1996

# Determining the Factors Influential in the Validation of Computer-based Problem Solving Systems

Leslie Anne Morehead
*Portland State University*

Recommended Citation
Morehead, Leslie Anne, "Determining the Factors Influential in the Validation of Computer-based Problem Solving Systems" (1996). *Dissertations and Theses.* Paper 1245.
https://doi.org/10.15760/etd.1244

# DETERMINING THE FACTORS INFLUENTIAL IN THE VALIDATION OF COMPUTER-BASED PROBLEM SOLVING SYSTEMS

by

**LESLIE ANNE MOREHEAD**

A dissertation submitted in partial fulfillment of the requirements for the degree of

**DOCTOR of PHILOSOPHY**
**in**
**SYSTEMS SCIENCE: BUSINESS ADMINISTRATION**

Portland State University
1996

DISSERTATION APPROVAL

The abstract and dissertation of Leslie Anne Morehead for the Doctor of Philosophy in Systems Science: Business Administration were presented May 6, 1996 and accepted by the dissertation committee and the doctoral program.

COMMITTEE APPROVALS: [redacted]

George G. Lendaris, Chair

[redacted]

Kurt Fedra

[redacted]

Kish Sharma

[redacted]

Wayne W. Wakeland

[redacted]

Henry D. Crockett
Representative of the Office of Graduate Studies

DOCTORAL PROGRAM APPROVAL:

[redacted]

Beatrice T. Oshika, Director
Systems Science Ph.D. Program

* * * * * * * * * * * * * * * * * * * * * * * * * * * *

ACCEPTED FOR PORTLAND STATE UNIVERSITY BY THE LIBRARY

by [redacted] on 17 June 1996

# ABSTRACT

An abstract of the dissertation of Leslie Anne Morehead for the Doctor of Philosophy in Systems Science: Business Administration presented May 6, 1996.
TITLE: Determining the Factors Influential in the Validation of Computer-Based Problem Solving Systems.

Examination of the literature on methodologies for verifying and validating complex computer-based Problem Solving Systems led to a general hypothesis that there exist measurable features of systems that are correlated with the best testing methods for those systems. Three features (Technical Complexity, Human Involvement, and Observability) were selected as the basis of the current study. A survey of systems currently operating in over a dozen countries explored relationships between these system features, test methods, and the degree to which systems were considered valid.

Analysis of the data revealed that certain system features and certain test methods are indeed related to reported levels of confidence in a wide variety of systems. A set of hypotheses was developed, focused in such a way that they correspond to linear equations that can be estimated and tested for significance using statistical regression analysis. Of 24 tested hypotheses, 17 were accepted, resulting in 49 significant models predicting validation and verification percentages, using 37 significant variables. These models explain between 28% and 86% of total variation. Interpretation of these models (equations) leads directly to useful recommendations regarding system features and types of validation methods that are most directly associated with the verification and validation of complex computer systems. The key result of the study is the identification of a set of sixteen system features and test methods that are multiply correlated with reported levels of verification and validation. Representative examples are:

- People are more likely to trust a system if it models a real-world event that occurs frequently.
- A system is more likely to be accepted if users were involved in its design.
- Users prefer systems that give them a large choice of output.
- The longer the code, or the greater the number of modules, or the more programmers involved on the project, the less likely people are to believe a system is error-free and reliable.

From these results recommendations are developed that bear strongly on proper resource allocation for testing computer-based Problem Solving Systems. Furthermore, they provide useful guidelines on what should reasonably be expected from the validation process.

# Contents

## Tables & Figures

# ACKNOWLEDGEMENTS

First I must thank my committee and colleagues in Systems Science: Thanks to Kurt Fedra of the International Institute for Applied Systems Analysis (IIASA) in Laxenburg, Austria, for countless discussions over several summers that helped shape this dissertation, and for teaching me philosophy. To Eduard Löser at IIASA for years of research support. To Harold Linstone who first mentioned to me the importance of IIASA – that unique place where scientists from East and West exchanged ideas (and became friends) during the Cold War. To Kish Sharma, a most civilized man of many cultures; his dignity has taught me much about the world. To George Lendaris, who taught me how to question and to go directly for the answers. To Wayne Wakeland, for his enthusiasm and for helping me collect great data. To Henry Crockett, a fine example of good business sense and exciting computer theory melded together. To Beatrice Oshika; she simply told me her story when I said I was thinking about giving up. To Dawn Kuehnle, whose strength, optimism, and organization are unequaled.

Also to my friends. Thanks to Gladys Underwood, for encouragement from day one. To Faye Palmerton, who still thinks getting a Ph.D. is easy and that I should have done it years ago. To Gail Martwick and Nadja Tchebakova, who always were willing to lend a hand no matter what. To my great friends Judy and Jim Wick: You make it possible for me to be in my Portland, where I am most alive and where I can work and learn and love without reservation. To my godchildren, Kristina and Fraser Wick, truly *my* children and the reason we all continue on. They think that now that I am a "doctor" surely I can cure computer viruses.

And thanks to my husband, Robert Monserud, for his trust, his patience, his wacky humor. Those who have been through this process know that the

spouse also “earns” the Ph.D. Bob didn’t need another Ph.D. But he was my editor, my critic, and my sounding board for what seemed like a very long time; now he, too, knows a great deal about the validation of complex Problem Solving Systems.

# OVERVIEW

This dissertation begins with a general statement of the problem and a short discussion on the philosophy of knowledge in Chapter 1. The state of knowledge on Verification and Validation is examined in Chapter 2. Three measurable characteristics of complex Problem Solving Systems are presented in Chapter 3. Methods for testing hypotheses derived from the ideas in Chapters 2 and 3 are then discussed, utilizing data collected on a variety of systems (Chapter 4). Results of regression analysis are reported and interpreted in Chapter 5. The Discussion (Chapter 6) makes suggestions for using the results of this study in business or research software development environments.

The first 3 chapters focus on background, philosophy and definitions. The reader interested exclusively in the study's analysis and results is invited to begin at Chapter 4. Also read the first few pages of Chapters 1 and 2 for a brief outline of the verification and validation issues that this study addresses.

# CHAPTER 1: Introduction

Computer programmers don't usually think of themselves as engaged in the Search for Truth, but they are. Software designers always wish to demonstrate that their products work perfectly; certainly clients and users would like to be convinced. In the last 40 years of software development, much effort has gone toward finding out how to prove that software is correct, or "true". These efforts are widely known as **Verification and Validation ("V&V").**

That the computer executes its model correctly (verification) is extremely important. But how the model performs in the real world (validation) is the overriding consideration. The computer code can be likened to a theorem. Verification is necessary but not sufficient to demonstrate that the model is correct. Validity—if it can be demonstrated—is both a necessary and sufficient condition for proving that the model accurately represents some relevant aspect of reality.

As computer technology continues to influence more aspects of our lives, the necessity of developing reliable computer systems is of fundamental importance. Research efforts on Verification and Validation are hampered by several factors. Misunderstanding arising from incorrect and inconsistent terminology is a nagging problem. Some researchers switch the definitions and use validation for verification and *vice versa* (c.f. Vick and Lindenmayer, 1988 and Castore, 1987). Because demonstrating complete reliability of complex software is virtually impossible, unrealistic expectations are self-defeating and undermine credibility. It is desirable to find more efficient ways to build reliable systems. It is equally important to find ways to increase the likelihood that those systems will be judged to be valid.

The goal of this study is to define a robust framework of validation methods for computer-based problem solving systems and to identify specific methods in that framework that are most likely to produce systems that users trust. Useful verification tools and techniques have been produced; similar tools for validation are still in development. In this study, characteristics of Problem Solving Systems will be analyzed to determine the extent that it is meaningful to use those tools to attempt validation.

This research was designed to study the relationship between systems characteristics and testing methods, and validation. It will be of value to know whether certain system characteristics or test methods can be related to high reported levels of validation. A very large number of individual validation methods used on individual systems is reported in the literature, but very little is written about features or test methods that have been used on groups of systems that are regarded to be valid to a high degree. The objective of this study is to make general statements about how and to what extent Problem Solving Systems can be validated, and whether or not there is a theoretical or practical limit to validation.

