex market. The future verification and validation technologies include standards emphasizing prevention, a versatile development environment, software verification and validation knowledge bases, increased use of formal verification, statistical quality control, more effective and efficient test techniques, guidelines on technique selection and combination, and complementary modeling and analysis techniques. The author concludes with the realization that the market pressure and demand for productivity and quality will make us forge ahead on verification and validation technology. The work shows a structured model will continue to be valid into the future.

Musa and Ackerman (1989) provide insight into quantifying software validation. They provide concepts of software reliability measurement and acceptance. They base much of their analysis on the assumption that software failures are a Poisson process. It is imperative that in the testing process the user define the difference between a failure and a fault. When outputs do not conform to requirements a failure has occurred. A fault is the underlying cause of the failure. To be most effective, reliability

measurement must be included early in the project. In the requirements phase, you must create system test and failure objectives. You must also define failures early in the process. The synopsis of the work is that in order to effectively validate software you must take advantage of both the fact that you are dealing with a vast number of input states and the fact, for commercial grade software only, that a small percentage of these will result in failure. These conditions warrant a rigorous statistical approach of software reliability measurements. The author's largest contribution is showing that verification and validation efforts can be, and should be, quantitatively analyzed.

Pace (1995) considers a new verification and validation proposal that relies more on automation of some of the standard verification and validation processes. The author states present verification and validation techniques are primarily judgment based and manpower intensive. The result is undesirable variability and limited scope of efforts due to cost prohibitions. Consequently, modeling and simulation credibility is sometimes much less than desired. Pace's proposal includes automation of information items and documentation throughout the verification and validation process. The result is an explicit statement of which portions of a simulation have been examined, and which have not yet been reviewed. Because of this, there will be increased incentive to review more of the simulation to enhance its credibility. The largest contribution of his work is to detail the current status of verification and validation efforts and the result of judgment based approaches.

The next portion of the literature review centers on exploration of possible tools to use for the development of a model. The first need for a tool is in determining how often and at what levels to conduct interim experimentation. Verification and validation theory was searched to find some of these type tools. Wallace and Fuji (1989) provide an excellent overview of software verification and validation techniques. The authors provide an overview of what verification and validation is, show how verification and validation efforts relate to other efforts, describe how to apply verification and validation, and summarize verification and validation effectiveness. Some of the authors' most significant contributions include a table of verification and validation tasks, methods for organization of the verification and validation group, and phases of verification and validation. The verification and validation test activities occur throughout all phases of the life cycle, but there are four major test planning activities. The first, component testing, verifies the design and implementation of software units on the lowest level. The next, integration testing, verifies functional requirements as the sub-elements are integrated. The focus at this level is interfaces. The third level is system testing. System testing validates the entire program against system requirements and performance objectives. Acceptance testing is the final level, and it validates the software against verification and validation acceptance criteria. The article concludes with the realization that verification and validation costs are more likely to be recovered when you start early in the life cycle at the requirements phase.

Banks (1989) provides methods for testing, understanding, and validating complex simulation models. Methods for accomplishing this were sought in four areas. Analogies from testing, understanding, and validating systems in general was one approach. The second was inferences from large specific systems. The next was investigation of the use of statistical methods in new ways, and the last was extension of current validation methods. The following contributions were significant from Banks' work. Models should be revalidated at projected intervals, and they should be designed with diagnostic features. The tester should utilize extreme condition tests whenever possible to evaluate performance of algorithms or models. Simulation runs should be conducted with all parameters and variables initialized across a wide spectrum of possible values. Two of the most applicable statistical tools mentioned were acceptance sampling and fractional factorial analysis. Extensions of common validation techniques include face validation, event validation, model assumption validation, comparison with historical data, comparison to other models, consistency checks, extreme conditions tests, sensitivity analysis, and Turing tests. Bank's work provides some principles that will be included in defining the levels and detail of interim testing.

Whitner and Balci (1989) provide a guide for selecting and using simulation model verification techniques. In the work, the authors provide several programmed model verification techniques, including informal, static, dynamic, symbolic, constraint, and formal. Informal analysis is the analysis through the employment of informal design

and development techniques. Informal analysis techniques includes desk checking, walk-through, code inspections, reviews, and audits. These techniques are valuable in the early stages of development because they rely on human reasoning to proof model logic. The same reliance on human reasoning can also be a disadvantage, based on knowledge and experience of the expert involved. Static analysis is analyzing characteristics of the static source code. Static analysis includes syntax analysis, semantic analysis, structural analysis, data flow analysis, and consistency checking. Static analysis can be automated and very effective when complementing other techniques. Static analysis does not have the ability to check whether the intentions of the modeler are being met. Dynamic analysis analyzes results gathered during model execution. Techniques include top-down testing, bottom-up testing, black-box testing, white-box testing, stress testing, debugging, execution tracing, execution monitoring, execution profiling, symbolic debugging, and regression testing. Dynamic analysis provides proof that, within certain context, a model is performing as intended. Its draw-back is that it can be very expensive; since cost of complete testing is not feasible, an adequate test must be determined. Symbolic analysis is analyzing the transformation of symbolic inputs to outputs along model execution paths. The symbolic execution techniques are symbolic execution, path analysis, cause-effect graphing, and partition analysis. Symbolic analysis is very costly and is not a stand alone technique. Constraint analysis is comparison of actual model execution state with assumptions. Constraint analysis techniques include assertion checking, inductive assertion, and

boundary analysis. The largest advantage of this class of techniques is the ability to verify that the model is functioning correctly according to its specification. Formal analysis includes formal mathematical proof of correctness. This type of analysis is not practical with most applications. The contribution of this work was providing techniques and tools to use in determining what type of testing to conduct at what levels in a model.

Experimental Design

The next area of theory searched for model development tools is experimental design. Experimental design is needed for finding ways to efficiently organize individual experiments. The first article provides a good example of how experimental design can benefit verification and validation efforts. Dzuiban, Curry, Knepell, and Riley (1992) explore synergism of simulation and experimental design. The authors state that organizations striving for world class process improvements will benefit most by combinations of these fields. The largest contribution of the authors is showing how design of experiment techniques are invaluable in verifying and validating complex simulations. A benefit to using design of experiments is achieving predictive power on the mean and on the variability by using a sequential, resource conserving approach. Design of experiments offers efficient ways to robustly test the model's reaction to varying inputs, allowing the responses to be compared with the expected, to help validate and verify the model. When complex simulation codes are tested at the

module, integration unit, and systems levels, the parallel concerns are the underlying conceptual model and coding. Design of experiments is an excellent tool for answering both concerns because designed experiments robustly cover the entire factor space.

Because of the large complex structure of most training simulators, experiments will have to be grouped, or blocked, in some manner during testing. In some cases, testing all possible treatment combinations may not be practical, even with blocking. The remainder of the experimental design literature search will identify tools that will assist in blocking experiments and reducing the number of treatment combinations sampled.

Bishop and Letner (1986) provide material taken from mainstream applied statistics and provide details on both the design and analysis of experiments. A key design consideration in verification and validation experiments is that the experimental units are not homogenous; we might have different results from different skill levels in a simulator. Because of this characteristic, blocking methods should be applied. The authors define a block as a grouping of an experimental unit which provides homogenous effects on a response. A complete block is a homogenous grouping of an experimental unit upon which the number of treatments appear equally often and at least once. Within experimental unit blocks, each block effect contributes equally to each treatment. This equal contribution is called balance. Balance can be achieved by using a balanced complete block design. The authors believe balanced designs can help keep the number of different information measures required to a minimum. They also have the feature that the difference of every pair of treatment parameters is estimated

with the same amount of information. In some cases the capacity of a block will be less than the number of treatments, or it may not be practical to include all treatments. Balanced incomplete designs provide a tool to use in these situations.

Montgomery's (1984) work on design and analysis of experiments is a good starting point in the theory of both balanced incomplete block designs (BIBD) and partially balanced incomplete block designs (PBIBD). As mentioned before, in some experiments you may not be able to run all treatment combinations in each block. Situations like this occur because of shortages of experimental apparatus or facilities, or because of physical size of the block. Balanced incomplete block designs are derived from randomized block designs where every treatment is not present in every block. When all treatment comparisons have the same importance, the combinations used should be selected in a balanced manner. The resulting total number of treatments in the balanced incomplete block design experiment will be less than found in a balanced complete block design. There are numerous existing designs; use of existing designs is a straight-forward method for design construction. One great restriction of this type design is the need for the number of times a pair of treatments appears in the same block to be an integer. These integer requirements can force the number of blocks, or replications, to be very high. The partially balanced incomplete block design provides a technique to avoid the large experiment size. It allows some pairs of treatments to be together 2 times, some to appear together 3 times,..., and some to appear together m times. Pairs of treatments that appear together m times are called m-th associates. The

design is said to have m associate classes. There are also numerous existing designs for partially balanced incomplete block designs that serve well for design construction.

Kempthorne (1952) provides a list of heuristics for determining a good partially balanced design. He gives a general overview of how experimental design fits in the field of statistics and on the statistical theory on which it was based, but his largest contribution was for PBIBD. The enumeration of PBIBD provides information concerning construction methods. The methods include geometrical configurations, application of finite geometry, the method of differences, and other methods. The other methods section contains some simple ways of obtaining designs given a set of specified conditions. The largest contribution of Kempthorne's work is realization of the difficulty of design generation. Use of existing design will be very efficient.

The purpose of the experimental design review was to search for tools for deciding which elements to test in an interim experiment. The focus now needs to move to deciding when an interim test has been successfully completed.

Quality Assurance

The last area of the literature search is concerned with finding methods for determining when a given interim test has been successfully completed. The quality assurance field was searched for appropriate tools. These tools provide motivation for approaches with less of an appetite for data. Quality assurance is a field rich in theory governing inspections and acceptance of lots of products. Ishikawa's (1990) work in

quality control provides a good overview of the quality field, and does a good job in relating the importance of applying the theory of quality to the industrial workplace. His work provides a general guide of quality and acceptance sampling theory. Ishikawa goes to great length to discuss the problems associated with 100% inspection. The direction of his work is concerned with general philosophy more than quantitative analysis, and this line of thought was primarily applied to the manufacturing arena. Different types of sampling inspections are classified according to number inspected, stage in product, inspection detail, method of judgment, inspected items useable or not, location, and selection of supplier. The next section deals with errors associated with inspecting less than 100% of a lot. The two types of sampling errors are consumer's risk - probability of accepting a bad lot, and producer's risk -probability of a good lot being rejected. The larger the size of inspection the smaller the levels of possible error; however, it also increases the cost. The objective is to identify the sampling plan that will optimize the technical and economic aspects, taking into account the desired quality, and the various probabilities and policies involved. The text also defines operating characteristic curves, inspection by attributes, and offers single, double, and sequential sampling plans. Ishikawa provides a discussion of cases in which a given plan is more appropriate, rather than give technical details of each plan. Another benefit of his work was a list of situations when sampling inspections were preferred to 100% inspection, since 100% inspection tends to be imperfect. Sampling inspections are better when there are many characteristics to be inspected. Cost of sampling is

lower. Sampling is good when there is a desired level of percent defective of a lot. It is also good as a motivation for a producer to raise quality, practice quality control, and improve receiving inspections.

The next work used in quality assurance was Duncan's (1986) work on quality control and industrial experiments. The author provides many examples of acceptance sampling techniques, with his greatest contribution being the various methods provided for applying these methods. The text provides a discussion of statistical principles and procedures pertinent to the control of production process output quality and of research in general on quality problems. The specific parts of the text that were most applicable were lot acceptance sampling plans that focused on sampling by attributes. The plans include single sampling plan, double sampling plan, sequential sampling, and multiple sampling plans; each of these areas will be discussed in general terms. Acceptance sampling defines a future course of action, it does not control quality. Acceptance sampling by attributes is graded according to conforming or nonconforming. A single sample plan specifies the sample size to be taken, and the acceptance number for the sample. Based on the results of the single sample, the lot, or experiment, would be either accepted or rejected. Another concept of the single plan, and all others, is the Operating Characteristic curve. This curve shows how the probability of accepting lots varies with the quality of material presented for inspection. There are two types of curves; type A and type B. Type A is the most appropriate because it gives characteristics of a sampling plan for isolated or individual lots. The largest benefit is

that it tells us what to expect under certain conditions of quality, lot size and acceptance number. The curve can prove useful in this tool and others where a certain level of desired quality or risk are driving the plan. This OC curve concept will hold through all plans discussed.

The general procedure for a double sampling plan include the following parameters: n1, n2, c1, c2, and c3. The relationship of the c's is c1<c2<=c3. The procedure is as follows: Sample size n1 from a given lot. If c1 or less nonconformities occur then accept the lot. If more than c2 nonconformity's occur, reject the lot. If the number of nonconformity's is between c1 and c2, take a second sample from the lot. The size of the second sample should be n2. If in the combined samples the number of nonconformity's is c3 or less, accept the lot. If there are more than c3, reject the lot. A frequently used technique is to set c2=c3. An advantage of this method is curtailment of the inspection if the second sample is not to standard.

In the sequential sampling plan the number of samples to be taken is determined entirely by the process. There are several variations of this plan, but the general concept is after a sample is inspected, each conformity is assigned a "+" value and each nonconformity is assigned a "-" value. The sampling process is continued until either score reaches a designated level. The process is then stopped, and a decision is made concerning acceptance or rejection of the lot. An extension of this plan is the group sampling plan where a group of items comprises each sample.

Multiple sampling plans are derived from the group sequential sampling plans. Again there are acceptance and rejection boundaries for accumulated groups of samples. If at the completion of any stage the number of nonconformity's equals or falls below the acceptance number, the lot is accepted. If the number of nonconforming items equals or exceeds the rejection number, the lot is rejected. Otherwise, another sample is taken. This multiple decision procedure continues until a fixed number of samples is taken. At that point a decision to accept or reject the lot is made. The first sample is usually inspected 100%, and subsequent samples stop as soon as the rejection number is reached. This plan is very similar to the group sequential plan, but has some different properties that make it appealing. The first is that the computations for determining the acceptance/rejection regions are easier. The second is the finite number of steps for a lot decision.

Schilling and Sommers' (1988) work on quality control provides an overview of many different techniques for acceptance sampling. This work explores the theory of each procedure in great detail, providing good flow diagrams and examples of how to apply each acceptance sampling procedure. Another benefit of this work is the analysis of when a given technique is more appropriate than another. The scope of the text is to serve the quality function for every major department of an organization. The greatest contribution was the section concerning acceptance sampling which begins with a definition of acceptance plans classified by attributes and discusses the two divisions of those plans according to concerns focused on risk or level of quality. Also important is

a section on implementation of acceptance sampling procedures. It follows with a definition and general description of single, double, multiple, and sequential sampling plans. Lastly, a major contribution is the section providing a guide on selection of the most appropriate sampling plan.

The literature search substantiated the benefits of experimentation and verification and validation on an interim basis. It also provided a group of tools that can be developed into an interim experimentation model. The tool from verification and validation theory is a list of principles that will be used to define and organize the number of experiments. Experimental design provides tools for design of each experiment. From quality assurance, tools will be used to develop rules for progression of exercises. These tools will be developed further for a model in the following chapter.