Specifically, the following questions will be addressed:

- What is *meant* when we say such systems are valid? (Chapter 2)
- How can these systems be *measured*? (Chapter 3)
- To what extent or degree can we say a system is valid? (Chapter 5)
- What methods lead to relatively higher levels of *reported* validation? (Chapter 5)

To this end, the following general hypothesis will be tested:

**Hypothesis:** The extent to which a Problem Solving System can be said to be valid can be determined from the following features of the system:

Technical Complexity, Human Involvement, and Observability

and from the following ways of testing systems:

Technical, Semi-Technical, and Human Judgment Validation Methods.

## 1.1 Definitions

Herbert Simon (1982) observes that **"decision making"** includes "the whole range of problem solving, thinking, and choosing activities ... involved in productive work." Computers are fast becoming the most important tool used by humans to aid in thinking and making choices. In the proposed research, any computer program specifically designed and used for decision making within a specified domain will be called **problem solving software.** This definition is intentionally broad enough to include most computer programs, both algorithmic and non-algorithmic, that are specifically designed to generate output used by humans in the process of making decisions. Algorithms are step-by-step procedures that 'mechanically' produce a solution to any problem out of a certain class of problems (Groner, 1983). Non-algorithmic software include Knowledge-Based Systems (KBS), Expert Systems (ES), and Decision Support Systems (DSS). Output can be printed reports, video displays, audio signals or any other interaction that transmits information to the human user. A **domain** is a problem or topic area with commonly agreed-upon conceptual boundaries.

The problem domain can also be called the problem environment. Problem solving software is written to assist people in making decisions on specific topics, subject to existing conditions and constraints. The problem, the software and the users of the software exist together as a system. Thus, problem solving software, decision makers, and the environment in which the problem of interest exists are said to form a **Problem Solving System (PSS).**[1]

[1]Churchman (1971) refers to "problem solving machines"; his meaning is not as entirely technical as this sounds. The PSS definition above expands Churchman's concept with even greater emphasis on environmental and human aspects. Linstone's (1984) Multiple Perspectives approach to decision making also proposes to balance Technical, Organizational and Personal

**Verification** of computer code is the process of determining its internal correctness. Compilers typically perform the initial steps of this task by making a judgment on whether the programmer's code adheres to all of the rules of the chosen programming language. Passing this test means that the written language can be translated into executable code. Traditional compilers have been able to find only syntax errors; much work is currently under way on more sophisticated techniques designed for object- oriented code that can determine some aspects of semantic correctness as well. However, the compiler has no way of knowing whether the internal logic of the computer program is the correct one for solving a specific problem. The definition of verification of software thus often also includes the notion of how well an implemented system performs according to its initial specifications. Although widely accepted regarding conventional software, this extension of the definition does not apply to non-algorithmic code (Preece, 1990).

**Validation** of computer code generally refers to the process of determining if the program is reliable *externally*, i.e., if it accurately represents some relevant aspect of reality in the world outside the computer (Preece, 1990). A more general statement about the validation of many different types of models is given by Ziegler (1976): "The validity of the model ... is, how well the model represents the real system." A program is said to be valid if it consistently generates results that users find correct and useful. Validation methodologies range from comparing the results of running the code with initial system specifications, to testing by third parties with independent data sets, to asking an expert's opinion. Virtually all validation processes involve a significant amount of human judgment.

It is usually assumed that there is a close relationship between the process

views equally.

of verification and the process of validation, particularly in the software industry. It is generally agreed that verification comes first, followed by validation. Some writers state that verification must be accomplished in full before meaningful validation can be undertaken; others (Plant & Preece, 1996 and Ho Kang *et al.*, 1996) maintain that the two processes can overlap. Validation as a methodology is rarely referred to separately from verification in the literature, and there is a frequent confusion of the two terms. The emphasis of the present research is on the theory and methodology of computer software **validation**, but it is clear from the current literature that validation cannot be treated separately from verification. Thus, this study reviews software Verification and Validation research to date, and discusses what realistically can be expected from existing "V&V" efforts applied to a group of relatively new, non-algorithmic software products such as Knowledge-Based Systems and Expert Systems. In tracing the development of "V&V" from its original methodologies and goals to the current attempts to verify and validate state-of-the-art software, this study emphasizes not only the difference between the two processes but also how the usual understanding of the nature of validation of computer software has been in error and should be changed.

## 1.2 Philosophical Background

All humans have an intuitive notion about whether or not something is true. The concept of models as representations of reality raises the question of the how to determine the truth (i.e., accuracy) of those representations. There is a rich literature on epistemology and truth in philosophy, but there is a dearth of discussion on this subject in the computing literature.

Gale (1979) distinguishes two major theories about truth:

- The **correspondence theory**: an intuitive judgment that "a statement is true if and only if it corresponds to what it refers to. For example ... 'Today is Wednesday.'" This works well for observable, or factual, knowledge and ideas.

- The **coherence theory**: a judgment that a statement such as "$2 + 2 = 4$" is true simply because it coheres, that is, it is logically consistent with the entirety of the conceptual system of arithmetic. Coherence theory is very useful in domains that are defined by a set of widely accepted theories, principles and laws (e.g., mathematics).

This distinction is noteworthy because these are the common reasoning methods that people use to determine if something is true. These theories form the basis of virtually all "V&V" testing methods. Most verification methodology is analogous to coherence theory, while most validation methodology is analogous to correspondence theory.

Neither of these theories goes far enough in explaining on what basis humans accept some ideas and reject others. It would clearly be naive to hold that human thought patterns are restricted to only observable phenomena or logically consistent paradigms. More importantly, it must be recognized that "correspondence" and "coherence" as defined above lose their meaning as reality—and thus the "truth" about that reality—changes. There will always be some people willing to argue that fundamental, absolute truth exists. But modern philosophers are persuasive that even the most basic "truth" can and does change. The most general way to describe this thinking is as:

- The **social theory** of truth: an acceptance that something is true based on influence by culture, peer groups, schooling, historical developments or other changes in society

Other theories of truth are the "evidence" theory (what appears to be true based on evidence) and the "instrumentalist" theory (what is useful, or pragmatic) Since they are clearly conditional—not absolute—theories, these can be thought of as subcategories of the social theory of truth. They will be found to be of value in considering realistic approaches to validation.

### 1.2.1 Wittgenstein and Popper

An outgrowth of the seminal work on logical constructionism by Whitehead and Russell (1910-1913), the Logical Positivists (Vienna Circle of philosophers, 1920's to 1930's) were interested in making the distinction between meaningful ("scientific") and non-meaningful statements. For a statement to contain meaning it must be empirically verifiable. That is, it would have to be shown, at least in principle, to be definitively true or false. Ludwig Wittgenstein formulated the Verifiability Principle: the idea that the meaning of a statement is identical to the method of verifying it. In his seminal work, *Tractatus Logico-philosophicus* (1922) Wittgenstein used truth tables and other logical analysis techniques to show that some sentences can be said to be true or false. He maintained that "a logical picture of facts is a thought," and "a thought is a meaningful sentence."[2] His method of proof was to assign a truth value to each entity in a sentence, map the verbs of the sentence onto legal operators, and allow his logical analysis mechanism to reach a conclusion.

Wittgenstein was originally interested in proving that a sentence was equal to something real, and that in the process of constructing sentences, human beings continuously construct models of reality. To find out if a sentence were true, one needed to only know the primary entities (words), their truth values, and how to perform the logic. If the sentence were true,

[2]English translations of #3 and #4 of Wittgenstein's seven primary theses; Edition Suhrkamp 12, 1980.

then it was a clear representation of reality. Later, in *Philosophical Investigations* (published posthumously, 1953), Wittgenstein shifted to the opinion that sentences have meaning only within special circumstances that must be understood along with the sentence. It is not enough to know the elemental parts of a sentence to determine its truth; we can judge the truth of a sentence only if we understand the knowledge contained in the sentence first.

Although the Verifiability Principle is out of favor in modern philosophical thought, the idea of "proving something true" is still very much alive in testing methodologies of all kinds. The real legacy of the Verifiability Principle is the recognition (and the frustration) that not all sentences can be verified. This realization struck both the mathematical and philosophical communities like a bomb in 1931 with the proof of Gödel's Theorem (van Heijenoort, 1972). It held that the consistency of a formal system adequate for number theory cannot be proven within the system. Thus, there exist some statements that cannot be proven to be either true or false. The firm belief of all mathematicians (save Gödel) and many philosophers that all statements could be proved or disproved was dashed.

Wittgenstein and the Logical Positivists did not hold that there are meaning*less* statements. They merely meant to identify the statements about which humans can have meaningful discussions. With the same goal in mind—but rejecting the notion that *any* statement can conclusively be shown to be true—Karl Popper developed the famous Falsification Criterion. In *Conjectures and Refutations* (1963), Popper stressed that not only do humans learn from their mistakes, but that correcting mistakes is the *only* way we learn. Continuous testing frequently results in feedback about errors, omissions, incorrect assumptions, and the like. Receiving a negative result in an experiment, for example, indicates either that the experimental design is faulty and should be abandoned, or that it possibly could be fixed and the experiment run again. Failing to receive negative results after repeated testing is an indicator that the hypothesis (or model) may in fact be correct. The falsification criterion is the key to the Scientific Method. Scientists today generally agree that, as increasingly more sophisticated and novel methods are used unsuccessfully to invalidate (falsify) a hypothesis, the burden of evidence for accepting the hypothesis as true mounts.

To progress toward more useful, possibly correct solutions to a problem, Popper emphasizes refining the hypothesis and retesting, rather than analyzing the knowledge that went into designing the potential solution.

> "In general we do not test the validity of an assertion or information by tracing its sources or its origin, but we test it, much more directly, by a critical examination of what has been asserted—of the asserted facts themselves." (Popper, 1978, p.25)

An individual scientist is not alone in the responsibility of carrying out this critical examination of new hypotheses. The scientific community shares the burden by performing independent testing. Popper, and others, note that criticism of existing theories has always been the main way in which knowledge is advanced.

The influence of Falsification Theory is widely felt in many disciplines, particularly in the hard sciences such as physics. The theory also has been applied in many other areas, including social science, business and political science. And it is of increasing interest as a validation methodology in many disciplines that make significant use of complex computer models.

Falsification theory encourages extremely creative model testing. When considered as a verification or validation methodology, however, these two problems become apparent:

1. It is not the case that every model should be thrown out—or even reworked—after one apparently negative result. Some programs, for example, are designed primarily to make the human user think about alternative strategies, some of which are not at all desirable.

2. Popper claims that it is not necessary to consider the origin of knowledge in order to falsify statements (or models). While this may be true, it *is* necessary to know and understand the nature and meaning of the source knowledge in a model in order to correct it, or use it more appropriately. Attempting to "fix" program code without understanding the meaning of the knowledge used to build it would be folly.

Although neither Popper nor Wittgenstein (and the Logical Positivists) solved the problem of how to determine truth, they contributed significantly to our understanding of the nature of the problem of determining "truth". Wittgenstein understood that only a very limited number of sentences could be analyzed using truth tables. Popper said that *no* statement could be positively shown to be true. And no philosopher since has convincingly argued that absolute truth can be proved.

Philosophers are interested in finding out what is real (metaphysical questioning) and in figuring out the nature of knowledge (epistemology). Ordinary humans, however, are much more interested in getting on with their lives. Rarely in the course of a day are human beings aware of determining if things are "true." But that they do this continuously and act on the results cannot be denied.

Interpretation by humans is essential in attempting to validate a model. Human interpretation is a major factor in judging truth, as well. All people do not have identical notions of truth; our backgrounds, personalities and frames of reference differ. Alasdair MacIntyre's article in *Paradigms & Revolutions*

(Gutting, 1980) points out that what the person accepts as true depends on the person's "schemata" in which reality is perceived. In order to survive an "epistemological crisis" the individual must

> "...come to understand how the criteria of truth and understanding must be reformulated. Because in such crises the criteria of truth, intelligibility, and rationality may always themselves be put in question ... we are never in a position to claim that now we possess the truth. The most we can claim is that this is the best account which anyone has been able to give so far." (p.56-57)

### 1.2.2 Churchman and The Systems Approach

C. West Churchman (1971, 1979), one of the founding fathers of modern Systems Science Theory, used most of philosophy from the seventeenth century on as a tool in searching for an ideal problem solving system, i.e., one that is capable of independent inquiry. In particular, he used the history of epistemology to show how learning can be designed and how the design can be justified. Churchman translates philosophical systems into a language for the design of inquiring systems. Fundamentally, a desired feature of inquiry has for decades been a direct method for certifying the truth or falsity of simple, clear statements and proposals. Recognizing that different philosophers over the years have had quite different opinions about how to determine truth, Churchman compiles the contributions of several major philosophers, and argues that each of these historically presented a framework within which we can view whole systems. The following brief summary of the ideas of Churchman's favorites illustrate what he was up against in such a compilation:

- Leibnitz: Reality is a set of facts and relationships; the goal is to find the underlying absolute truth that links everything together.

• Locke: Truth is what the community agrees upon; new members of the community learn and accept definitions from older members.

• Kant: There exist certain given *("a priori")* true principles that are universal and necessary for understanding existence. Human beings arrive at contingent truths through sensory experience.

• Hegel: Truth is found by using the dialectic method (thesis, antithesis, synthesis) of emphasizing and elaborating contradictions until certain common features remain.

• Singer: Progress toward truth is a pragmatic process of wide-ranging (not reductionist or simply logical) inquiry, useful at many stages, but apparently without end.

This summary is a good place to leave the discussion of philosophy in a paper that is addressed primarily to the practical business and management issue of producing useful and reliable software. No consensus exists among philosophers for a definitive answer to the great question, "What is Truth?" Therefore, from a practical point of view, something else must (and can) be done. That something else is the understanding of effective and realistic ways to put into people's hands the best software systems possible. They won't be perfect; they won't work "100%" (or, if they do, this cannot be conclusively demonstrated). But our job as system developers, sponsors and project administrators is to make them come as reasonably close as possible. This is the goal of the research described in this thesis.

# CHAPTER 2. Model Validation Literature

This study identifies and organizes over 90 types of validation methods reported in the literature. Similar attempts at such a synthesis are few. Finlay & Wilson (1992) published a list of 50 methods, and noted that theirs was probably not an exhaustive list. They did not attempt to categorize these various methods into groups. Balci (1994) uses a taxonomy of 45 validation, verification, and testing techniques in six categories.

The terms verification and validation are so closely related that it is easy to understand that they are frequently confused.[3] The way they are used in the software industry is simply a matter of convention; in Europe the accepted definitions are opposite from those used in North America. Traditionally, verification and validation have been viewed as two distinct, but related, processes. However, Ho Kang *et al.* (1996) notes a new trend, stating that verification and validation "are not separate tasks, but ... continue throughout the lifecycle of the system." Kleijnen (1995) notes that the "V&V" terminology has not been standardized.

Imprecise and inconsistent use has led to interesting oddities in the literature. A case in point: Two exactly opposite uses of the terminology were published in the same year (*cf.* Vick and Lindenmayer, 1988 and Castore, 1988). Finlay et al. (1988) point out that "British modelers tend to switch the definitions, and use validation for verification and *vice versa*." Thus the field of validation suffers from confusion as well as a proliferation of terminology, which works against a synthesis of methods. Contributing to the problem are the following:

**Multiple names for similar meanings.** Many validation types that are

[3]The Oxford English Dictionary (1898, revised) uses 'valid' in the first definition it lists for 'verify'.

called by different descriptive names are similar to and often overlap considerably with other types but are still different enough to reference individually. It is not unusual for different names to be used by different writers for nearly the same validation methods. For example, validation based on first impressions by experts is variously called Face Validity (Saunders, 1985; Landry *et al.*, 1983), Positive Initial Reaction (Gass, 1983), and Subjective Validation (Finlay & Wilson (1991).

**Same name for different meanings.** Substantially different validation methods might be identified by the same name by different authors. For example, Landry *et al.* (1983) define Predictive Validation as comparing a model's predictions to the real behavior of the system, but Gass (1983) defined Predictive Validation as comparing a model's predictions with predictions that human experts would make. Black Box testing is another good example.