CHAPTER 4

MODEL DEVELOPMENT

For interim experimentation to be valuable, it should be a sequential series of tests that helps achieve a system's verification and validation at various stages in the product life cycle. There is no exact solution to the number of experiments that should be conducted or the organization of each of the experiments. For a given interim experiment to be successful, it must have certain characteristics in terms of content and organization. The literature search provided many potential tools for the development of a model. This section of the paper will outline the methodology used to develop the structure of the interim testing technique by developing tools to implement in the model. This development will begin with an analysis of verification and validation theory used to determine the number of experiments to conduct. It will include a section outlining experimental design theory used for determining the size of each of the interim experiments. It will then conclude with an exploration of acceptance sampling theory to determine when a given test is successful and rules for progression to a new test.

The development of the model will differ from the theory discovered during the literature search. The verification and validation subjects studied in the literature search focused primarily on verification and validation of software, or verification and validation of simulation models. There was also research in systems engineering that provided a systems approach to life cycle testing. The first step in the model is a combination of rules and theory found in each of the three areas. Applicable portions from each of the literature areas were combined to form a tool that applies to training simulators. The next step in the model, determining sample size, also differs from research in the literature search. The experimental design techniques were discussed in theory with little application in the literature review. In the model, experimental design techniques are applied to the problem of taking a large training matrix and obtaining a smaller sample that is representative of all of the conditions found in the larger matrix. The tools developed will provide not only an application, but also a means of randomly selecting specific treatments that provide the level of balance desired. Acceptance sampling theory was explored in the literature review, and will be used to develop the third step in the model. The major differences between the theory obtained in literature and the application to the model are differences in producer and consumer risk and a departure from sampling discrete lots from a given process under study. In the model, sampling theory will provide motivation for progression rules at each level of testing. Since each level does not form a discrete lot for inspection, like in theory, the application of acceptance sampling techniques are not as rigid. If acceptance and

rejection numbers do not make sense, acceptable quality levels should be relaxed until they do. Relaxation is permitted because of the theory of cumulating error. This theory, as well as other differences, will be described in greater detail as a part of the model development.

Determine Levels

The first step of the model will be determination of the number of levels of interim exercises to conduct. Since the purpose of conducting these exercises is the verification and validation of the system under testing, principles of verification and validation and general systems engineering concepts will determine the number of levels of exercises to include. There can be no exact set of rules for making this determination; we will explore the theory of verification and validation to develop a set of heuristics. General theory of verification and validation will each be covered initially, and then specific levels involving both theories will be presented.

As stated earlier, the purpose of verification of the system is to determine that the system functions as it was originally conceived, specified, and designed, and that it meets the needs of the user as specified in the requirements documents (Department of the Army Pamphlet 5-11, 1993). The system will likely be comprised of many components all of which must be verified individually before integration into the simulator. There may also be obvious levels of how verifications should be conducted as a part of the individual components integration into the system. These levels of

individual component verification through integration into the system will form the basis for determining the number of levels of interim exercises to conduct. This type of verification should be conducted during each life cycle phase to ensure requirements are met from previous phases (Wallace and Fuji, 1989). The document that will most greatly assist in this process is a statement of requirements generated from the user. It will most likely provide a framework of logical steps to determine the number of levels to include. Some organizations, like the US Army, have these documents already embedded in their acquisition cycle. An important consideration in determining the number of verification levels is that verification should be largely completed prior to validation efforts.

The purpose of validation of the simulator is to determine the extent to which the system accurately represents the intended real world phenomenon from the perspective of the intended user of the system (Department of the Army Pamphlet 5-11, 1993). The validation effort is centered around both structural validation of the components of the simulator, and validation of the output of the simulator. An example of structural validation is ensuring the replicated crew compartment in a tank simulator emulates the real crew compartment within given specifications. An example of the validation of output data is ensuring the probability of kill for a given tank munition against a given target is the same in the simulator as in the real system. There are many different forms of validation. The level necessary will be determined by the type of simulator under development, time available, and cost considerations. Validation

efforts in interim experimentation will most likely focus on face validity. Face validity makes a simulator seem reasonable to a person who is knowledgeable about the actual system (Law and Kelton, 1991). However, the extent of validation chosen will drive the number of levels of experimentation. In general, the first step is determining the extent of the real world that the system will be validated against. An example of this type of an issue is whether validation will occur by a SME review, or by a detailed test of the system followed by a field test.

The natural flow is to complete verification prior to validation efforts. However, there is some ability to overlap the two levels. In an early verification experiment where all components of the system are integrated, some aspects of the structural validation of the system could occur. The concept of separating the two components of verification and validation results from the fact that large scale validation of the training ability of the system cannot be complete until the system is functioning correctly. For the model there are four basic levels of verification and validation (Wallace and Fuji, 1989). Component or module level is the first. An example of this level is the tank commanders station in the AGTS. The commander's station is a separate entity that must individually function properly prior to any type of integration. The next level is integration. This step involves integrating various components of the simulation system with the focus being on interface. Integration of the commander's station with the gunner's station in AGTS is an example of this level. This integration involves more than one component, but falls short of full system integration because all of the

elements of the trainer are not present in the integration level described. The third level is system testing. This level involves the entire program against system requirements and objectives. This level of verification and validation is most commonly associated with user exercises. An example of this level is the Limited User Exercise for the IAGTS described earlier in the paper. The last level is Acceptance testing. This level of testing uses the system in its operational environment to check whether or not the system meets acceptance criteria. An example of this is an operational test and evaluation conducted by the Operational Test and Evaluation Command (OPTEC) in support of an acceptance decision on any given Army simulator. The model can involve all four levels of experimentation. The designation of the number of levels on which to conduct interim experimentation should normally be less than the total number available because of cost. The user and contractor will have to agree on which issues warrant involvement by both parties, which should be left to the producer, or which should be left to the final tests. The involvement and participants at each level should be mutually agreed upon by both the user and producer. The number of levels that are used in a given development cycle for a simulator should also be agreed upon up front. Each of the levels of participation should include both user and producer representatives that are knowledgeable in the system and in testing methods. The key players from the user side in all cases are the subject matter experts. Additionally, the techniques used to verify and validate at each level should vary. Some techniques do not serve well as "debugging" types, so they

should only be applied at specific levels (Whitner and Balci, 1989). Each level will now be discussed in terms of techniques used for verification and validation and the participants involved.

Component Level

Testing at the component level should involve techniques that will readily identify errors in logic or code of a given component. Static analysis is a good technique because it does not require the execution of the simulation model, and it focuses on the identification of semantics and syntax errors. In addition, informal analysis is a good choice. Informal analysis involves the walk through of code or review of code; a technique relying on human reasoning (Whitner and Balci, 1989). At this level the objective is to ensure each individual component is functioning correctly prior to its involvement in the larger system. When an experiment is conducted at the component level, the participants should be those who are knowledgeable in the coding and logic required by the particular type of components. For the automated opposing forces example, this would include personnel knowledgeable about the type of coding used, as well as experts in opposing force doctrine needed to verify the methodology captured in the code.

Integration Level

The integration level of verification and validation requires different techniques than the previous level (Whitner and Balci, 1989). Here a dynamic approach is more applicable. The desire is to conduct a "bottom up testing approach" that begins at the component level and integrates from the bottom up. In this type testing, debugging techniques are more important, especially concerning interfaces between components. At the integration level, a different type of expert may be required. At this level you are not as concerned with individual code lines, so emphasis needs to shift from pure software experts to experts familiar with how the system behaves in similar real environments. These experts should also be more aware of how the real system reacts to inputs that are tested as various components are integrated.

System Level

Techniques used at the system level should now focus on 'black box' type approaches. This type of dynamic analysis is more concerned with the output of the box given certain inputs rather than what is happening inside the box. Also, this level should involve stress testing and extreme conditions. At the system level, the scope of participation is largest. In this case you need experts on the simulator, experts on the real system, and experienced users in the field. The requirement to stress the system to check how it responds to extreme conditions will require a high quality expert; a novice may not be able to apply or interpret these conditions.

Acceptance Level

One of the objectives of conducting multiple interim experiments is to reduce or eliminate the requirements for a formal, large scale acceptance test, or at least reduce its size and scope. If, however, a formal level of acceptance testing is agreed upon, the techniques and participants should mirror those used in the system level.

Determine Sample Size

After the first portion of the model is complete, the number of experiments to conduct at each level and a total number of required experiments are derived. The second step in the model is determination of the number of entities to test at each level of the exercises. Consideration of the training simulator environment is required when seeking techniques to determine the sample size. A unique characteristic of the training simulator environment is its controllable nature. In many other verification and validation scenarios uncontrollable factors in the environment may cause undesirable variability and are a primary concern. Taguchi's philosophy or techniques within response surface methodology may prove useful in these scenarios (Myers and Montgomery, 1995).

One technique to determine a sample size is to conduct 100% inspection by testing every possible combination of exercise available. This technique is largely inefficient for several reasons. Testing every possible combination may not be possible because of

constraints on time, personnel, or budget. Because 100% inspection is not efficient, using this technique can drive up total cost of the verification and validation effort. Today's training matrices in most simulators are very complex, and enumeration of each possibility would be very difficult. In addition, the possibility of inspector error increases with 100% inspection (Ishikawa, 1990). Because of these problems, the nature of software code, and the integration of the code in the simulation, a method for testing only a portion of the exercise population is useful. Interim experimentation by its nature is well suited for testing a well represented portion of a simulator. The overriding concern is that whatever portion is tested needs to be representative of the population of exercises. The primary factor that allows testing a portion of the population is the controllable nature of the training simulator environment. As mentioned earlier, experimental design theory provides several tools for ensuring the fraction of exercises tested will be a good representation, and that if quantitative measurements are taken they can be analyzed statistically.

Before exploration of experimental design tools can be accomplished several terms must be defined relating the training simulator environment to experimental organization. Each simulator has a group of scenarios or exercises used to train an individual. This entire set of possible scenarios represents the population. These exercises will be organized to train individuals at given skill levels from a beginner to an expert. These skill levels are excellent candidates for blocking criteria. Within each skill level there will also be some organization of the exercises that moves from a less

difficult training scenario, usually at the beginning, to more difficult exercises, usually at the end of a skill level. These differing grades of difficulty represent treatments, or factor levels. When all treatment combinations are enumerated it will equal the population. Replication is the number of times we chose an exercise from a given block and factor level. The tools presented by the literature search were Balanced Complete Block Designs, Balanced Incomplete Block Designs, and Partially Balanced Incomplete Block Designs. The number of treatment combinations included in each decreases as you move from balanced complete to balanced incomplete to partially balanced incomplete. As a general rule the number of possible training exercises to include in a given experiment should be the maximum allowed by experimentation constraints. Each tool will now be developed for use in the model.

Balanced Complete Block Designs

A good description of Balanced Complete Block Design is contained in the work of Lentner and Bishop, 1986. In this technique block capacity is sufficient to support the desired number of experimental units. The desired number of treatments are randomly assigned to the experimental unit within each block. In addition, all treatments will occur equally often within each block. There are several advantages to using Balanced Complete Block Designs. The design provides straightforward analysis. Meaningful results may be obtained even with missing observations in some of the blocks. Results are more accurate because differences due to experimental units are eliminated from

treatment contrasts. Variability due to heterogeneous groups of experimental units, if a concern, is removed. There are no limitations on the number of treatments or blocks, so this technique is very flexible. If the desired number of treatments is very large homogenous blocks may be difficult to obtain or not practical. Investigation of the Balanced Incomplete Block Designs or Partially Balanced Incomplete Block Designs will be more appropriate.

To construct the Balanced Complete Block Design treatments are randomly allocated within the blocks. One of the easiest ways to accomplish the random allocation is by using random permutations. Tables of random number strings are an effective way of conducting the permutation. This construction method will be presented in greater detail in the following chapter. In some cases it is desirable to increase the block size to allow for replication. This replication translates to multiple samples of the same skill level and scenario difficulty in a training simulator. To accomplish this each treatment is randomly assigned to an experimental unit within each block. This continued randomization preserves balance.

Balanced Incomplete Block Designs

The concept of balance in the experiment is crucial in verification and validation efforts. If the capacity of the blocks will not allow the allocation of the desired number of treatments, then the Balanced Complete Block Design will not be practical. If this is the case or if the size or number of blocks is too large then the Balanced Incomplete

Block Design provides an acceptable alternative. As the term incomplete suggests, the design will not be as robust as a complete balanced design. The trade off is between size of the experiment and amount of information. If possible the experimenter should use the Balanced Complete Block Design. Some of the benefits of this design are that it allows analysis and replication, like the former design. In this design every pair of treatments occurs together the same number of times. There are numerous construction methods for Balanced Incomplete Block Designs. Tables of existing designs are readily available and will be applied to this model. This procedure will be discussed in greater detail in the following chapter.

Partially Balanced Incomplete Block Designs

Balanced Incomplete Block Designs do not exist for all combinations of parameters because they require too many blocks or blocks that are too large (Montgomery, 1984). The excessive size or amount of blocks is driven by the need to balance the design. The sizes are determined to be too large because they require a level of commitment of money or time that is not acceptable. In this case Partially Balanced Incomplete Block Designs offer a good alternative. There are trade-offs associated with choosing this technique. The design does provide some degree of balance, but not as much as in a Balanced Incomplete Block Design. In addition, there is not as much information present as in the Balanced Complete Block Design. The benefits of Partially Balanced Incomplete Block Designs are the reduced size of the experiment and analysis of

results. In this design, pairs of treatments do not occur together an equal amount of time. The amount of time they occur together is determined by the associate class of the design. For the model a two associate class design is acceptable. There are numerous tables of existing two associate class designs, and using the tables will be the most straightforward approach to design construction. Details will be explored in the following chapter.

Once the appropriate design technique is selected, the number of exercises to include is generated. This number will be the sample size input required for the progression rule formulation.

Determine Progression Rules

The organization of the model provides a logical sequence of levels of experimentation for the verification and validation of the simulation system. It also provides a number of exercises to test within the total number available at each level. The next issue concerns rules for determining when it is appropriate to leave one level of experimentation and enter the next level, or complete the process. The rules established are based on quality assurance theory. The general concept is that testing at a designated level will continue until acceptable quality has been reached. The method chosen should allow a good system to proceed more rapidly than a poor system. If the system performs miserably at a given level, the testing should be terminated until the producer has had an opportunity to make detailed correction to the system. The general

theory of acceptance sampling will now be explored, and a subsequent tool will be provided from the theory of multiple sampling plans. The output of this step in the model is a set of rules for progression to the next level of experimentation.

The first area to explore is the quality assurance theory of acceptance sampling. Acceptance sampling defines a future course of action, it does not control quality, therefore it can provide rules for progression (Duncan,1986). Acceptance sampling by attributes grades according to conforming or nonconforming. These properties make this methodology appealing to the model.

Single Sample Plan

The single sample plan specifies the sample size to be taken and the acceptance number for the sample. The single sample in our case would be an individual experiment of n exercises that would be sampled. This number would be determined by the experimental design portion of the model. Based on the results of the single sample the lot, or experiment, would be either accepted or rejected. Acceptance in this case would signify that the particular interim portion under test has passed and the cycle can now move to the next series of experiments. If the lot were rejected, the current level of test would not be passed, and future tests against this level would be scheduled. The general theory of acceptance sampling is appropriate for the model. However, since the single plan does not provide a mechanism for determining number or extent of future exercises oriented on correcting deficiencies found, it is not as attractive as the

other tools. Another concept of the single plan, and all others, is the Operating Characteristic (OC) curve. This curve shows how the probability of accepting lots varies with the quality of material presented for inspection. There are two types of curves; type A and type B. Type A is the most appropriate for our case because it gives characteristics of a sampling plan for isolated or individual lots. Since our lot will equate to the total number of exercises in an experiment, this is appropriate. The largest benefit is that it tells us what to expect under certain conditions of quality, lot size, and acceptance number. The OC curve can prove useful for providing information on acceptance and rejection numbers and motivation for determining the scope of future exercises if the current exercise at a given level was not accepted. This OC curve concept is applicable for all plans discussed.