**Validation methods proliferate.** Because conclusive validation is such an elusive goal, the myriad attempts at validation have produced a plethora of methods. Attempts to validate individual systems usually consist of a selected few of the hundreds of validation methods reported in the literature. For proper communication to occur, each of those steps or methods needs a name, whether it is performed alone or as part of a longer process.

Validation is a theoretical process that in actual practice is never accomplished fully, and therefore validation of actual models is never finished. However, the process of *attempting* to validate models seems to accomplish two purposes: (1) it develops model builders' skill in determining (and therefore designing) reliable models, and (2) it enhances confidence in the model's usefulness.

A review of the validation literature follows. For purposes of synthesis, methods have been organized into three major groups: Technical Methods, Semi-Technical Methods, and Human Judgment Methods. A total of 27 subgroups are described.

## 2.1 Technical Validation

**Traditional Verification.** The distinction between verification and validation has been discussed extensively by Morehead (1990) and other authors. Verification is not validation, but many writers make the case that it is part of the initial stages of a validation process. This first validation category, the first of 27 groups of methods that progress from the most strongly technical to the most highly judgmental, should be the dividing line between "real" verification and "real" validation. In theory, perhaps this is so, but in practical terms it is not. This study set out to study *validation*, and it meets that goal. But one of the very important discoveries of this study is that we *cannot* cleanly analyze, or even discuss, validation without acknowledging verification. Unless and until there is more effort (and justification) made to distinguish and categorize test methods as either verification or validation, the two terms and their effects on judging the usefulness and reliability of systems will necessarily be intertwined. "Implicitly, validation includes verification" (Plant and Preece, 1996).

There are always discussions of verification in the validation literature, so the first group in the Technical Validation section begins with five verification definitions. First in this group is Compilation & Execution (Saunders, 1985), the primary indication that a system will perform. Verification is concerned with showing that a program or model is internally correct, i.e., that it performs exactly as its programmer intended it to perform (Gass, 1983). A common method is to analyze all of the equations in the program code to determine that they are implemented correctly, as did Green and Kolesar

(1989) in analyzing a police patrol car queuing model. Application of verification is not limited to traditional algorithmic programming. Valluy *et al.* (1989) used a verification method for consistency and completeness (Static Analysis) to analyze the rules in a knowledge base called *Microbe* that provides treatment plans for patients with bacterial infections.

**Mathematical Treatments.** All computer programs are designed on the basis of logic and mathematics. Mathematical testing techniques attempt to show that calculations are accurate (Gass, 1983), that appropriate variables and values of variables are used (Saunders, 1985), that formulas express correct relationships, and the like, in the strict mathematical sense. Mathematical treatments such as formal proofs of correctness (Davis, 1989) are certainly the most rigorous. Finlay *et al.* (1988) provide a good discussion of how traditional mathematical and logical testing methods contrast with currently evolving methods to validate expert systems.

**Statistical Treatments.** Using sound statistical procedures to determine the probability of legitimate outcomes has been commonly used since the early days of computer programming (O'Keefe *et al.*, 1987). It is realistic to accept that most programs cannot be exhaustively tested for all possible combinations of values and relationships. Therefore a good statistical result on a representative sample of model input/output has long been the standard for a "good" model, such as the Statistical Conclusion Validity method (Straub, 1989).

**Complete Model Checking.** In this type of testing, each variable is tested with every possible value it can represent. All possible combinations of variables and the formulations representing relationships among the variables are identified and tested. All progressions of logic through a program (e.g., paths through a knowledge base) are traced thoroughly to determine if they end in legitimate results. It would appear that this can be accomplished only for extremely small models in extremely limited domains, but Enand *et al.* (1990) claim that they developed the technique named *Testbench* that is successful in exhaustively validating large knowledge bases.

**Testing Representative Parts of the Model.** This sampling method employs statistical techniques that assume that if certain parts of a model work well, then the entire model should work as well (Hollenbeck & Whitener, 1988). It has been a standard validation method in the field of Operations Research (Landry, *et al.*, 1983). Such testing can be convincing if all parts, or modules, of the model are essentially the same in structure and function.

**Model Sensitivity.** It is important to know how robust the model is to variations and possible inconsistencies. It is desirable that a model exhibit consistency in the relationship between its input and outputs. Most sensitivity testing involves systematic altering variables and parameters to see how output is affected (Gass, 1983; Saunders, 1985; Landry *et al.*, 1983). Sensitivity analysis also can be studied by changing key assumptions (Green & Kolesar, 1989), or investigating if the output could have been caused by factors other than those specified in the model. Straub (1989) presents an interesting example of the latter in a discussion of validating questionnaires on computer fraud and abuse in business.

**Philosophical.** All modelers would like to prove the absolute correctness of their models. The entire field of validation is built on this goal, but the goal is limited by the contrary argument that absolute proof is not possible. Philosophers (Popper, 1963) have shown that hypotheses cannot be proved; instead, we can only falsify or fail to falsify them. One can never say a theory is "proved", because there always exists the possibility that a counterexample will occur. It is, however, persuasive to be able to say that a counterexample has not been found *yet*. For example, if a model is run thousands of times and never produces erroneous results, this is evidence that the model can be relied on to continue to give good results. It is not an absolute validation of the model, however. That, according to Popper, is never possible. Both Loehle (1983) and Caswell (1976) propose the vigorous application of Platt's (1964) Strong Inference as a workable procedure for refining hypotheses and thereby improving the validity of ecological models.

**Comparisons with Other Models or Ideas.** Models are often built to improve upon, replace, or compete with other existing models. There are many ways to structure comparisons between models: simulation results can be compared to the results of simpler analytic models of the same problem (Ignall *et al.*, 1978). Falk and Gordon (1978) compare their financial risk assessment model to other, apparently competing, models in the field. Scarl *et al.* (1987) wrote an expert system to test an algorithmic model of problems that can occur in a liquid oxygen system. Model comparison may also happen by surprise: a linguistics model describing how Native Americans settled in the New World was independently (and unexpectedly) corroborated by a study of the genetics of the same people (Greenberg & Ruhlen, 1992). This validation group is in the Technical category because the methodologies included here rely primarily on quantitative comparisons to determine how closely there is a match between the various sets of events.

**Redundant Model Creation.** If two or more programmers (or teams of programmers) are given the same problem to solve (same input and same output units), a strong case can be made that the problem has been solved correctly if their outputs match, even if their models of the problem are different. If the results of these different models are not similar enough, then this is taken as evidence that the problem definition was not clear enough, and it must be restated with more detail. This testing method is among the most expensive. It is called for in military applications (Davis, 1989) and situations where safety concerns allow no tolerance for failure (Miller, 1989).

**Modeling Language.** Language tests attempt to determine if the most appropriate language has been chosen to model a particular problem, based on characteristics of both the language and the nature of the problem modeled. (Landry *et al.*, 1983). Fortran is an efficient language for applications requiring mathematical formulas, but awkward for working with knowledge-based systems.

**Client Acceptance.** There has long been a tradition in computer programming that programmers choose and perform specific technical tests on program code. These have generally been traditional verification tests, but include some of the technical validation tests as well. If the programmers can certify (verify) to the client that the program code has passed the specified tests, then the client accepts these test results as the basis for accepting and implementing the program (O'Leary, 1987).

## 2.2 Semi-Technical Validation

**Model Structure & Data.** The techniques included here are based on Verification and other Mathematical testing in the previous section, but with the acknowledgement that numerical test results are interpreted qualitatively to a significant extent. These methods are widely used in Operations Research. The method called Structured Data Validity is described as "verification with interpretation" (Gass, 1983). Landry *et al.* (1983) notes that data validation has to be done within acceptable cost limitations.

**Total Model Analysis.** This method parallels Complete Model

Checking in the previous section but is not a completely automated analysis. The emphasis here is on determining how well the whole model performs, using techniques such as scenarios and user observations combined with automated testing. Complete checking is expensive, but the cost is justified in large systems built to ensure health, safety, and defense. Davis (1989) describes how hand-analysis methods for checking computer memory printouts for AWACS systems developed at Boeing were expanded with expert system software. Enand *et al.* (1990) report that they were successful in developing a tool that finds all possible paths through large knowledge bases, and then making a judgment about the similarity between the system and the expert that the system represents. Pappas and Remer (1984) note, however, that early attempts to include everything about a problem in a model and testing to ensure that models were "sufficiently faithful to the real-life process" in the business world only ensured, more often than not, that the models became obsolete before they were ever put to use.