Double Sampling and Multiple Sampling Plans

The next group of plans are the double sampling and multiple sampling plans. The general procedure for a double sampling plan includes the following parameters: n1, n2, c1, c2, and c3. The relationship of the c's is c1<c2<=c3. The procedure is as follows: Sample size n1 from a given lot. If c1 or less nonconformities occur then accept the lot. If more than c2 nonconformities occur, reject the lot. If the number of nonconformities is between c1 and c2, take a second sample from the lot. The size of the second sample should be n2. If in the combined samples the number of nonconformities is c3 or less, accept the lot. If there are more than c3, reject the lot. A frequently used technique is

to set $\underline{c2}=\underline{c3}$ (Duncan 1986). Even though the theory from acceptance sampling does not exactly match our situation, this type of thought is ideal for our case. A variation to double sampling is the multiple sampling inspection plan. This plan is based on the same theory as the double sampling plan, but the number of successive samples to be carried out is a discrete number larger than two. Our situation does have a discrete lot size to sample. However, the number of exercises to check is determined from the previous step in the model, not purely by acceptance sampling theory. From theory, there should be a discrete lot with an unknown amount of nonconformities. There is risk associated for the producer; the risk of a good lot being rejected. There is also consumer's risk, which is the risk of accepting a bad lot. In our situation of interim experimentation, there is a discrete lot with unknown numbers of nonconformities, but the risks are shared. The risks of moving forward to another level with a bad lot are shared because of the carry over of error in the simulator. This accumulated error may not become apparent until late in the life cycle, and it could present major obstacles to both producer and user. The benefit of cumulating errors is the ability to relax acceptable quality levels in the early stages of experimentation. The lot could again equate to the total number of exercises in a training matrix. The sample number is derived from the portion of the model that uses experimental design. If the first sample results warranted a second sample, the producer will be allowed to make corrections (i.e., software code or interface), and proceed with the second sample after corrections. If the second sample allows the lot to pass, the next experiment would focus on the next

higher level of experiments. This type of multiple sampling with accumulating errors is ideal for a software based system. If the producer finds the fault and corrects it, the concept of accumulating errors will not affect the progress of the system to the next level. If the producer creates additional nonconformities when correcting a given fault, the cumulative error nature will not allow progress until an acceptable solution is made. An advantage of this method is curtailment of the inspection if the second sample, or subsequent sample, is not to standard. If the second sample failed, it would indicate an error or accumulation of errors of larger proportions. The producer would make detailed correction, and subsequent testing at the same level would be scheduled in the future. In addition, the procedure could be used in an iterative fashion after a second failed sample and subsequent detailed rework. In classic acceptance sampling theory, the parameters of this method are based largely on consumer and producer risk, and consumer and producer quality. Based on these parameters, acceptance and rejection numbers are given by standard tables (Schilling and Sommers, 1988). In our case, because of the nature of the shared risk and because of the model's generated lot size, the method will be embellished. The embellishment will be based largely on the Military Standard (Mil. Std.) 105D method (Duncan,1986). A critical component of the rule determination is classification of errors. There will have to be a determination of errors as minor or major, and these should be agreed upon by both parties and detailed in the test and evaluation plan. The major errors will be counted in the rule progression scheme. Classification as a minor error may allow for willingness to carry

over errors. Ideally the minor errors will be corrected prior to movement to the next level, but no formal test measures are included to verify them prior to movement. More detailed discussion of the impact of error classification is included in the next section outlining the model. Organization of the model in terms of acceptance number and critical issues and requirements is a crucial portion of the model. The critical operational issues should be the driving factor in the selection of acceptance and rejection numbers. If several different issues that are not closely related are tested simultaneously, they should have separate acceptance and rejection rules. If they are related issues, the co-existence of several issues can possibly drive the number of acceptable errors a little higher. The model will provide a guide to assign these values. This method will be detailed in the following section. Many organizations already have critical requirements imbedded in the acquisition process. An example of critical issues and requirements and classification of errors is as follows. A critical issue and criteria may deal with the effectiveness of a tank simulator's main gun operation in comparison to a real tank. If there is an error, and the simulator's main gun is not effective, testing should not continue. Another critical issue is realistic sound of the combat environment. If enemy artillery sound was not realistic, testing could continue with the intent of repairing the fault in the future.

Each of the above tools presented in model development will be refined into the IEDM in the next chapter.

CHAPTER 5

GENERAL MODEL DESCRIPTION

There are three steps in the Interim Experiment Design Model (IEDM). The first is determination of the number of interim exercises. This determination is based on verification and validation issues and systems engineering concepts. This step will provide the total number of experiments in the verification and validation effort, as well as the number at each level. The second step is determining the number of training scenarios to include in each experiment. This is driven by experimental design concepts, and will provide a balanced representation of large training matrixes. The last step deals with determining the rules for moving from one level to the next. The rules are based on quality assurance principles. These rules will provide a progression scheme without requiring detailed statistical analysis. The second and third steps will be applied to each individual experiment, at each level, determined from step one. The specific requirements and participants for each step will be discussed in detail along with the general description of each step. Flow charts are provided in Figures 1 & 2 to illustrate the process. The participants, in general, will vary from case to case as appropriate. There will always be three basic members in each case. The first is a

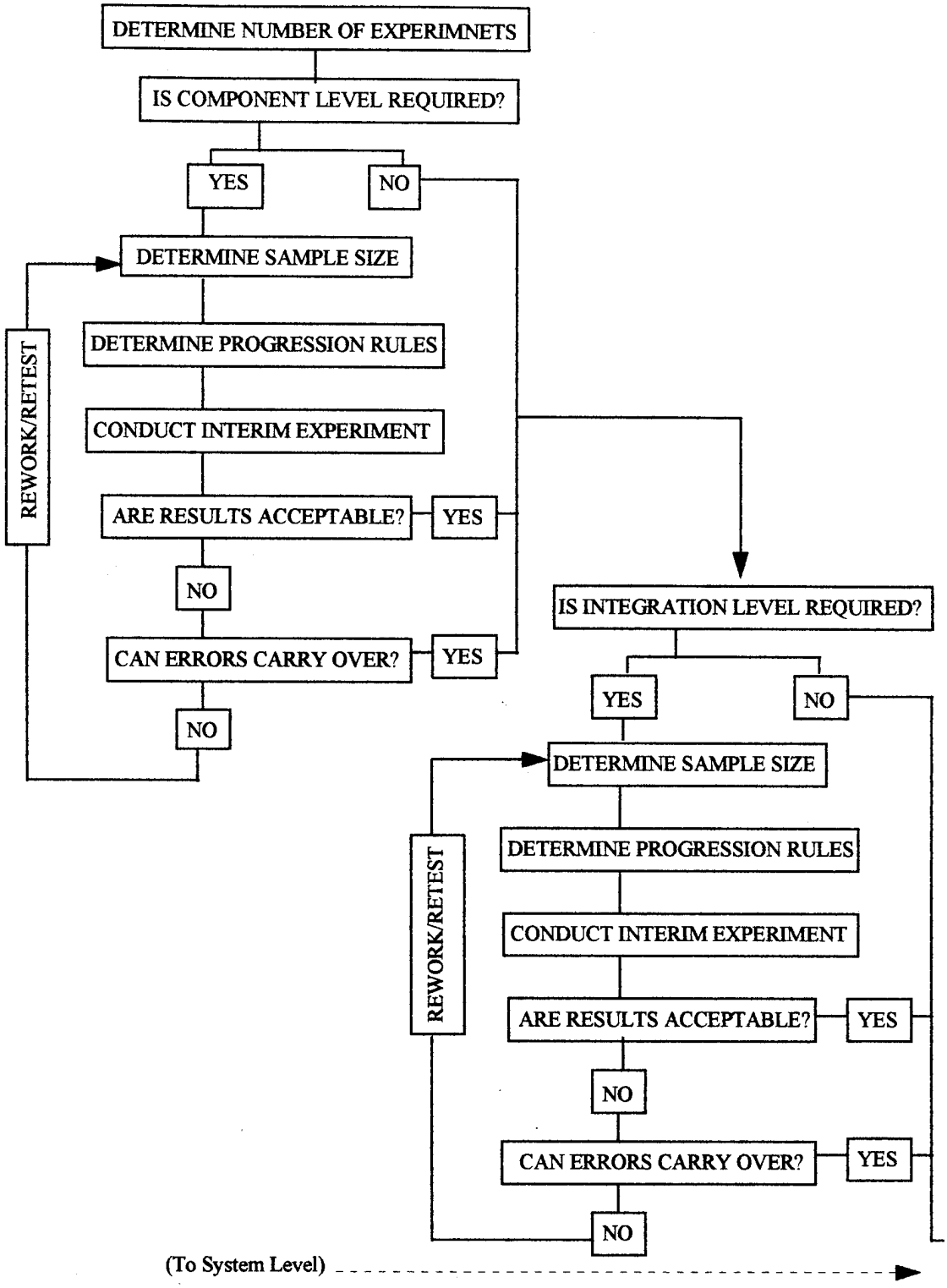


Figure 1. IEDM Flow Chart: Steps 1 & 2.

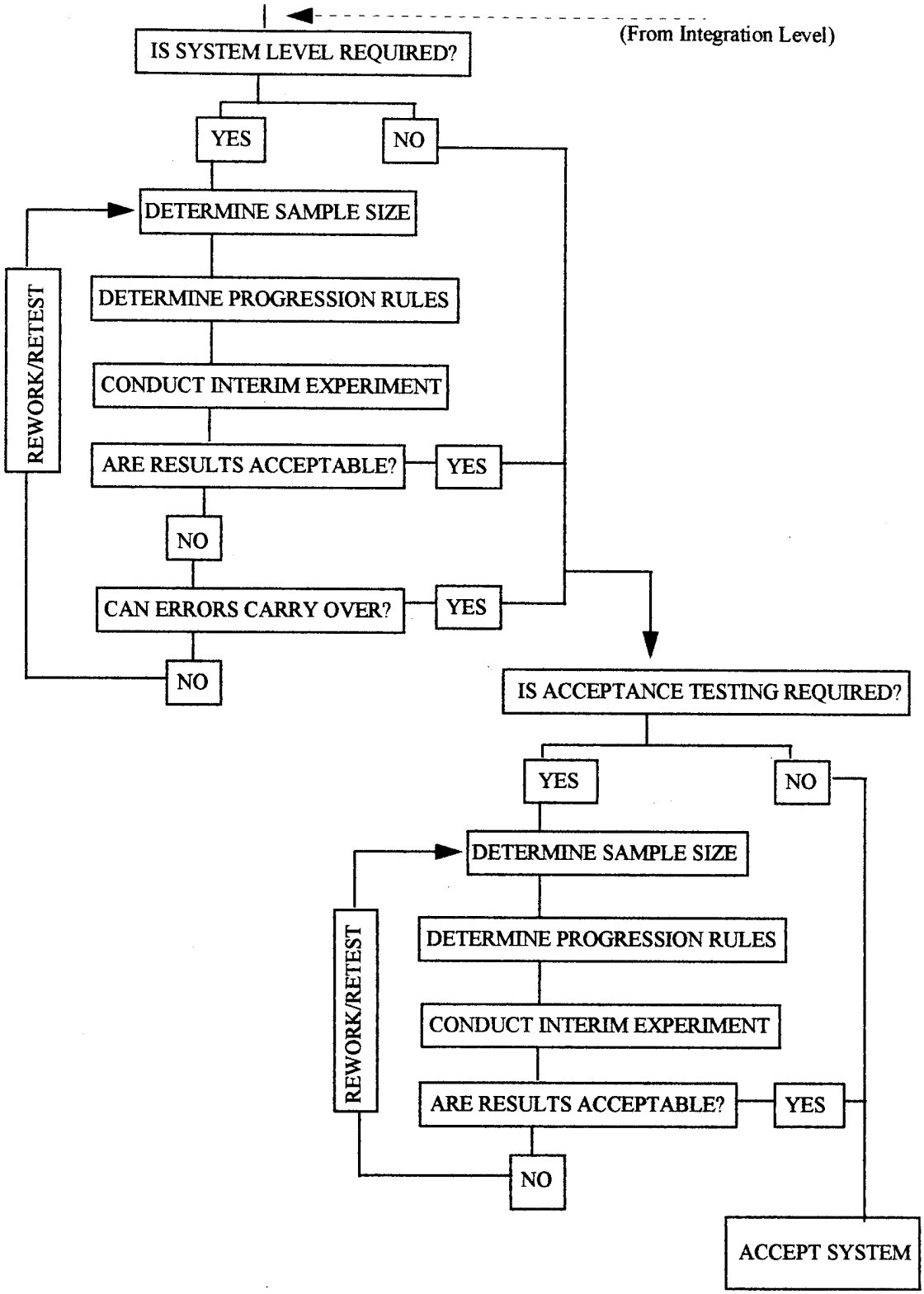


Figure 2. IEDM Flow Chart: Steps 3 & 4.

group of personnel representing the producer group. This group will vary with the level of interim experimentation conducted, but it should always have an individual designated as the leader of the producer team. The user group should have a similar team formed. Properties should be the same in terms of variable composition and leadership. Both teams should strive to keep the same person in charge of the testing team throughout all phases of interim experimentation. The last member is the independent review team. Composition of the team should be determined jointly by the user and producer. The purpose of the team is to have an independent agent involved in the testing to ensure unbiased procedure. The independent review team should be responsible for the conduct of each interim experiment and for overall verification and validation efforts (Army Regulation 5-11, 1992). The team should be led by a testing team chief, who will be in charge of every test. Roles of each participant will be discussed in more detail through each step in the model.

Each step in IEDM should be documented in a master test plan. As each step in the model is processed, or as a change occurs, the plan should be adjusted accordingly. A critical component of the plan is the list of Critical Issues and Requirements (CIR). The CIR represent specific requirements or specifications of the simulator contractually agreed upon by the user and producer. They should form objectives for each interim experiment. The completion of all CIR will represent completion of verification and validation efforts.

Determine The Number Of Experiments

The first step in the IEDM is determination of the number and levels of interim experiments to conduct. The purpose of conducting these experiments is the verification and validation of the system under testing. The decision of the actual number of levels should be made jointly by members of the user and producer team. The testing team chief can act as a third party to help resolve conflicts, and provide guidance as required. A requirement that should be monitored closely, particularly by the user team members, is that all critical operational issues and criteria are tested at least once in the overall test plan. The following heuristics should be applied to determine the number and levels of interim exercises to conduct. The result of this step is the number of experiments that are required at each level.

Component Level

1. Component level is the first level analyzed. The producer team should develop a list of all major components of the training simulator. From this list, both teams will jointly determine which components are critical and warrant the participation of both the user and producer in verification and validation through the means of an interim experiment..

2. Each team should develop a list of personnel required for participation in the component level of testing. Each team leader is responsible for his own group. Lists should be agreed upon by both teams, so each team is happy with representation provided by the other group. From the user team, potential personnel include those who can readily check software code, methodology of the code, structural and static validation, and other details of the component nature. From the producer team, members who can readily explain and correct code errors, structural errors, or defend logic of the methodology of each component would be ideal candidates. This list will form the interim experimental exercise team for the component level. The producer and user teams should also agree on the role and participation of the independent review team at the component level.

3. Given the list of critical components to check and the requirements and availability of critical personnel on each team, the testing team chief should determine how many exercises are required to capture all of the required items. The team chief should present this data to the user and producer leadership for approval. This number is the number of experiments conducted at the first level.

The following example of a tank simulator should clarify component level issues. Components involved are the data base, used for storing information files; the gunner's power control handles, used for firing the tank; the tank commander's independent thermal viewer, used for acquiring and engaging targets; and the instructor/trainer's display panel, used for monitoring the training session. There will always be a large

number of individual components, but all are not critical. Testing the gunner's control handles is probably not a critical item for all members of the testing team, especially if the part is relatively common. If the commander's independent thermal viewer is a new addition to simulator, and has never been seen before by the user on a simulator, then it could be considered critical and worth detailed testing. The next level to consider is the integration level.

Integration Level

1. Integration level analysis is done after component level. The producer team should identify each phase of integration of components from individual components through complete integration. From this list both the user and producer team leaders should decide which phases are significant enough to warrant detailed experimentation. The testing team chief , again, resolves conflicts and offers advice.

2. The user team leader should identify personnel who are experts needed to compare integration and system response given specified inputs with responses of real systems. The list should include personnel familiar enough with the simulator to be able to explain accepted trade-offs. From the producer side, the team leader should include experts able to readily correct interface errors, and debug systems given indicator errors.

3. The testing team chief needs to compare the number of critical phases of integration to the number and quantity of required personnel from all sides in order to determine the number of required exercises at the integration level.

4. The following example relates integration level issues on a tank simulator. Examples of integration are combining the database system with the computer that drives the training management system. The result of this integration is the ability to provide an acceptable training exercise given a record of past performance by the trainee. Integration also includes combining the power control handles, the ballistic computer, laser range finder, and primary sighting system in the gunner's station. The specific code issues of the computer interface may be of less interest to the user than the end result of what happens when he pulls a trigger in the gunner's station.

System Level

1. System level requirements are checked after integration level. From the available training menu and the detailed system requirements and specifications, each team leader should determine which training components require detailed testing that involves both the user and producer. The independent review team should be included in this decision.

2. Each team leader is responsible for deciding which experts and participants will be needed at each level of the training menu to effectively compare the simulator with the real system, identify errors at the source and implement corrections, or discuss

trade-offs accepted by all participants. The testing team chief again plays a large role in resolving problems.

3. The testing team chief will decide how many experiments should be conducted based on the list of critical training menu checks and required personnel.

4. An example of this level of testing is testing the entire tank simulator system, at levels that both sides feel are required. The type of experiment will most likely involve the higher end and more difficult scenarios in the training matrix. Testing that stresses the system is the focus of this level.

Acceptance Testing

The benefit of interim experimentation is reduced requirements at the final testing level. The testing at this level should focus on critical verification and validation issues that have either been carried over from a previous level with errors, or have special interest from the producer or user. The team leaders and testing team chief should jointly decide which issues need to be tested again. An example of this level is checking the performance of a tank simulator with induced computer and primary sight failures on multiple long range moving targets in the high end of a training matrix.

The total number of interim experiments required is the total of each of the four levels. This number , as well as the scope and required participants, should be outlined and agreed upon formally in a master test plan. This plan should be a fluid document based on the fact that system and acceptance level testing requirements can be affected

by previous interim experiment performance. The last check in the first step of the IEDM model is ensuring that all written system specifications of requirements, or CIR, have been captured in at least one of the designated levels.

Determine Sample Size

The next step in IEDM is determination of the number of training scenarios to include in each experiment, and to develop the design of the experiment. The designs are used to provide a representative sample of a simulator's training matrix. Detailed hypothesis testing, or other rigorous statistical analysis, will not be required in this model because of the progression rules scheme. Each of the three tools presented for determining an experiment's sample size and organization require a group of the same initial inputs. These inputs are common to all of the techniques presented, but their nature and relationship vary. The first input is the number of blocks. For training simulators, blocking on skill levels is a logical choice. The number of blocks is equal to the number of skill levels present in the training matrix of the simulator. The second input is the number of factor levels. The number of levels of difficulty of exercises within a skill level is the factor level or treatment, and specifies the number of treatments in a training simulator example. The last input is the number of desired replications. This number is determined largely by the amount of time available for a particular experiment. The user and producer groups should jointly determine how much time is available for a given experiment, analyze how much time is required for

one replication, then determine the total number of replications desired. Detailed examples of the three design parameters will be provided in the implementation chapter.

Balanced Complete Block Designs

The first step in design construction is determination of factor levels and number of blocks. Step two requires the testing team chief to designate the desired number of replications. Step three involves the random assignment of treatments to blocks. Refer to the tables provided in Lentner and Bishop, 1986, or to the excerpt provided in Appendix A. The experimenter must start with any arbitrary point in the table. The number of permutations picked should equal the number of blocks. The number of integers present in a permutation should equal the number of factor levels. If the number of integers available in the table is greater than needed, ignore integers above the number of factor levels in the experiment. Continue this process for all required replications. To chose a specific exercise within a given block and factor level, use the same random assignment procedure.

Balanced Incomplete Block Designs

The input parameters for this design differ slightly from the balanced complete design. The design requires the number of factor levels, number of blocks, and number of replications. In addition, Balanced Incomplete Block Designs require the block

capacity. The relationship between these parameters is also different. The reason an experimenter departs from complete balanced design is because of constraints placed on either block capacity or number of blocks. One of these numbers is given by the constraint; the other must be determined. In this design the number of blocks multiplied by the block capacity provides the total number of training exercises present in the design. As in the above design, the testing team chief must consider the time available for a given experiment, and determine with the user and producer groups the number of experiments to conduct. The number of experiments desired will drive the unknown parameter and the number of replications.

Construction of Balanced Incomplete Block Designs by use of existing designs is straightforward. The first step is to find all parameters. These parameters are number of blocks, number of treatments, and block capacity. The number of treatments is given, as well as the constrained parameter. Use the desired number of training exercises to solve for the unknown parameter. Use the two given and one solved parameters to get the number of replications. The second step is to go to a table of existing designs. Beyer, 1968 provides a good table, and an excerpt of it is included in Appendix B. Given a set of parameters, the table will point the experimenter to a plan reference number. The third step is to use the reference number to find the plan description. The description will include organization of the experiment in terms of which treatments are present in given block and replication numbers. The last step is random assignment of a training exercise to each given block and treatment

combination. The random assignment technique used in the Balanced Complete Block Design is a good method.

Partially Balanced Incomplete Block Designs

The input parameters required of Partially Balanced Incomplete Block Designs are the same as those required in the preceding design. The parameters include number of factor levels, number of blocks, block capacity, and number of replications. As in the preceding case, one of the block capacity or number of blocks parameters will be constrained. Procedures for solving for the unknown parameter, and the number of replications, is done the same way. An added parameter to this design is the number of associate classes. The classes describe the number of times given pairs of treatments occur together. Two associate class structures are most common and are sufficient for IEDM. The most desirable feature of Partially Balanced Incomplete Block Designs is the reduced experiment size. This is achieved because some pairs of treatments appear less often than others.

Construction via existing designs is very similar to the Balanced Incomplete Block Design method. The first step is to find all parameters. The number of treatments and the constrained parameter are given. Use the desired number of training exercises to solve for the unknown parameter. Use the two given and one solved parameters to determine the number of replications. The second step is to go to an existing design table. Clatworthy, 1973 lists several good tables. Appendix C contains an example of

these tables. Given a set of parameters, the table will point the experimenter to a plan reference number. The third step is to use the reference number to find the plan description. The description will include organization of the experiment in terms of which treatments are present in given block and replication numbers. The last step is random assignment of a training exercise to each given block and treatment combination. The random assignment technique used in the Balanced Complete Block Design continues to be a good method.

It is important to remember to use the Balanced Complete Block Design if possible. If it is not practical, use the Balanced Incomplete Block Design. If that design is not practical, use the Partially Balanced Incomplete Block Design. Personnel that are responsible for establishing these designs on a regular basis should obtain a copies of the appropriate tables. The excerpts provided are intended for illustration purposes only. The number of training exercises included in the experiment is the sample size parameter required in the next step of IEDM.

Determine Progression Rules

The progression rule scheme provides a means of determining whether a given experiment passes or fails without relying on rigorous statistics. Because of the organization of IEDM, the sample size of each level tested, n, is determined from information provided by step 2. The lot size equates to the number of different exercises from a simulator's training matrix that could possibly be tested in a given

experimental level. The lot will be divided up into multiple samples of size n. This number is the one provided from the experimental design technique used earlier in the IEDM. The next required input for a multiple acceptance sampling plan is the number of samples to take, m. Because of the nature of interim experimentation, n and m will have an inverse relationship. If n is chosen at a high level, constraints affecting the length of the test will force m to be at a low level. Conversely, if n is chosen at a lower level, then m can be picked at a higher level. Because of the nature of the software code in a training simulator, and the organization of the training matrix, we desire a larger n. This will allow a better balance and representation of the training matrix. In most cases m will be determined by the nature of each experiment. Conditions affecting the choice include the number of days available, the money available, and the size of n previously selected An example is for a lot of 100 exercises, and if a crew can realistically shoot 20 in a day, $n = 20$ and $m = 5$. For most situations m will be in the range from 2-4. There are many models used to set up multiple acceptance sampling plans.

The characteristics common to each model are as follows. If at any stage the number of nonconforming items equals or falls below the acceptance number, the lot is accepted and the system can move to the next level of testing. If, during any stage, the number of nonconforming items equals or exceeds the rejection number, the lot is rejected. Rejection means that the current level of testing has failed. Testing, however, should continue until the lot is completely tested. This will ensure that an appropriate

balance of the training matrix is tested, and that all errors are identified at the given level. Once rejection has occurred, the participants must schedule future testing at the same level. If acceptance or rejection do not occur, another sample is taken. The model will allow repair of the system between samples. This multiple decision procedure continues until the m-th sample is tested, when a decision to accept or reject is made. This decision can allow carry over of errors to the next level if deemed appropriate. It should be made clear that the current level of testing has failed even if testing continues after the rejection number is reached.

Table 1

Multiple Sampling Plan

Cumulated Sample Size	Acceptance Number	Rejection Number
n	0	4
2n	2	6
3n	4	8
4n	6	10
5n	8	12
6n	10	14
7n	12	16
8n	15	16

An example of appropriate acceptance and rejection numbers for an 8 stage sampling plan is the Barnard-Enters-Hamaker model (Duncan 1986) in Table 1. The number of stages, m, again was determined by available time, and number of exercises to complete within a given level. The example highlights the fact that at the last stage a

decision is required to accept or reject. The acceptance and rejection parameters are determined through the building process of the multiple acceptance sampling plan.

There are multiple techniques for building multiple acceptance sampling plans. The most appropriate for the purposes of IEDM is determined through the average quality level (AQL). AQL is an abbreviation of the OC curve that describes the risks associated with the producer and user with a given acceptance sampling plan (Schilling and Sommers, 1988). AQL is defined as the maximum percent defective that can be considered satisfactory as a process average for the purposes of sampling inspection (Duncan, 1986). Although the theory is based on methods used for sampling a given production process, it is applicable to IEDM. AQL was devised for producer's protection with the intent of keeping producer's risk small. This is ideal for IEDM because in a given interim experiment, only a portion of the total system is under test; therefore, it is more logical to provide protection for the producer because of the possibility of a false picture of the entire system. The user is protected by the fact that the interim experimentation is sequential in nature. The forwarding of an error will compound its effect, and there is a great likelihood it will be discovered in the next level of more detailed testing. Forwarding of errors continues to be a risk equally shared by producer and consumer.

Definition of the errors is a critical portion of successful progression rules. The specific definitions should be agreed upon by user and producer. In general a catastrophic error is one that will not allow continued training on the simulator. A

major error is one that significantly impairs the ability to train on the simulator. A minor error is one that degrades the realism of the training session but does not prohibit further training. Examples of these errors will be included in the implementation section. Definition of these errors are required for selection of AQLs.

Because of the fact that we are using an AQL approach we are able to use the tables of acceptance plan construction established in Military Standard 105D. There are many effective methods for constructing a sampling plan; Mil. Std. 105D is a very effective and easy one to apply. An advantage of Mil. Std. 105D is that it allows different AQL for major and minor errors in the system, and it provides different plans for different levels of inspection. The first step in using Mil. Std 105D is selection of the AQL. It is the responsibility of the testing team chief to select the AQL, but the process should involve high levels of management from both the producer and user sides. A good AQL procedure for IEDM is to use 0% for catastrophic defects, 1% for major defects, and 2.5% for minor defects. The testing team can see how the levels of AQL affect the acceptance and rejection parameters, and based on experience, management desires, or other circumstances, these levels can be changed. The next step in building a plan is to decide an inspection level. In IEDM this should be a normal level (level 2 in tables), unless unusual circumstances warrant a more or less detailed choice. Given an appropriate choice of AQL, inspection level, and sample size, constructing the acceptance and rejection numbers is a matter of looking up the values in a Mil. Std. 105D table. A good source of the tables is Duncan (1986). The output of this step is

the acceptance and rejection numbers that form the progression rules for a given level of experimentation. As acceptance or rejection decisions are made, the master test plan should be updated and COIC verified to ensure their completion.

CHAPTER 6

INTRODUCTION TO ADVANCED GUNNERY TRAINING SIMULATOR

The true benefit of IEDM lies in the application of the model to a training system. In this section, IEDM will be applied to the Advanced Gunnery Training Simulator (AGTS). The AGTS and IAGTS have been discussed in general terms, and used for some basic examples in the previous sections of the text. This section will explore the simulator in more detailed terms. The following section will apply IEDM to the AGTS program to determine appropriate numbers and levels of experimentation to conduct, to choose a sample to test, and to determine progression rules for each experimentation level.

The AGTS is a precision gunnery training device for armored systems with limited tactical training capability. The simulator is a direct outgrowth of two previous tank simulators - Unit Conduct Of Fire Trainer (UCOFT) and Platoon Gunnery Trainer (PGT) (Pellegrino, 1995). The UCOFT was a tank simulator designed to train only precision training skills at the crew level. It utilized a complex training matrix where the crew entered at the bottom end, and slowly progressed exercise by exercise through

the entire matrix. All crews shot the same exercises, and a single crew would fire the same exercises on repeat visits to the same portion of the matrix. UCOFT relied on data files to portray a given exercise; there was no intelligent opposing force. The PGT was a simulator that networked four UCOFT devices together. This network created the ability to train soldiers at the platoon level on precision platoon gunnery skills, and limited tactical training skills. It used the same type of matrix and the same data files as the UCOFT.

AGTS is an evolution of both UCOFT and PGT. It was originated with the M1A2 family of main battle tanks. Since the changes in the turret of the M1A2 were so significant from the M1A1, the UCOFT shelter was no longer a practical device, and the need for a new simulator was born. The development of the AGTS was based largely on improving some of the systematic problems of the UCOFT. The simulator was designed to fit a much larger family of vehicles than UCOFT. The vehicles represented within the AGTS family are the M1A1 main battle tank, M1A2 main battle tank, M2A3 Bradley Fighting Vehicle, and the XM8 Armored Gun System. Each AGTS is a simulator that provides individual and crew precision gunnery training for commanders and gunners of the respective vehicle. The simulator also can train at the platoon level, much like PGT. AGTS is used to train target acquisition, identification, and engagement with the vehicles weapons using both primary and alternate fire control and sighting systems. The AGTS provides simulated mobile and stationary threats, single and multiple target arrays, in a realistic battle environment during day, night, and

reduced visibility conditions. The simulator consists of a crew station that resembles the commander's and gunner's position of the respective vehicle, and an instructor/operator's (I/O) station which initiates, monitors, controls, and provides feedback for each exercise. (Pellegrino, 1995).