**Representations Analysis.** It is assumed that a model cannot be correct if the variables do not accurately represent reality. This analysis attempts to determine if a model's variables describe or represent measures of the events of interest without error or bias (Straub, 1989).

**Replicating the Past.** A common way to test a model is to see how well the model "predicts" the past. This is retrospective analysis. The model is run with input from a situation that has already occurred, and the output is compared with the results of the real event. The comparison is more straightforward the more the output of both the model and the real system can be represented quantitatively. Examples are found in Elleby *et al.* (1987) regarding scheduling semiconductor manufacturing, and in Recknagel and Benndorf's (1982) Retrospective Scenario Analysis of models of water ecosystems. Other applications of Historical Validation have been in global modeling (Richardson, 1978), leadership and management (McCall & Lombardo, 1982), and inventory management (Baglow, 1977).

**Comparisons to the "Real World".** Many models try to mimic the behavior of a system dynamically. The state of the model can be compared at a given time to the actual state of the system. Loehle (1983) cites Cutler (1980) in arguing that it is far more reasonable to test ecological and biological models by comparing them directly to the natural systems being modeled, rather than to apply techniques from computer science. The comparison can be done from many aspects of interest. For example, the structure of the model can be compared to the structure of the real system, or the behavior of the model can be compared to the behavior of the system (Barlas, 1989). Barlas reports on both Structural Validity and Behavior Validity testing of Systems Dynamics models. Also Structural validity has been defined to include examining the key assumptions of the model (Green & Kolesar, 1989).

**Predicting the Future** The ultimate test of a model is how well predictions compare to results for a situation that has not happened yet. For models with short time frames, this method is practical. Examples are medical diagnosis (chest pain analysis system in Hudson *et al.*, 1984), short-term investment models (Gale, 1978), and utility industry load models (Mayer, 1980). Predictive Validation is basically a "Black Box" method, for it does not matter how the model is constructed if it can make accurate predictions. If automated tools are employed to help make a validation judgment, this is called Objective Black Box Validation (Finlay & Wilson, 1991). Black box testing was the third of five stages of validation reported by Green and Kolesar (1989) for a police dispatch queuing model. This validation method is clearly more difficult in situations that happen infrequently or rarely (e.g., risk models of nuclear power plant explosions, or for systems with very long time horizons (e.g., forest management models: Sterba *et al.,* 1995).

**Input/Output Comparisons.** These techniques are very similar to Sensitivity Analysis in the previous section, but with qualitative interpretation of the results by an expert (Enand *et al.*, 1990). Establishing controlled conditions such as laboratory and computer environments are important in these methodologies. Graphic animation and interactive visual tools have been developed for input/output tracking through knowledge bases (Bell, 1985, 1989; Richer & Clancey, 1985). Graphical representation tools have been built into some very sophisticated hazard assessment software, allowing users a "coarse yardstick" with which to judge the robustness of the models without much knowledge of or reliance on how the programs work (Fedra *et al.*, 1987 and Fedra, 1988).

**Performance Comparisons.** This is another form of testing model behavior against events and conditions in the real world. The emphasis is on understanding the real world well enough to judge how well the predictions fit it. Many early efforts in Operations Research were along this line (O'Keefe *et al.*, 1987; Gass, 1983), including the use of linear programming in engineering (Orden, 1979).

**Model Maintenance.** Because of cost it is important to consider how long the model's productive lifetime will be. Productivity and length of use are determined to a great extent by how well a model's design allows for continued improvement. Rapid prototyping incorporates model design, testing, and actual use together as an iterative process. Sacerdoti (1991) compares prototyping of expert systems to black box testing, because the focus is on quickly developing usable results. Prior to the concept of rapid prototyping, it was recognized that systems have dynamic lifecycles that call for periodic maintenance (Gass, 1983). Now it is common for programmers as well as clients to accept that models are rarely finished, and should continue to be refined and expanded as both modelers and users discover new and useful knowledge and ways to represent the knowledge. This is model maintenance.

## 2.3 Human Judgment Validation

**First Impressions.** This method stresses the importance of how potential users regard the model when they first encounter it (Oxman, 1991). A simulation model of wilderness recreation (Shechter & Lucas, 1980) was tested by asking staff and investigators for a judgment, based on their experience. This is not a comprehensive evaluation, but rather a determination made about how a model looks after only a preliminary introduction (e.g., "makes sense", "looks OK"). Sometimes this is all a busy client has time for, and the model is accepted or rejected depending only on whether it passes this quick test. Uses of Black Box testing may fit in this category if judgment of the output is reached from people's opinions only (Long & Neale, 1990, on a life insurance model). This is Subjective Black Box testing, as distinguished from Objective Black Box testing (see Predicting the Future, above), which employs automated tools to assist humans in assessing validation.

**Expert Opinions.** Comprehensive evaluation by experts is one of the most common methods used today to evaluate complex models built for decision making. The Delphi technique is a method involving multiple experts combining their ideas over a period of time that has been used with success in medicine (Kors *et al.*, 1990, on electrocardiograms) and in business (Saunders, 1985). Less formally, it is common to ask a group of expert users to judge the content of a system (e.g., wilderness recreation managers by Shechter & Lucas, 1980, and educators by Straub, 1989). This may be a time-consuming and extremely detailed evaluation, with varying amounts of assistance from computerized testing techniques, but for many systems the final decision on implementation comes down to whether or not knowledgeable people are satisfied enough with the system to use it (O'Leary, 1987).

**Comparisons to Expert Sources.** Similar to testing methods that compare model output to past events, this testing technique attempts to compare model predictions with other sources considered to be expert. Comparisons can be to published literature (Buchanan & Shortliffe, 1984, on the medical expert systems MYCIN and CADUCEUS). MYCIN was unique in its time for many reasons, including that a primary method used for its validation was the Turing test. In the famous Turing test, the goal was to determine if the performance of the model is indistinguishable from the performance of human experts (Turing 1963; Rich, 1983; Charniak & McDermott, 1987). The Turing test is not interested in model structure, but only with how realistic the output appears to be (Schruben, 1980). Other Expert Source techniques are Scenario Testing (Derkinderen & Crum, 1988, on Strategic Decision models) and comparison of opinions of two or more experts to the results of a model (Long & Neale, 1990, on life insurance).

**User Participation.** Often the users of a model are excellent judges of the reliability of the predictions, even if they are not domain experts.

**User Participation – Initial Design Phase.** Soliciting advice from potential users at the beginning of model development can give model developers a realistic view of both the content and acceptable boundaries of the problem to be modeled. This is commonly done with Focus Groups to develop marketing strategies (Percy, 1981), formal presentations of the assumptions in a manufacturing model (Law & McComas, 1990), and involvement of the users from the initial conceptualization (many authors, including Shechter & Lucas, 1980).

**User Participation – Model Development.** Involving users in the model development process can provide modelers information on whether users will eventually adopt the model for its intended use (Naylor, 1978, commenting on Gale, 1978). Such features as how easy it is to learn, whether the model uses understandable terminology, clarity of instructions and the user interface, and understandability of the model components will all determine if users will use the model (Straub, 1989). User-friendly features of higher level software (e.g., expert systems development shells) actually allow users to trace rules through a knowledge base and determine if they are correctly used for test cases (Oxman, 1991). In the process, developers have the opportunity to discover technical nuances or important considerations that were not captured in the original design but which could invalidate the model if ignored (Landry, *et al.*, 1983).

**User Participation – Testing.** Traditionally, models were designed, programmed, and tested in the lab and then sent out for field testing by users (Cochran & Hutchins, 1987, on field testing a Honeywell refrigeration maintenance system). Digital Equipment Corp. abandoned this tradition in developing the first large, commercial expert system (known as R1 and XCON) for configuring computers. Their success was largely due to continual feedback by users (Bachant & McDermott, 1984). With rapid prototyping, this method of User Participation is essentially the same as User Participation in Model Development (O'Leary, 1987). It is common to build models today in modules that can be tested, either as prototypes or as parts of the complete system, almost immediately after programming begins. Model developers generally agree that testing is a continual process that does not end with formal model implementation but continues as maintenance and expansion of the model throughout its life (Bachant & McDermott, 1984: "It is difficult now to believe that R1 will ever be finished.") .