The AGTS crew training program consists of special purpose skills, basic gunnery skills, advanced gunnery skills, with levels one through three, and sustainment skills. A diagram of the training program is included at Figure 3. The special purpose skills matrix is designed to train crews in ten specialized exercises as required. An example is coaxial machine gun gunnery. The basic gunnery skills level of exercises is designed to prepare a crew for live-fire gunnery. The advanced gunnery skills consists of three levels. Each level contains four levels of systems management: stationary own vehicle with stationary targets, stationary own vehicle with moving targets, moving own vehicle with stationary targets, and moving own vehicle with moving targets. Within each systems management level there are nine reticle aim levels. These levels include fully operational day; fully operational night; Nuclear, Biological and Chemical warfare (NBC) malfunction; stabilization system failure; laser rangefinder malfunction; gunner's primary sight, gunner's primary sight extension malfunction; manual/gunner's auxiliary sight conditions; commander's independent thermal viewer engagement; gunner's primary sight extension engagement; and random extra exercise. The sustainment level chooses at random from any of the advanced gunnery levels (Lockheed, 1995).

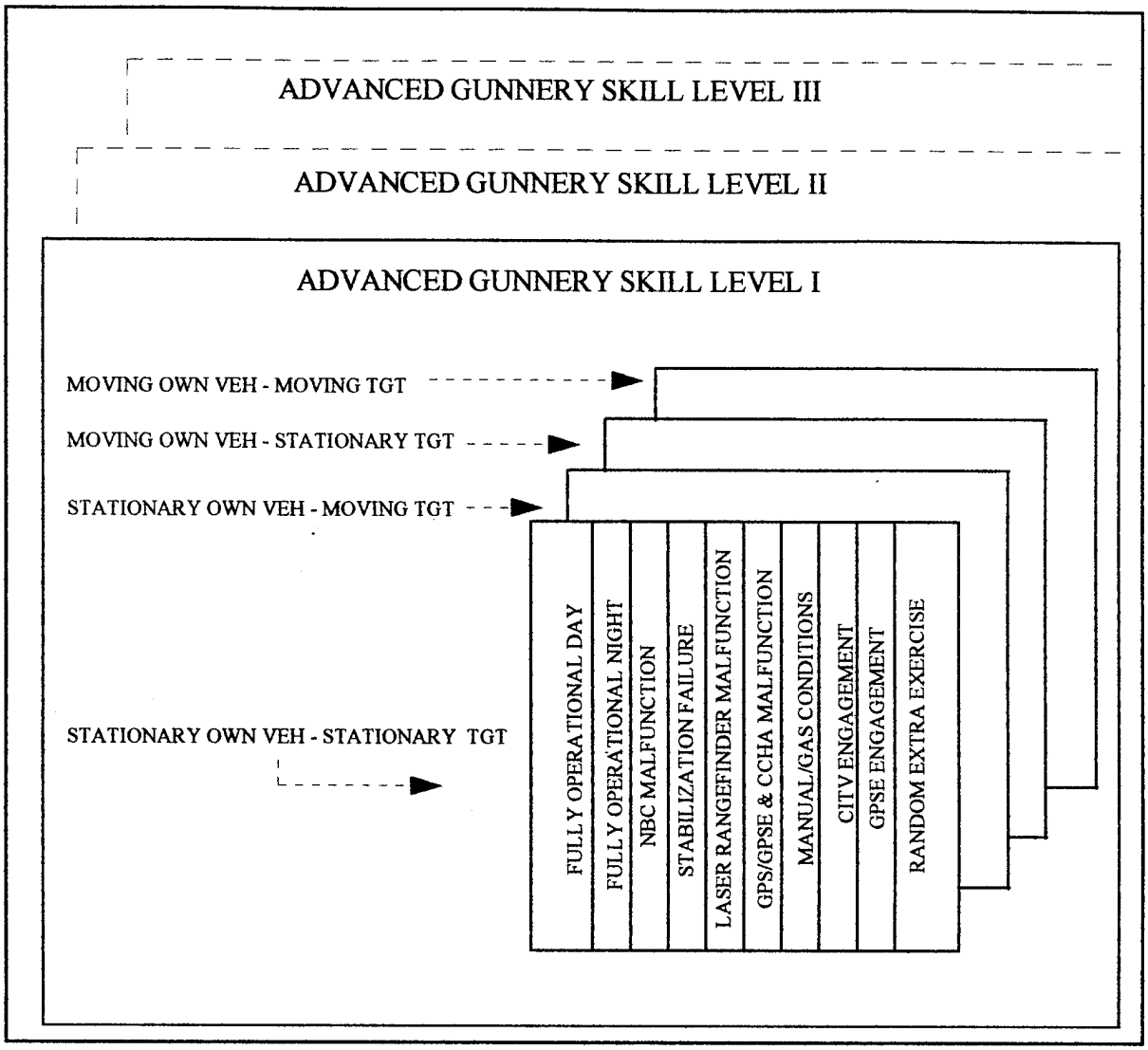


Figure 3. AGTS Crew Level Training Matrix.

The system management for the training device is complex. It allows for rapid progression of a crew that performs well and slower more deliberate progression for a crew that does not perform well. The simulator now randomly picks scenarios within a given exercise to ensure that the same crew does not see the same identical exercise twice. This randomization will complicate the verification and validation efforts. The

AGTS continues to use data files to represent opposing force scenarios; there is no intelligent opposing force. The system management and training program is similar at the platoon level. The AGTS is also required to be able to link into the Distributed Interactive Simulation network (DIS) (Lockheed, 1995). Verification efforts at crew training level will be complex. These efforts are now compounded because of requirements at the platoon and DIS levels. Given the complex nature of the simulator, enumeration of the training matrices at the three levels is not feasible. The cost and complexity of deferring verification and validation efforts until the end of the system life cycle are enormous. These conditions make AGTS a good candidate for implementation of IEDM.

AGTS is currently under development. One of the stipulations of the Army's contract for this simulator was that an interim system had to be fielded very early in the AGTS production cycle. The IAGTS was deployed to meet this requirement. The experiment described early in the paper was conducted for verification and validation of IAGTS to allow deployment. IAGTS uses a completely different training matrix than AGTS. AGTS will require a completely new verification and validation effort.

CHAPTER 7

MODEL IMPLEMENTATION

The purpose for implementing the model to a simulator under development is two-fold. First, implementation allows clear illustration of each step of IEDM. Second, the implementation will provide a mechanism for demonstrating the benefits of using IEDM. In this chapter, IEDM will be used on the AGTS simulator. The AGTS is a system that is currently under development. The example shown in this chapter is the sole work of the author. This work does not include any input by any agencies participating in the development or acquisition of the simulator. Each portion of the model will be illustrated, but the example provided will not cover the entire scope of application of the model to AGTS.

One of the benefits of IEDM is that it provides a tool for ensuring verification and validation through interim experiments. Rather than replacing current acquisition systems or models, IEDM fits within such existing models and enhances the acquisition process. In our example the US Army does have an existing acquisition system. Piplani, Mercer, and Roop (1994) provide a good general description of this process. The acquisition cycle begins with determination of a mission need, or determining that

a tank simulator for precision gunnery training is needed. The next phase is concept exploration and definition. Here a feasibility study is conducted, and the Army required Testing and Evaluation Master Plan (TEMP) is born. This TEMP includes many of the required inputs into IEDM. For example the TEMP requires statements called critical operational issues and criteria. These statements are the same as CIR required by IEDM. The next phase is demonstration and validation where multiple design approaches and technologies for candidate systems are examined. Engineering and manufacturing development occur as the next phase. The key phase for the purposes of IEDM is the production and deployment phase. After a contract is awarded to a company for a designated system, the prototype system must go through some required tests prior to production decisions. These tests are referred to as Operational Testing and Evaluation (OT&E). The interim experiments developed by IEDM are not designed to replace these required tests within the Army acquisition system. IEDM's interim experimentation is designed to supplement them. Verification and validation through interim experimentation will ensure the system is ready to pass the mandatory tests. The output of IEDM will fit nicely into the required TEMP as a verification and validation annex. The last phase in the acquisition cycle is Operations and Support. The benefits of a system that has been effectively verified and validated will be seen in this phase.

The participants included in the AGTS program are as follows. The producer is Lockheed Martin Corporation. The user is the US Army, but more specifically the US

Army Armor School. For the purposes of this paper the author will serve as the independent review team. In the actual acquisition system this role could be filled by personnel in the Army's Operational Testing and Evaluation Command (OPTEC). OPTEC is also responsible for the Army's mandatory testing described earlier. As we progress through the model, CIR will be formed in the appropriate steps.

Determine Number of Experiments

Component Level

In general because of limited funding, the US Army Armor School will be most willing to take risk in this level and rely on the producer to check components. Of the four main components in AGTS the one critical component is the crew compartment shell. It is critical because in this component there are trade-offs between cost and emulation of the actual tank's compartment. Participating members from the producer side will include representatives from the program management division, systems engineering personnel, and training department representatives. The user participants will include program management representatives and experts. The user subject matter expert should be extremely knowledgeable about the physical and functional characteristics of the tank compartment. The expert should be able to identify potential problems and verify whether stated trade-offs are acceptable or risky. This level can be completed in one experiment.

Integration Level

At this level the number of activities will increase. The participants from both sides should first be concerned with issues focused at crew level. When the crew level issues are resolved, the group should move to platoon level issues. Two integration level checks that are necessary are verifying target scenarios and verifying training management systems.

The first integration level checks should focus on verification of target scenarios. This general description should include numerous CIR. Example CIR are as follows.

1. The targets appear within 50 meters of the range specified in the scenario description.

2. The target is oriented in the same direction and has the same route of movement as specified in the scenario description.

3. All targets appear to be physically positioned in a manner realistic for actual armored vehicles.

4. Threat vehicles display identification features consistent with scenario specifications.

5. Targets can be destroyed by applicable weapons and munitions.

6. Targets can be destroyed at appropriate ranges for weapon types as specified.

Members participating from the producer will include program manager representatives, system engineering representatives, members from the training division, and software engineers. Participating members from user side will include

program management members, training experts, threat vehicle experts, and a subject matter expert on the tank system. This level of integration testing can be accomplished in one experiment.

Another integration level check would be verification of the training management system. CIR included are as follows. The crew progresses forward in the training matrix in accordance with procedures outlined in the specifications document. A crew that fails a scenario moves to additional training in accordance with the specifications document. Crews displaying unique problems on a consistent basis are recommended to special exercises in accordance with the specifications documents. The crew returns to the correct matrix position after an interrupted training session. Participants are generally the same on both sides. On the user side the expert from threat vehicles will not be needed. An expert capable of manipulating the simulator at a level that would stress all training management scenarios will be added. This integration check could be accomplished in one experiment. The total integration level experiments is two.

System Level

Once integration level checks are complete at crew and platoon level, a system level check will help prepare for the US Army's mandated test. CIR for this level will focus on any issues carried forward from previous experiments and on system level issues. An example is whether the system startup procedure can be accomplished within time limits provided in specifications documents. Another example is whether the system

adequately replicates friendly vehicle fratricide scenarios at the platoon level. Producer representatives should include representatives from all departments involved in the AGTS. User participants should include program management representatives, training representatives, and a group of subject matter experts. These experts should be tank qualified soldiers from line units. The line unit soldiers will provide a good level of face validation for the entire system. This system check can be accomplished in one experiment.

Acceptance Test Level

The Army's mandated OT&E prior to full scale production can be considered as an acceptance level test. In accordance with Army standard operating procedure, OPTEC is exclusively responsible for the organization and conduct of the test. The benefit of IEDM is all interim experiments have prepared for this level of test, and the system will have a much greater chance of good performance with less error.

The total number of experiments determined in the first step of IEDM is four. One at component level, two at integration level, and one at system level. The acceptance level test will not be counted because of the nature of the control of the exercise.

Determine Sample Size

To illustrate the second step in IEDM the model will be applied to the integration level experiment for checking the target scenarios. The model will begin with applying

the Balanced Complete Block Design to the AGTS simulator. The model will then be applied to the IAGTS simulator, for the same experiment, to illustrate Balanced Incomplete Block Designs and Partially Balanced Incomplete Block Designs. In each example provided, the assignment of blocks and treatments is arbitrary because we are not making numerical measurements. In this case we will rely on the progression rules for determining the success of an experiment.

Balanced Complete Block Design

Two of the input parameters necessary for construction of Balanced Complete Block Designs exist by virtue of the design of the simulator's training matrix. One parameter must be determined by the testing team chief by considering the first two parameter values and the amount of time available for the experiment. The first step in construction of this design is definition of the existing parameters. In AGTS, skill levels is a good blocking factor. There are three skill levels, so there will be three blocks. There are four levels of difficulty within each skill level. These four levels will equate to four factor levels or treatments. Assume for our experiment we have six testing days available. The relationship between the input parameters and the training matrix are shown in Figure 4. All available scenarios within a given exercise code and treatment level are assumed to be equally difficult.

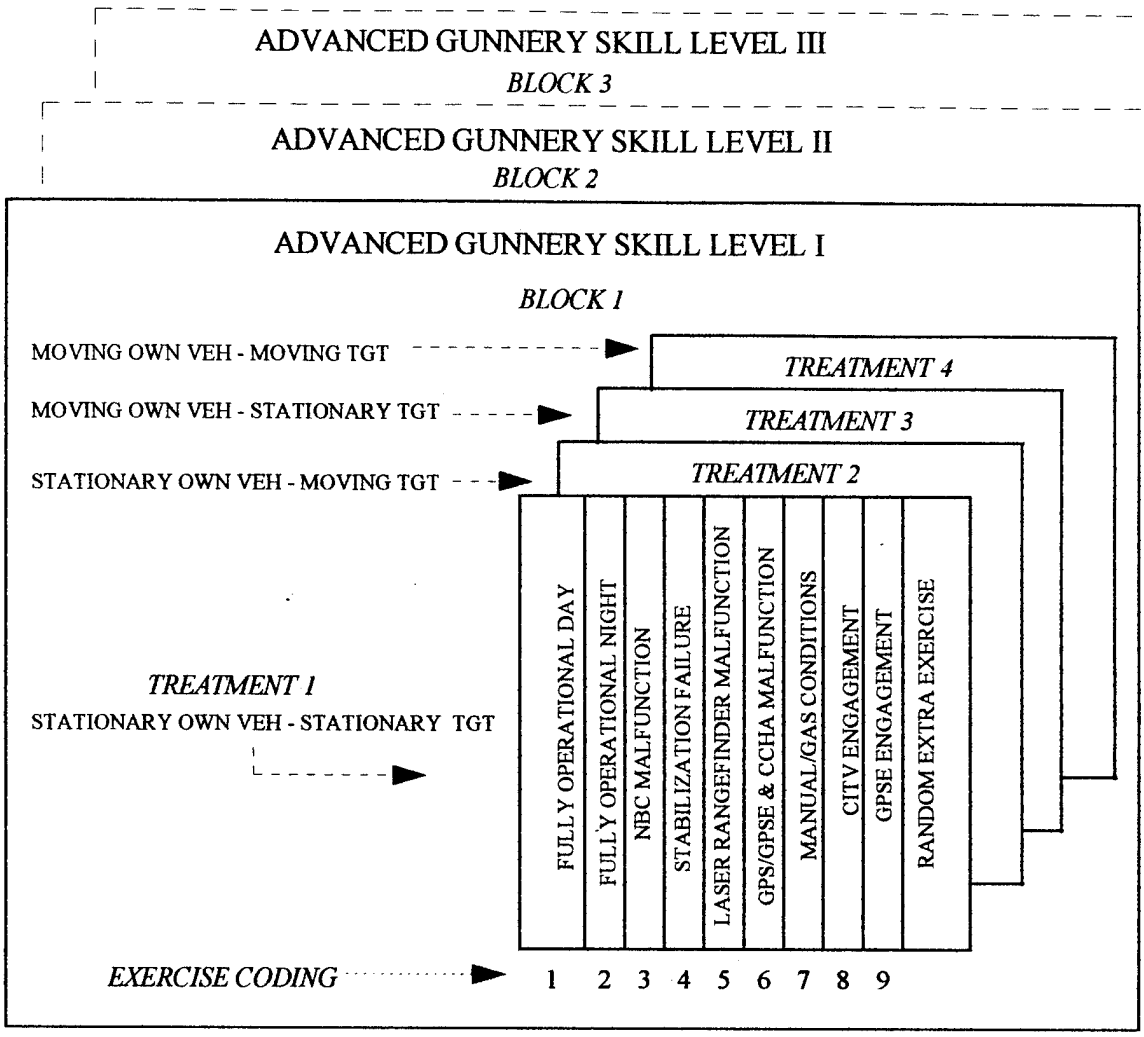


Figure 4. Definition of BCBD Parameters

Because of the parameters and organization of training matrix in this scenario, there are no constraints on block capacity. This feature makes a Balanced Complete Block Design ideal. The second step requires determination of number of replications. Since we have six days available for testing we will assume the participants have mutually agreed to use three days for initial testing and three for retesting, if necessary. Assume we can reasonably shoot twelve exercises in a day. Each replication will contain twelve

exercises: three blocks multiplied by four treatments. There will be three replications available because of consideration of the time constraints, three days, and the parameter definitions.