**Model Improvement.** The process of constructing a model can clarify understanding of the system under study, resulting in a corresponding improvement in the model itself. The performance of the expert can even be improved (Mitroff, 1969). Rigorously examining the original hypotheses in relation to the actual model code can result in more realistic results, especially in analyzing games and simulation models (Hermann, 1967; Emshoff & Sisson, 1970). Iterative model testing combined with Platt's (1964) strong inference can lead to a refinement and improvement of the model's original theories. Thus, the model is improved by improving the underlying hypotheses. Caswell (1976, 1983) proposes extensive use of this approach for improving ecological models.

**Client Judgment.** Clients may have pre-established criteria for judging model reliability, without regard for formal validation methods. A client may choose a model developer simply on reputation of past success, then implement the model with little emphasis on formal testing (Finlay *et al.*, 1988; Finlay & Wilson, 1991).

**Economic Effects.** The costs of model development, maintenance, and user training are important aspects of the success of a model (O'Leary, 1987). The trade-off between the costs of extensive testing and the benefits of getting a system into use must be weighed, especially in a business environment. Cumming *et al.* (1976) used economic feasibility as the final phase in testing a planning model for regional blood supplies, asking if the model will improve the current system enough to justify the expense of implementing it at a large number of sites. In business, even the most valid system may not be worth building if it is not economically feasible to implement it. Sharma (1996, personal communication) names this idea as "business value" or dollar content value. He notes that a measure of such a return, while difficult to construct, would be a very useful validation measure.

## 2.4 The Spectrum of Validation Methods

To date there has been no compilation of validation methods that have actually worked with *large numbers of models.* The literature shows a large number of individual validation methods that have been used successfully in testing individual models. Progress towards synthesis could be made if certain validation methods, or groups of methods, could be found that are successful on a large and diverse number of models.

After the preceding extensive examination of the model validation literature, the conclusion is that there is no coherent theory or methodology for the validation of complex computer models designed to assist people in solving problems and making decisions. The few surveys of the validation literature support this conclusion (Finlay & Wilson 1992; Balci 1994). It does not follow that a useful and cohesive synthesis of these myriad validation methodologies cannot be found, however. To this end, the following Validation Spectrum is proposed.

After analyzing the extensive literature on model validation, methods are organized in a meaningful progression from the most technical, or quantitative, methods to the most qualitative, or purely judgmental, methods. This progression is termed the "Validation Spectrum." This Validation Spectrum can be used as a tool to see if any patterns occur, especially trends in method usage over time (historical patterns), or groupings of types of problems that have been successfully modeled and tested (domain patterns).

The Validation Spectrum is arranged into the following progression of validation methodologies, from Technical, through Semi-Technical, to Human Judgment:

Technical
Semi-Technical
Human Judgment

**Figure 1:** Spectrum of Model Validation Methodologies

- **Technical Validation** (Table 1a): uses mechanical, mathematical, statistical, algorithmic, or other logical techniques to exhaustively analyze a model, with the result of the analysis represented quantitatively. Some of these methods closely resemble model *verification.*
- **Semi-Technical Validation** (Table 1b): uses technical analysis methods combined with interpretation of the results by humans.
- **Validation based on Human Judgment** (Table 1c): makes little use of technical methods, relying primarily on human judgment. Model testing and evaluation based on the opinions and judgment of people knowledgeable about the modeling methodology, problem domain, or how the model will be used.

Tables 1a, 1b and 1c (next four pages) recapitulate the 11 Technical, 9 Semi-Technical and 7 Human Judgment validation methods described earlier in this Chapter, along with references for the interested reader. An expanded listing, comprising 38 Technical, 44 Semi-Technical and 32 Human Judgment validation, including examples of their use, found in the literature, is given in Appendix A.

Table 1a. TECHNICAL VALIDATION METHODS

| Group ID # | Validation Group Name | Validation Methods Name | Methods References |
|---|---|---|---|
| T-1. | Traditional Verification | Compilation & Execution | Saunders, 1985 |
| | | Verification | Saunders, 1985 |
| | | " | Gass, 1983 |
| | | " | Green & Kolesar, 1989 |
| | | Static Analysis | Valluy *et al.*, 1989 |
| T-2. | Mathematical Treatments | Mathematical Validity | Gass, 1983 |
| | | Formal Proofs of Correctness | Davis, 1989 |
| | | Checking Operational Variables | Saunders, 1985 |
| | | | Finlay *et al.*, 1988 |
| | | Logic Model Validation | Sell, 1985 |
| T-3. | Statistical Treatments | Quantitative Validation | O'Keefe *et al.*, 1987 |
| | | Statistical Conclusion Validity | Straub, 1989 |
| | | Statistical Tests | Saunders, 1985 |
| T-4. | Complete Model Checking | | Finlay & Wilson, 1991 |
| | | Analytical | Finlay *et al.*, 1988 |
| | | Exhaustive Validation | Enand *et al.*, 1990 |
| T-5. | Testing Representative Parts of the Model | Synthetic Validity | Hollenbeck & Whitener, 1988 |
| | | Tracing | Landry *et al.*, 1983 |
| | | Selective Validation | Enand *et al.*, 1990 |
| T-6. | Model Sensitivity | Internal Validation | Landry *et al.*, 1983 |
| | | " | Straub, 1989 |
| | | Internal Validity | Saunders, 1985 |
| | | Sensitivity Analysis | Landry *et al.*, 1983 |
| | | " | Straub, 1989 |
| | | " | O'Leary, 1987 |
| | | Sensitivity Testing | Green & Kolesar, 1989 |
| | | Variable-Parameter Validity | Gass, 1983 |
| T-7. | Philosophical | Induction | Caswell, 1983 |
| | | Refutation | Caswell, 1976 |
| T-8. | Comparisons with Other Models or Ideas | Empirical Validation | Falk & Gordon, 1978 |
| | | Simulation Models | Ignall *et al.*, 1978 |
| | | Expert System | Scarl *et al.*, 1987 |
| | | Model Significance | Green & Kolesar, 1989 |
| | | Independent Corroboration | Greenberg & Ruhlen, 1992 |
| T-9. | Redundant Model Creation | Dual Programming | Davis, 1989 |
| | | Triple Redundancy | Miller, 1989 |
| T-10. | Modeling Language | Logical Validation | Landry *et al.*, 1983 |
| | | Experimental Validation | Landry *et al.*, 1983 |
| T-11. | Client Acceptance | Formal Acceptance Testing | O'Leary, 1987 |

**Table 1b. SEMI-TECHNICAL VALIDATION METHODS**

| Group ID # | Validation Group Name | Validation Methods Name | Methods References |
|---|---|---|---|
| S-1. | Model Structure & Data | Model Validity | Gass, 1983 |
| | | Data Validation | Landry *et al.*, 1983 |
| | | Raw Data Validity | Gass, 1983 |
| | | Structured Data Validity | Gass, 1983 |
| S-2. | Total Model Analysis | Rigorous Hand-Analysis | Davis, 1989 |
| | | Procedural Validation | Enand *et al.*, 1990 |
| | | Bottom Up Validity | Pappas & Remer, 1984 |
| | | Subsystem Validation | O'Keefe *et al.*, 1987 |
| | | Synoptic Validation | Finlay & Wilson, 1991 Finlay *et al.*, 1988 |
| S-3. | Representations Analysis | Construct Validity | Straub, 1989 |
| | | Discriminant Validity | Straub, 1989 |
| | | Convergent Validity | Fischer, 1977 Straub, 1989 |
| | | Concurrent Validity | Straub, 1989 |
| | | Predictive Validity | Straub, 1989 |
| | | Instrument Validation | Straub, 1989 |
| S-4. | Replicating the Past | Retrospective Scenario Analysis | Recknagel & Bennsdorf, 1982 |
| | | Historical Validation | Landry *et al.*, 1983 |
| | | Replicative Validity | Gass, 1983 |
| | | " | Finlay & Wilson, 1991 Finlay *et al.*, 1988 |
| S-5. | Comparisons to the "Real World" | Events Validation | Landry *et al.*, 1983 |
| | | Spectral Analysis | Landry *et al.*, 1983 |
| | | Structural Validity | Gass, 1983 |
| | | " | Barlas, 1989 |
| | | " | Green & Kolesar, 1989 |
| | | Program Proving | Loehle, 1983 |
| | | Behavior Validity | Barlas, 1989 |
| | | Pattern Prediction Testing | Barlas, 1989 |
| S-6. | Predicting the Future | Prospective Scenario Analysis | Recknagel & Bennsdorf, 1982 |
| | | Predictive Validation | Landry *et al.*, 1983 |
| | | " | O'Keefe *et al.*, 1987 |
| | | Predictive Validity | Gass, 1983 |
| | | " | Finlay & Wilson, 1991 Finlay *et al.*, 1988 |
| | | Black Box Objective Validation | Finlay & Wilson, 1991 |
| | | Output Validity | Green & Kolesar, 1989 |
| S-7. | Input/Output Comparisons | Domain Validation | Enand *et al.*, 1990 |
| | | Visual Interactive Interaction | Enand *et al.*, 1990 |
| | | Experimentation | Landry *et al.*, 1983 |

**Table 1b.** *-(cont'd)*

| Group ID # | Validation Group Name | Validation Methods Name | Methods References |
|---|---|---|---|
| S-8. | Performance Comparisons | Physical Validity | Gass, 1983 |
| | | Implementation Validity | Gass, 1983 |
| | | Operational Validity | Gass, 1983 |
| | | Qualitative Validation | O'Keefe *et al.*, 1987 |
| S-9. | Model Maintenance | Dynamic Validity | Gass, 1983 |
| | | Prototyping | Sacerdoti, 1991 |
| | | | Miller, 1989 |
| | | Lifecycle Validation | Balci, 1994 |

**Table 1c. HUMAN JUDGMENT VALIDATION METHODS**

| Group ID # | Validation Group Name | Validation Methods Name | Methods References |
|---|---|---|---|
| H-1. | First Impressions | Face Validity | Landry *et al.*, 1983 Saunders, 1985 |
| | | Positive Initial Reaction | Gass, 1983 |
| | | Black Box Subjective Validation | Finlay & Wilson, 1987 |
| H-2. | Expert Opinions | Delphi Technique | Kors *et al.*, 1990 Saunders, 1985 |
| | | Content Validity | Valluy *et al.*, 1989 O'Leary, 1987 |
| | | " | Schechter & Lucas, 1980 |
| | | " | Straub, 1989 |
| H-3. | Comparisons to Expert Sources | Convergent Validation | Landry *et al.*, 1983 |
| | | Criterion Validity | O'Leary, 1987 |
| | | Turing Tests | Landry *et al.*, 1983 |
| | | " | O'Keefe, 1987 Saunders, 1985 |
| | | " | Buchanan & Shortliffe, 1984 |
| | | Scenarios | Derkinderen & Crum, 1988 |
| | | Intra-Subjective Consistency | Long & Neale, 1990 |
| | | Performance Validation | O'Keefe *et al.*, 1987 |
| | | Literature Comparison | Buchanan & Shortliffe, 1984 |

**Table 1c.** *-(cont'd)*

| Group ID # | Validation Group Name | Validation Methods Name | Methods References |
|---|---|---|---|
| H-4a. | User Participation – Initial | Focus Groups<br>Structured Walk-Through<br>User Involvement | Percy, 1981<br>Law & McComas, 1990<br>Schechter & Lucas, 1980 |
| H-4b. | User Participation – Model Development | Conceptual Validation<br>Rationalism<br>Measurement Accuracy<br>White Box Testing | Landry *et al.*, 1983<br>Naylor, 1978<br>Straub, 1983<br>Oxman, 1991 |
| H-4c. | User Participation – Testing | Periodic Informal Validation<br>Operational Validation<br>Field Tests | O'Leary, 1987<br>Landry *et al.*, 1983<br>O'Keefe *et al.*, 1987<br>Cochran & Hutchins, 1987 |
| H-5. | Model Improvement | Nonformal Test Interrogator<br>Hypothesis Validity<br>Hypothesis Testing<br>Strong Inference | Mitroff, 1969<br>Gass, 1983<br>Saunders, 1985<br>Loehle, 1983<br>Platt, 1964 |
| H-6. | Client Judgment | Faith-in-the-Modeler | Finlay *et al.*, 1988 |
| H-7. | Economic Effects | Cost-Benefit Analysis<br>Economic Feasibility<br>Practical Validation | O'Leary, 1987<br>Cumming *et al.*, 1976<br>Finlay & Wilson, 1991 |

## 2.5 Searching for Trends

To date, 199 examples of applications of various validation methods have been placed on the Validation Spectrum (Appendix A). They are arranged along a continuum of validation methodologies, from the most technical validation through validation that relies almost entirely on human judgement. The fact that validation methods themselves form such a continuum is interesting, but it still does not address the question of which validation methods (or groups of methods) are best for large numbers of problem types or applications. This part of the study is significant because it analyzes validation methods in two additional ways – by publication dates and by domains (i.e., application areas). If either of these two rearrangements of individual validation methods can be shown to demonstrate any patterns or trends, this would suggest new information about the direction in which validation as a discipline is proceeding. For example, a clear pattern of relationships between specific validation methods and specific problem domains might show that modelers have learned over the years how best to use certain validation methods for validating models of specific problems based on the nature of the problems. Knowledge that the appropriate validation methods were chosen to validate a specific model would help in convincing clients that a model has been adequately validated. Or, a pattern over the last forty years showing that some validation methods have been abandoned in favor of more sophisticated, or combined, methods could indicate that modelers are developing sufficiently complex validation methods that better handle today's more complex models. Any such patterns or relationships found could support an argument that validation technology is progressing in a certain direction, and that conclusions can be drawn about which validation methods are appropriate for certain general classes of modeling situations.

An effort to detect historical trends was made by first examining if the nature, emphasis, and frequency of use of the various validation methods has changed over the years. Appendix B.1 lists the number of examples found in each of these three groups, in relation to the year of publication of each validation study. Because the Validation Spectrum includes validation methods discussed in theoretical articles that did not give specific examples of the use of these methods, some of the validation groups have no examples listed in this table.

Appendix B.1 uncovers no evidence of trends for starting, stopping, or changing validation techniques over the years. Rather, it shows that techniques in all general categories have been used since the 1950's, and their use continues in the 1990's. The expectation that the Technical methods were used earliest, before the more qualitative methods, is not true. Interestingly, one of the most controversial, nonscientific of the Human Judgment methods—User Participation in Model Development—is reported in 1951, whereas Technical treatments such as statistical methods and sensitivity analysis appear to have come into use a decade later. Other unusual Human Judgment methods were reported rather early on, such as the Expert/Subject Effect, where the *expert* — not the non-experts — improved his performance as a result of working with the system (Mitroff, 1969), and Theoretical treatments involving repeated hypothesis testing (Platt 1964). The earliest-used Technical methods appear to have been Statistical Methods and Testing Representative Parts of the Model. The earliest Semi-Technical methods (also appearing in the 1950's and 1960's) are closest to the Technical, including testing the Total Model and Representations. In spite of this wealth of information, no historical patterns could be found.

A second effort was undertaken to discover the existence of patterns according to problem domain: whether the methods chosen to validate models are related to specific problem domains. Appendix B.2 (basically a summary of the full Validation Spectrum in Appendix A) lists the major validation groups on the Validation Spectrum with domain examples found in the literature. Problem domains are listed in alphabetical order within their groups; numbers in parentheses indicate the number of replicate cases found in a particular domain (Appendix B.2).

Not only is there no pattern or trend apparent relative to the type of problem domains tested by various validation methods, there is also no evidence that certain validation methods are considered more appropriate or effective in a given domain. Models in medicine, for example, have been validated with Technical, Semi-Technical, and Human Judgment methods. Even though there is a large body of literature on statistical testing of simulation models (i.e., Technical validation), Semi-Technical and Human Judgment methods are also commonly used to validate simulation models. Likewise, models in domains as diverse as business planning, psychology, and the environment show up in all validation categories.