Step three requires the random assignment of treatments to blocks. Appendix A will be used for this step. To generate the assignment we will need three permutations of four integers for each replication. The three permutations correspond to the number of blocks. The number of integers required corresponds to the number of factor levels. Since we require less than seven integers, we will use the seven digit section of the table. After selecting the table (Lentner and Bishop, 1986) we pick an arbitrary starting point. Index the Table on row number 15, column 5 as indicated in Figure 5. Select the first three columns for replication one, the next three for replication two, and the next three for replication three. Replications are also identified in Figure 5.

The assignment of treatments to blocks is shown in Table 2. The string of numbers generated from the first permutation was 1672345. The factor level assignment is completed by deleting numbers higher than four. The results show that in the first replication, the first block will have treatments 1, 2, 3, and 4, in that given order. The second permutation provided 1635724 as its generated string. This string results in assignment of treatments 1, 3, 2, and 4 for block two. Block three for replication one and the other replications and blocks are shown.

The fourth step of construction of the design requires random assignment of a specific training exercise to a given treatment and block combination. The procedure

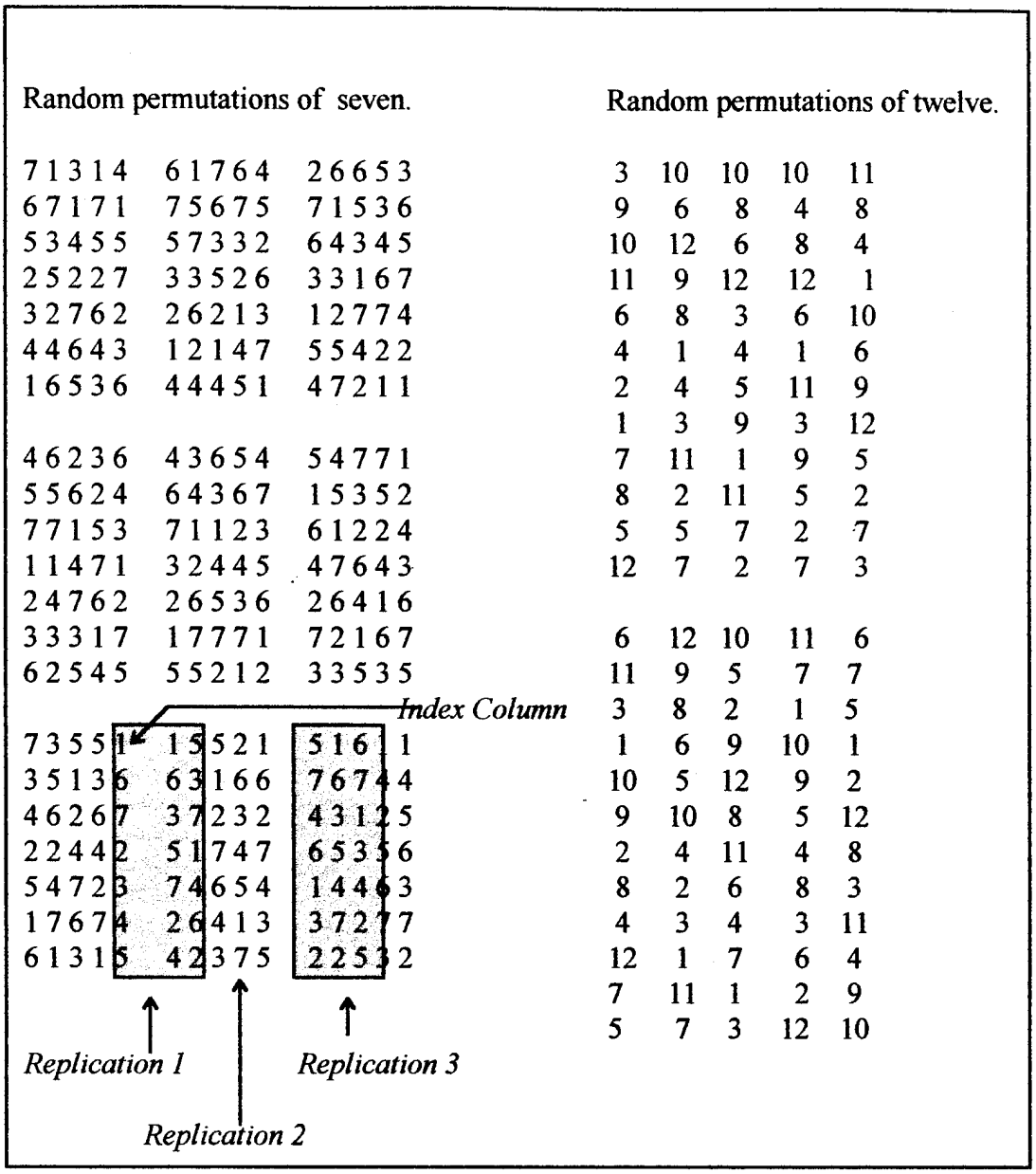


Figure 5. Selection of Random Permutations

used is the same permutation technique as described above. Each factor level will generate one permutation. In this design we will need four permutations. Each factor level appears a total of nine times throughout the design, so we will need nine integers

Table 2

Balanced Complete Block Design Example

Numbers From Table			
		Replication 1	
1672345	→	1234	Block 1
1635724		1324	Block 2
5371462		3142	Block 3
		Replication 2	
5127643	→	1243	Block 1
2634517		2341	Block 2
1627435		1243	Block 3
		Replication 3	
5746132	→	4132	Block 1
1635472		1342	Block 2
6713425		1342	Block 3

in each permutation. This requires the twelve digit portion of Appendix A. Each type of exercise in a factor level is coded numerically from one to nine. The coding and the simulator's training matrix are shown in Figure 4. After obtaining a permutation table (Lentner and Bishop, 1986) the first step is choosing an arbitrary starting point. The Table is indexed on row 36, column 1 as illustrated in Figure 6.

The number string generated for treatment one is 1, 12, 8, 4, 7, 1, 6, 11, 5, 9, 3, and 10. After the number string is generated, integers larger than nine are eliminated. This assignment is demonstrated in Table 3 for treatment 1.

Random permutations of twelve.

3	10	10	10	11
9	6	8	4	8
10	12	6	8	4
11	9	12	12	1
6	8	3	6	10
4	1	4	1	6
2	4	5	11	9
1	3	9	3	12
7	11	1	9	5
8	2	11	5	2
5	5	7	2	7
12	7	2	7	3

6	12	10	11	6
11	9	5	7	7
3	8	2	1	5
1	6	9	10	1
10	5	12	9	2
9	10	8	5	12
2	4	11	4	8
8	2	6	8	3
4	3	4	3	11
12	1	7	6	4
7	11	1	2	9
5	7	3	12	10

Figure 6a. Permutation for Random Assignment of Exercise to Treatment.

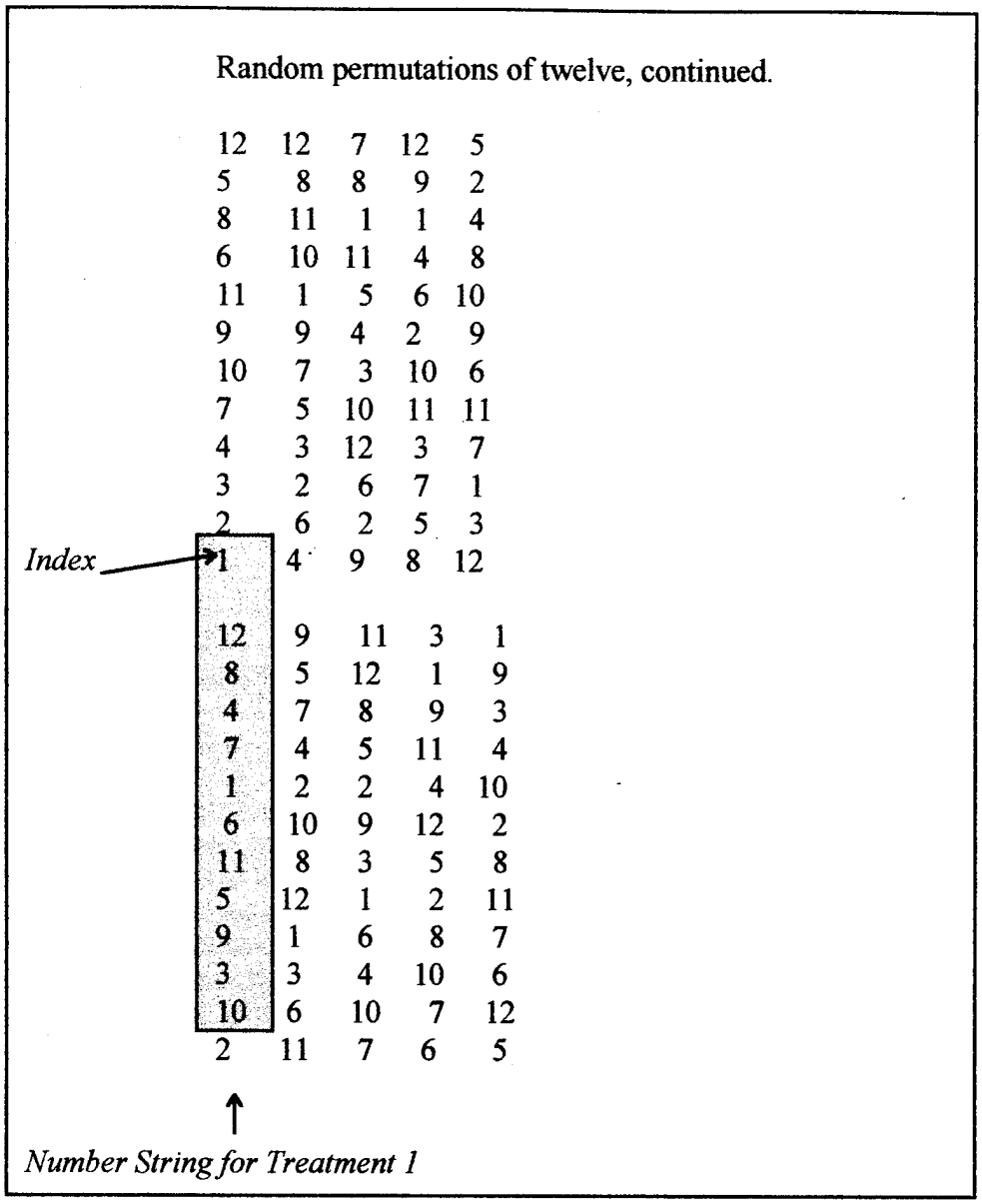


Figure 6b. Permutation for Random Assignment of Exercise to Treatment.

Results from Table 3 indicate that exercise Fully Operational Day will be fired in treatment 1, block 1, replication 1. The CITV exercise will be fired in treatment 1, block 2, replication 1, and so on. The rest of treatment 1 results are as follows. In exercise is Manual Gunner's Conditions. Block 2 of replication 2 is Fully Operational

Table 3

Training Exercise To Factor Level Assignment Example

Exercise Coding	
1 = Fully Operation Day	
2 = Fully Operational Night	
3 = NBC Warfare Malfunction	
4 = Stabilization Failure	
5 = Laser Rangefinder Malfunction	
6 = GPS/GPSE Malfunction	
7 = Manual Gunner's Conditions	
8 = CITV Engagement	
9 = GPSE Engagement	
Numbers From Table	Order of Exercises
1, 12, 8, 4, 7, 1, 6, 11, 5, 9, 3, 10	1, 8, 4, 7, 1, 6, 5, 9, 3

Day exercise. Block 3, replication 2 is GPS/GPSE Malfunction. The three exercises in replication 3, blocks 1,2, and 3 are Laser Rangefinder Malfunction, GPSE Engagement, and NBC Warfare Malfunction. The result of the second step of IEDM is a sample size of 36 exercises for the initial test for the target verification experiment. The next section highlights the differences between Balanced Complete Block Designs, Balanced Incomplete Block Designs, and Partially Balanced Incomplete Block designs. This section has no impact on the target presentation verification; it is an additional example presented for design comparison.

Balanced Incomplete Block Design

Both the Balanced Incomplete Block Design and the Partially Balanced Incomplete Block Design are appropriate when there is a constraint present on the number of blocks or the capacity of each block. The IAGTS training matrix in Appendix E provides a means of comparing the three experimental design methods. The organization of this matrix contains a constrained block capacity. In this case there are six factor level combinations. These correspond to the different reticle aim groups of exercises. There are four levels of systems management that make ideal blocking criteria. The IAGTS matrix has a different construction than the AGTS matrix. Now there are three levels of target acquisition. These levels provide a factor on which to constrain the block capacity. As we begin the construction of a BIBD the input parameters are six treatments, four blocks, and block capacity of three. The relationship of these parameters and the matrix are shown in Figure 7.

Because of the block capacity constraint and the number of treatments, we will need a larger design to achieve good balance. By looking in a table of existing designs (Beyer, 1968) we can find a design that fits our situation. In our scenario we assumed the block capacity was a constraint, and that we were not constrained on the number of blocks or replications. The process of selecting the appropriate design and the design plan are shown in Figure 8.

The plan had to be modified because BIBDs do not exist for all parameter combinations. In this case the plan from the table had 20 blocks (See Appendix B).

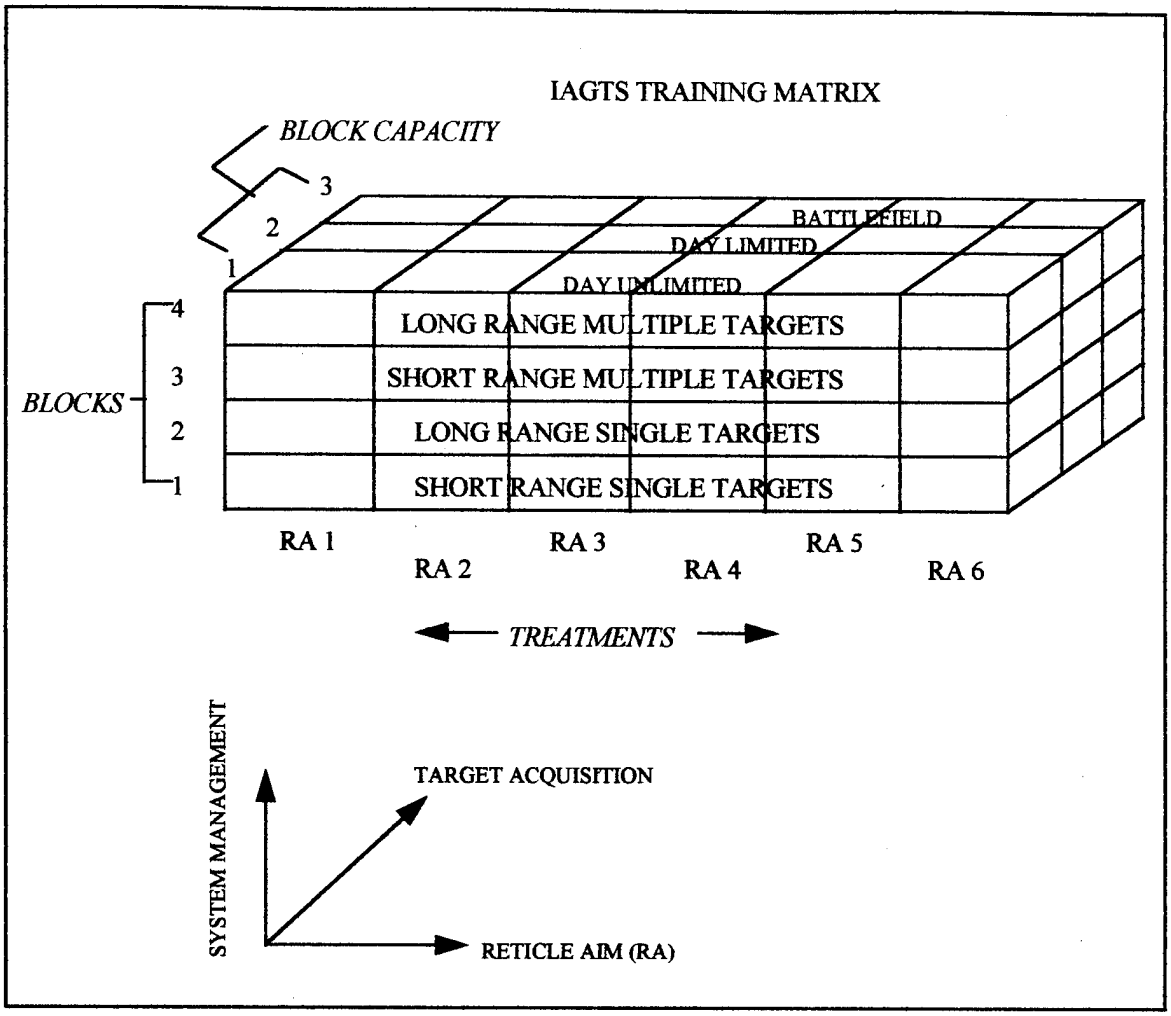


Figure 7. IAGTS Matrix with BIBD and PBIBD Parameters

The plan was modified for four blocks with more replication, by the author. The modified plan shows that in the first replication , the first block contains treatments 1, 2, and 3. The second replication, third block has treatments 1, 2, and 4. In this scenario we will run 60 exercises. Assignment of training exercises to each treatment is

The table that follows is an excerpt from the table provided in Beyers, 1968. Parameters are as follows.

t = number of treatments
k = capacity of blocks
r = number of replications
b = number of blocks

t	k	r	b	Plan Reference Number	
4	2	3	6	11.1	
5	2	4	10	11.2	
	3	6	10	11.1a	
6	2	5	15	11.3	
	3	5	10	11.4	
	3	10	20	11.5	← Select this plan
	4	10	15	11.6	
7	2	6	21	11.2a	
	3	3	7	11.7	
	4	4	7	11.8	
8	2	7	28	11.9	
	4	7	14	11.10	

Plan 11.5

Block	Rep I	Rep II	Rep III	Rep IV	Rep V
(1)	1 2 3	(3) 1 2 4	(1) 1 2 5	(3) 1 2 6	(1) 1 3 4
(2)	4 5 6	(4) 3 5 6	(2) 3 4 6	(4) 3 4 5	(2) 2 5 6

Block	Rep VI	Rep VII	Rep VIII	Rep IX	Rep X
(3)	1 3 5	(1) 1 3 6	(3) 1 4 5	(1) 1 4 6	(3) 1 5 6
(4)	2 4 6	(2) 2 4 5	(4) 2 3 6	(2) 2 3 5	(4) 2 3 4

Figure 8. Balanced Incomplete Block Design Example

conducted in the same manner as in the Balanced Complete Block Design example. In many cases this number of exercises will be too large. When confronted with additional block number constraints, the Partially Balanced Incomplete Block Design is a good tool.

Partially Balanced Incomplete Block Design

For this example we will use the training matrix and scenario developed for the IAGTS in the previous scenario. The IAGTS Matrix is shown in detail in Appendix E. In this example, just as in the previous section, there are six treatments represented by the various reticle aim groups. Blocking is done on the differing levels of difficulties of targets, or systems management. The number of blocks is four. Assume because of budgetary constraints we can only execute one day of experimentation that will include 12 exercises. This scenario might occur late in the development cycle because of a depleting budget. In this case we assume we need a small experiment to verify corrections of previously identified minor errors, and we do not want to spend a large amount on the test. For a relationship of the parameters of number of blocks, block capacity, and treatments to the training matrix refer to Figure 7. The initial step of defining the parameters is complete. The next step is to find an appropriate plan in an existing table (Clatworthy, 1973). Refer to Figure 9 for an illustration of this step. By looking up the parameters of number of treatments equal to six, number of blocks to be

Table Of Partially Balanced Incomplete Block Designs

The following table is an excerpt of one provided in Clatworthy, 1973. Parameters are defined as follows:

v = number of treatments
r = number of replications
k = block capacity
b = number of blocks

Reference Number	v	r	k	b
SR1	4	2	2	4
SR2	4	4	2	8
SR3	4	6	2	12
SR4	4	8	2	16
SR5	4	10	2	20
SR6	6	3	2	9
SR7	6	6	2	18
SR8	6	9	2	27
SR9	8	4	2	16
SR10	8	8	2	32
SR11	10	5	2	25
SR12	10	10	2	50
SR13	12	6	2	36
SR14	14	7	2	49
SR15	16	2	8	64
SR16	18	9	2	81
~~SR 17~~	~~20~~	~~10~~	~~2~~	~~100~~
SR18	6	2	3	4

←——Select this plan.

Figure 9. Selecting Partially Balanced Incomplete Block Design From Table

four, block capacity to be three, and replications to be two, we find a plan with a reference number SR18. Given the plan reference number, the next step is to look up the design description. Clatworthy, 1973 provides the plan described in Table 4.

Table 4

Partially Balanced Incomplete Block Design

Design SR18	
1 2 3	Block 1
1 5 6	Block 2
2 4 6	Block 3
3 4 5	Block 4

The plan shows that block one includes treatments 1, 2, and 3. The next step is random assignment of a training exercise to a treatment and block combination. The procedure will be exactly the same as illustrated in the Balanced Complete Block example. In this case since each treatment appears in the design twice, one permutation of two integers is necessary for each treatment. Because of the two integer requirement, the seven digit table can be used. There will be a total of six permutations (for each treatment) of two integers (number of times the treatment appears in the design). In this case there will be a sample size of 12 exercises.

Figure 10 illustrates the differences in terms of the training matrix representation for Balanced Incomplete Block Design, Partially Balanced , and Balanced Complete Block Designs.

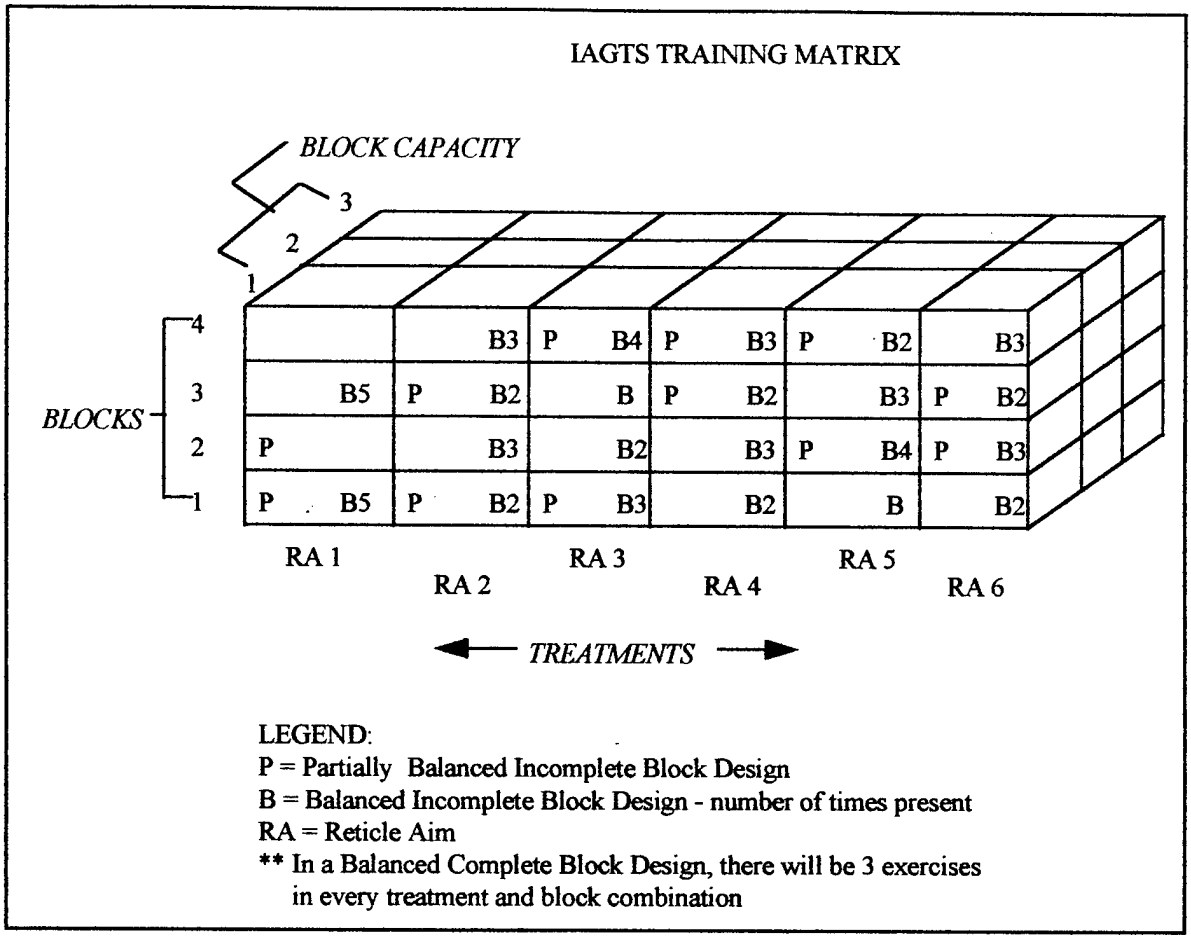


Figure 10. Comparison of Designs

The Balanced Complete Block Design will have three exercises in each factor level and block combination for a total of 72 exercises. Balanced Incomplete Block Design has a total of 60 exercises as shown. Partially Balanced Incomplete has 12 exercises. The figure verifies that the best representation of the training matrix is by Balanced Completed Block Designs, followed by Balanced Incomplete Block Designs, and Partially Balanced Incomplete Block Designs. Partially Balanced and Balanced Incomplete will be less expensive, but provide less information. Always choose the

best representation the budget can allow. Based on one of the three tools presented a sample size has been found. In theory historic data from a smaller design could be recovered and used in future construction of one of the larger designs. This is only possible if the treatment, block combinations, and response are the same and no changes have been made to the system that may have introduced error. The next step in IEDM is determining when an exercise has passed or failed.

Determine Progression Rules

The determination of progression rules will be illustrated by the experiment for checking target presentations that was developed earlier in the AGTS example. The first step in determining the progression rules are defining the appropriate parameters. The sample size for the AGTS system with a Balanced Complete Block Design was provided in the second step of the model, and the sample is 36. The next parameter is the number of samples to take. This was also defined in the second step by deciding to have an initial test, then a retest. The number of samples is two. The next step is defining the AQL. The AQL and error definitions are decided by both user and producer members, and the process is arbitrated by the testing team chief. In this example we will set AQL for catastrophic errors at 0%, major errors at 1%, and minor errors at 2.5%. The next step requires the inspection level. Normal or inspection level II is appropriate. Refer to the except from Military Standard 105D in Appendix D (Duncan, 1986). Look up the cumulative sample size for the experiment. In this case

sample size is 36. Go to the appropriate place in the double sample table for a 36 exercise sample, inspection level II, and appropriate AQL. Results are shown in Table 5.

Table 5

Acceptance and Rejection Numbers Example

AQL 1% (Major Errors)		
Acceptance Number	Rejection Number	
0	2	Sample 1
1	2	Sample 2
AQL 2.5% (Minor errors)		
Acceptance Number	Rejection Number	
1	4	Sample 1
4	5	Sample 2

The results show that if a catastrophic failure occurs, the experiment fails. If no major errors occur and one or less minor errors occur in sample 1, the experiment passes. If either two major failures or four minor failures occur in sample 1, the experiment fails. Rework should be conducted, and a future experiment should be scheduled. If either one major failure occurs or between one and four minor errors occur, make simulator corrections and proceed with sample 2. The number of failures from sample 1 is carried forward to the second sample. If either one or less major failures or four or less minor failures occur in sample 2 the experiment passes. If two or more major failures occur or five or more minor failures occur, the experiment fails.

In general, when an experiment passes it moves to the next higher level of experimentation (i.e., from integration level to system level). When a level fails, the simulator should be repaired and a new experiment should be scheduled at the same level, even if a failure occurs between samples. The exception is when a decision is made to progress with errors. In this scenario AQLs were established for major and minor errors. A manager may use this progression scheme by designating that major errors only have effect on progression. In this case, AQL and acceptance/rejection numbers should be ignored. If AQLs are established for both errors, then both major and minor errors can effect program progression.

The key issues in this step of IEDM are definition of catastrophic, major, and minor errors. Producer and user must agree on the definitions. Total loss of power in the simulator for unknown reasons is an example of catastrophic failure. A major failure would be the simulator is unable to destroy an enemy armored vehicle with its main gun in a specific type of munition. A minor failure example is absence of sound effects when firing the coaxial machine gun. Another issue is selection of AQL. If the acceptance and rejection number are mutually agreed to be too tight by both user and producer, less stringent AQL should be selected. This feature is possible because of the cumulating errors theory described earlier.

The example presented in this chapter highlights the user friendly characteristics of IEDM. Quantifiable verification and validation results are possible without overbearing analysis. The model is flexible and can fit within existing acquisition systems. The

user and producer groups also have the flexibility to tailor the model to their specific needs. Contributions of the model and interim experimentation will be explored in greater detail in the conclusion chapter.

CHAPTER 8

CONCLUSION

The contributions of IEDM are a result of the advantages offered by interim experimentation and by having a structured approach to interim verification and validation. The benefits of interim experimentation begin with early detection of errors. The earlier that errors, both suspected and previously unknown, are found and addressed, the less expensive they become. In addition, discovering them early decreases the effects they might have on other portions of the training simulator. Another benefit of interim experimentation is that it forces the user to get involved in the development of the training simulator. This allows the user to feel ownership of the system earlier and causes him to share responsibility in the level of verification and validation achieved prior to system fielding. Interim experimentation can reduce costs of the verification and validation process. It reduces cost by providing a mechanism for more efficiency in later experiments, by reducing the need for some subsequent testing, and by reducing manpower requirements.

Contributions of having a structured approach to the interim verification and validation process are also numerous. The IEDM is very flexible and can be

implemented into existing development and acquisition systems. If this type of system does not exist in the organization, it provides a methodology for achieving efficient organization in verification and validation efforts. IEDM provides a mechanism for determining the number of experiments to conduct at each level of product development. IEDM reduces experiment sizes by providing tools to take balanced representations of large training matrixes. It provides a progression scheme that is statistically sound but does not have a large appetite for data. Its foundation takes advantage of both the unique controllable environment of training simulators and of the benefits of interim experimentation. This allows experiment progression to be based on pass/fail criteria, and not on detailed hypothesis testing. However, if detailed analysis is required the design structure of the model allows for this analysis. The design approach used in the model also provides robust results. The mechanisms for design of the experiment can cover any possible combination of parameters. IEDM reduces verification and validation total costs by reducing the manpower required for data analysis, reducing the need for subsequent experiments, and by improving the efficiency of subsequent experiments.

The contribution of this research is a mechanism, based on statistically sound methods, for the progressive evaluation of training simulators through their development. There is room for continued development and research of the model. The experimental design field can be searched for other techniques for achieving small representative samples of the complex training matrixes. A generation algorithm for

both Balanced Incomplete Block Designs and Partially Balanced Incomplete Blocks designs should be investigated to provide an alternative to existing designs. In the area of Partially Balanced Incomplete Block Designs, the significance and benefits of different associate class designs should be investigated. Research should also focus on the impact of differing AQLs and the effect of error carry over. This work provides a structured approach to verification and validation of training simulators where none previously existed. All future efforts should focus on the scope of the impact of this model and on refinements in its structure.

Appendix A

Random Permutations

The following lists of permutations are excerpts of those computed by Lentner and Bishop, 1986.

Random permutations of seven.

7 1 3 1 4	6 1 7 6 4	2 6 6 5 3
6 7 1 7 1	7 5 6 7 5	7 1 5 3 6
5 3 4 5 5	5 7 3 3 2	6 4 3 4 5
2 5 2 2 7	3 3 5 2 6	3 3 1 6 7
3 2 7 6 2	2 6 2 1 3	1 2 7 7 4
4 4 6 4 3	1 2 1 4 7	5 5 4 2 2
1 6 5 3 6	4 4 4 5 1	4 7 2 1 1
4 6 2 3 6	4 3 6 5 4	5 4 7 7 1
5 5 6 2 4	6 4 3 6 7	1 5 3 5 2
7 7 1 5 3	7 1 1 2 3	6 1 2 2 4
1 1 4 7 1	3 2 4 4 5	4 7 6 4 3
2 4 7 6 2	2 6 5 3 6	2 6 4 1 6
3 3 3 1 7	1 7 7 7 1	7 2 1 6 7
6 2 5 4 5	5 5 2 1 2	3 3 5 3 5
7 3 5 5 1	1 5 5 2 1	5 1 6 1 1
3 5 1 3 6	6 3 1 6 6	7 6 7 4 4
4 6 2 6 7	3 7 2 3 2	4 3 1 2 5
2 2 4 4 2	5 1 7 4 7	6 5 3 5 6
5 4 7 2 3	7 4 6 5 4	1 4 4 6 3
1 7 6 7 4	2 6 4 1 3	3 7 2 7 7
6 1 3 1 5	4 2 3 7 5	2 2 5 3 2

Random permutations of twelve.

3	10	10	10	11
9	6	8	4	8
10	12	6	8	4
11	9	12	12	1
6	8	3	6	10
4	1	4	1	6
2	4	5	11	9
1	3	9	3	12
7	11	1	9	5
8	2	11	5	2
5	5	7	2	7
12	7	2	7	3
6	12	10	11	6
11	9	5	7	7
3	8	2	1	5
1	6	9	10	1
10	5	12	9	2
9	10	8	5	12
2	4	11	4	8
8	2	6	8	3
4	3	4	3	11
12	1	7	6	4
7	11	1	2	9
5	7	3	12	10

Random permutations of twelve, continued.

12	12	7	12	5
5	8	8	9	2
8	11	1	1	4
6	10	11	4	8
11	1	5	6	10
9	9	4	2	9
10	7	3	10	6
7	5	10	11	11
4	3	12	3	7
3	2	6	7	1
2	6	2	5	3
1	4	9	8	12

12	9	11	3	1
8	5	12	1	9
4	7	8	9	3
7	4	5	11	4
1	2	2	4	10
6	10	9	12	2
11	8	3	5	8
5	12	1	2	11
9	1	6	8	7
3	3	4	10	6
10	6	10	7	12
2	11	7	6	5

Appendix B

Index To Balanced Incomplete Block Plans

The table that follows is an excerpt from the table provided in Beyers, 1968. Parameters are as follows.

t = number of treatments
k = capacity of blocks
r = number of replications
b = number of blocks

t	k	r	b	Plan Reference Number
4	2	3	6	11..1
5	2	4	10	11.2
	3	6	10	11.1a
6	2	5	15	11.3
	3	5	10	11.4
	3	10	20	11.5
	4	10	15	11.6
7	2	6	21	11.2A
	3	3	7	11.7
	4	4	7	11.8
8	2	7	28	11.9
	4	7	14	11.10
9	2	8	36	11.3a
	4	8	18	11.11
	5	10	18	11.12
	6	8	12	11.13
10	2	9	45	11.14
	3	9	30	11.15
	4	6	15	11.16
	5	9	18	11.17
	6	9	15	11.18
11	2	10	55	11.4a
	5	5	11	11.19
	6	6	11	11.20

The reference number is then used to look up a designated plan. For example:

Plan 11.1	Block	Rep I		Block	Rep II		Block	Rep III	
	(1)	1	2	(3)	1	2	(5)	1	4
	(2)	3	4	(4)	2	4	(6)	2	3

Appendix C

Table Of Partially Balanced Incomplete Block Designs

The following table is an excerpt of one provided in Clatworthy, 1973. Parameters are defined as follows:

v = number of treatments
r = number of replications
k = block capacity
b = number of blocks

Reference Number	v	r	k	b
SR1	4	2	2	4
SR2	4	4	2	8
SR3	4	6	2	12
SR4	4	8	2	16
SR5	4	10	2	20
SR6	6	3	2	9
SR7	6	6	2	18
SR8	6	9	2	27
SR9	8	4	2	16
SR10	8	8	2	32
SR11	10	5	2	25
SR12	10	10	2	50
SR13	12	6	2	36
SR14	14	7	2	49
SR15	16	2	8	64
SR16	18	9	2	81
SR 17	20	10	2	100
SR18	6	2	3	4
SR19	6	4	3	8
SR20	6	6	3	12

An example of the plan description of Plan SR1 is as follows:

1	2
3	4
4	1
2	3

Appendix D

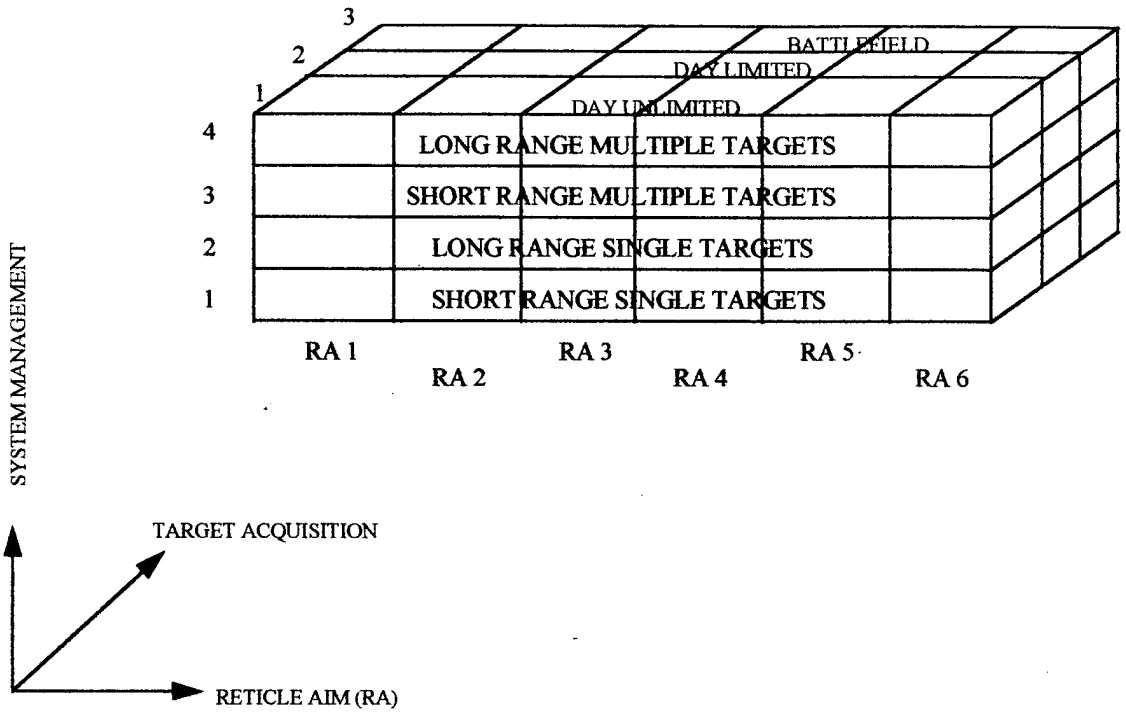
Normal Inspection Double Sample Table

Accceptable Quality Level

	Cumulative Sample Size	1.0 Ac Re	1.5 Ac Re	2.5 Ac Re	4.0 Ac Re
First	32	0 2	0 3	1 4	2 5
Second	64	1 2	3 4	4 5	6 7
First	50	0 3	1 4	2 5	3 7
Second	100	3 4	4 5	6 7	8 9
First	80	1 4	2 5	3 7	5 9
Second	160	4 5	6 7	8 9	12 13
First	125	2 5	3 7	5 9	7 11
Second	250	6 7	8 9	12 13	18 19

Appendix E

IAGTS Crew Training Matrix


3
2
1
BATTLEFIELD
DAY LIMITED
DAY UNLIMITED
4
3
2
1
LONG RANGE MULTIPLE TARGETS
SHORT RANGE MULTIPLE TARGETS
LONG RANGE SINGLE TARGETS
SHORT RANGE SINGLE TARGETS
RA 1
RA 2
RA 3
RA 4
RA 5
RA 6
SYSTEM MANAGEMENT
TARGET ACQUISITION
RETICLE AIM (RA)

LIST OF REFERENCES

Banks, J., (1989). Testing, Understanding, and Validating Complex Simulation Models. In E.A. MacNair, K.J. Musselman, and P. Heidelberger (Eds.), Proceedings Of The 1989 Winter Simulation Conference (pp. 549-551).

Beyer, W., (Ed.). (1968). CRC Handbook Of Tables For Probability And Statistics (2nd ed.). Cleveland, OH: The Chemical Rubber Company.

Bishop, T., & Lentner, M., (1986). Experimental Design and Analysis . Blacksburg, VA: Valley Book Company.

Blanchard, B., (1992). Logistics Engineering and Management (4th ed.). Englewood Cliffs, NJ: Prentice Hall.

Blanchard, B., (1991). Systems Engineering Management. New York: John Wiley & Sons.

Clatworthy, W., (1973). Tables of Two-Associate-Class Partially Balanced Designs. Washington, DC: U.S. Department of Commerce.

Duncan, A.J., (1986). Quality Control and Industrial Statistics (5th ed.). Homewood, IL: Richard D. Irwin, Inc.

Dunham, J.R., (1989, May). V&V In The Next Decade. IEEE Software, 47-53.

Dziuban, S.T., Curry, T.F., Knepell, P.L., & Riley, W.J., (1992). The Synergism Of Simulation And Experimental Design For Training, Optimization, And Validation Purposes. In Proceedings Of The 1992 Summer Simulation Conference (pp. 1002-1006).

Gledhill, D.W., (1994). The Use of Case Tools as an Aid to the Verification of Complex Software Systems. Simulation, 63:5, 329-336.

Headquarters, Department of the Army, (1992). Army Model and Simulation Management Program (Army Regulation 5-11). Washington, DC.

Headquarters, Department of the Army, (1993). Verification, Validation, and Accreditation of Army Models and Simulations (Department of the Army Pamphlet 5-11). Washington, DC.

Ishikawa, K., (1990). Introduction To Quality Control. Tokyo, Japan: 3A Corporation.

Kempthorne, O., (1952). The Design And Analysis Of Experiments. New York: John Wiley & Sons.

LaRocca, G.A., (1965). System Testing. In R.E. Machol, W.P. Tanner, Jr., & S.N. Alexander (Eds.), System Engineering Handbook (pp. 34-1 - 34-11). New York: McGraw-Hill Book Company.

Law, A. M., & Kelton, W. D., (1991). Simulation Modeling And Analysis (2nd ed.). New York: McGraw-Hill, Inc.

Lockheed Martin Corporation Information Systems Division, (1995). System Specification For The Advanced Gunnery Training System (AGTS) (Document Number 12C-I-0012). Orlando, FL.

Montgomery, D.C., (1984). Design and Analysis of Experiments (2nd ed.). New York: John Wiley & Sons.

Meyers, R., & Montgomery, D.C., (1995). Response Surface Methodology: Process and Product Optimization Using Designed Experiments. New York: John Wiley & Sons, Inc.

Musa, J.D., & Ackerman, A.F., (1989, May). Quantifying Software Validation: When to Stop Testing? IEEE Software, 19-27.

Pace, D.K., (1995). A Modest V&V Proposal. Phalanx (23:4), 16-17.

Pellegrino, J., (1995). Test and Evaluation Master Plan For The Advanced Gunnery Training Systems. Orlando, FL: Simulation, Training, and Instrumentation Command, U.S. Army.

Piplani. L.K., Mercer J.G., & Roop R.O., (1994). Systems Acquisition Manager's Guide For The Use Of Models And Simulations. Fort Belvoir, VA: Defense Systems Management College.

Schilling, E.G., & Sommers, D.J., (1988). Acceptance Sampling. In J.M. Juran & F.M. Gyrna (Eds.), Juran's Quality Control Handbook (pp.25.1-25.99). New York: McGraw-Hill Inc.

Wallace, D.R., & Fuji, R.U., (1989, May). Software Verification and Validation: An Overview. IEEE Software, 10-17.

Whitner, R.B., & Balci, O., (1989). Guidelines For Selecting And Using Simulation Model Verification Techniques. In E.A. MacNair, K.J. Musselman, and P. Heidelberger (Eds.), Proceedings Of The 1989 Winter Simulation Conference (pp. 549-551).