Again the conclusion is that no general trend or pattern emerges in relation to problem domain, whether domains are grouped in order of the validation type groupings, or whether domains are grouped by year. It is clear from the literature that all categories of validation methods have been used on models in virtually all problem domains. Given the general agreement that no one yet knows how to conclusively validate models, this result is not surprising.

# CHAPTER 3. Problem Solving Systems: Three Features

The goal of this research is to determine the extent validation of problem solving software can be meaningfully measured. To do this, we must study two things: Validation, and Problem Solving Systems (PSS). Validation has been examined in Chapter 2. This chapter focuses on three important features of Problem Solving Systems: Technical Complexity, Human Involvement, and Observability.

Validation of software has been defined as the process of determining if the software accurately represents some relevant aspect of reality, and thus consistently generates results that users find correct and useful. How systems are structured also clearly influences whether systems will be judged to be valid. The factors that determine if a system is well structured (i.e., if it accurately represents reality) are the following.

**Technical Complexity** of Problem Solving Systems affects the ability of developers and users to determine whether or not the system generates consistently credible and reliable results.

**Human Involvement** involves designing, building, testing, and using systems. People make the judgment about whether or not a system is valid. The importance of human beings in all phases of the lifecycles of systems cannot be overstated, but this has not been a widely investigated area.

**Observability** has two main components: (1) How frequently the computer system can be run (i.e., repeated under similar circumstances), and (2) How often the real-world event of interest occurs so that its outcomes can be compared to the output of the computer system. Many validation methods depend on repeated testing.

These characteristics of Problem Solving Systems are important determinants of how systems work, and thus whether or not people regard their systems as useful or credible.

## 3.1 Technical Complexity

New measurement methods for all three dimensions proposed for Problem Solving Systems (PSS) analysis – Technical Complexity, Human Interaction and Observability – are examined in this chapter. The first dimension addressed in this analysis of computer-based PSS's is **Technical Complexity**. Developing a consistent and sound way to measure Technical Complexity is the most difficult of the three, and it is also the most widely researched.

To begin, two things must be determined:

- **What** are the features of a PSS that will yield a legitimate measure of the Technical Complexity of that system, and
- **How** to measure these features in a meaningful way.

### Simple vs. Complex

The concepts of complexity in natural phenomena and complexity of nature as a system (or system of systems) are commonly linked. Orchard-Hays (1976) notes that it is this "complexity of the real world ... (that) is the object of systems analysis." Casti and Karlqvist (1980) say that "In everyday language, complexity is associated with structural features such as large numbers of system components, high levels of connectivity between subsystems, feedback and feedforward data paths". In describing complexity of social and biological systems, Casti (1980) contrasts simple and complex systems as follows (Table 2.):

**Table 2. SIMPLE vs. COMPLEX SYSTEMS,** *after Casti (1980)*

| SIMPLE | COMPLEX |
|---|---|
| Predictable behavior: (i.e., no surprises; output can be deduced from knowledge of the input). | Unpredictable behavior: (system behavior not known from knowledge of the input.) |
| Few interactions involving few variables & few feedback/feedforward loops. | Relatively large numbers of variables interacting in rich feedback/feedforward network. |
| Centralized control/decision-making (due to relatively few interactions.) | Decentralized control (i.e., many ways to influence system operation or cause interactions). |
| Decomposable (i.e., ignoring or omitting weak relationships does not significantly alter the system). | Functionally indecomposable: "A complex process is irreducible." |

Rosen (1980) defines a complex system as one that does not follow the Newtonian paradigm (i.e., the physical principles that explain the vast fields of mechanics, energy and planetary motion). Systems such as *information systems* fall primarily outside the Newtonian paradigm, for the reason that they are poorly understood and therefore not completely predictable. Rosen states: "If a system surprises us, or does something we have not predicted, or responds in a way we have not anticipated; if it makes errors; if it exhibits emergence of unexpected novelties of behavior, we ... say that the system is complex. In short, complex systems are those which behave counterintuitively." Unfortunately, Rosen confuses complexity with the lack of strong theories and laws. Relative ignorance of a system's workings may make it appear complex, but such a definition has no utility here.

Virtually all definitions of complexity include the concept of measuring complexity. Packard (1985) reports that John von Neumann, in developing and proving the theory of cellular automata, offered the admittedly heuristic idea that complexity should measure the ability of a system to do difficult and involved purposive operations, but he made no progress toward formalizing this idea. Von Neumann felt that the real biological world was far too complicated to model directly, and developed cellular automata to serve as simpler systems that display some of the essential dynamical features that might form the basis for understanding the physical laws that govern biological behavior (Packard 1985).

Merely saying systems are "simple" or "complex" does not help much; we need a way to quantify the complexity, at least in a relative sense. Casti (1980), in fact, argues that complexity is an issue only in terms of measurement, and only when systems are compared to one another. He says "Complexity cannot be thought of as an intrinsic property of an isolated (closed) system; it is only made manifest by the *interaction* of the system with another, usually in the process of measurement and/or control."

Complexity is like art: everyone agrees that you know it when you see it. In the field of computer science, programmers have for decades recognized what a "complex" program is. The term is used frequently, without definition, regarding computer systems. O'Leary (1987) writes about validating expert systems: "Complexity of the system is one of the most important variables in determining how difficult the validation process will be." Enand and Kahn (1990) describe a system as having "a moderate amount of complexity." Surprisingly, they are sure that their readers know, and agree, what moderate complexity is.

## Degrees of Complexity

Contrasting "simple" and "complex" may incorrectly imply that a system is either simple or complex, and that those two states are unique and identifiable. The notion of *computer system complexity* certainly makes more sense as a relative rather than as an absolute concept.

In the field of Systems Science, complex systems are defined as having many elements and many relationships among those elements. That standard Systems Science definition is a good basis for a definition of complexity, because the word "many" is imprecise and therefore open to taking on various meanings in different circumstances. In addition to *complex* systems, this interpretation clearly allows for the notion that there can be systems that are *not* complex (i.e., simple), *not very* complex, *quite* complex, and so on. In other words, there is a range of complexity in which it might be possible to identify useful degrees of both complexity and non-complexity. To decide on degrees of complexity, there must be a way to perform some measurement of complexity.

The measurement of complexity has long been recognized as an extremely difficult problem. All writers seem to agree that complexity is inevitable, virtually unmanageable, and in general simply too "complex" for the human mind to comprehend without immediately trying to simplify it in some way. Dörner (1987) notes that people have the following problems in dealing with complex systems:

- Regarding TIME: They are more likely to consider situations in the present only, rather than to try to consider the development of processes over time.
- Regarding SIZE: They have difficulty in understanding processes that develop exponentially.
- Regarding STRUCTURE: They think "in causal series instead of causal nets."

Rosen (1980) agrees: "The degree to which a system is complex can be specified ... variously as the dimensionality of a state space, or the length of an algorithm, or as a cost in time or energy incurred in solving system equations," but warns against "regarding as complex any situation which merely is technically difficult."

Both theoretical and practical approaches to measuring complexity have attempted to overcome these limitations. From the literature it appears that the majority of measurement methods are based on the concepts of time and space – concepts basic to all human beings.

### 3.1.1 Measuring Technical Complexity

#### Computation Theory: Space and Time

**Space:** Regarding complexity of computer systems, Gerardy (1981) says that complexity generally refers to "the amount of time or space that an algorithm requires, or, more precisely, the way in which that time or space grows with the size of the problem." He cites Zeigler (1976): "Complexity of a system ... (is) related to the amount of time or memory required for a computer program to simulate it." This suggests that the number of lines of code or the size in kilobytes of a computer program is a measure of the program's complexity. For example, O'Leary (1987) describes this classification scheme for complexity from the field of software engineering:

> "A simple program is one with fewer than 1,000 statements, written by one programmer, with no interactions with any other systems; an intermediate program is one with fewer than 10,000 statements, written by one to five programmers, with few interactions with other systems."

From the above, we get the impression that a more complex program is longer, has more modules, or is in some way "bigger" than less complex programs. This directly opposes another notion: the idea that truly complex programs could be smaller (i.e., occupy less space) than simple programs. Consider the term "elegance" as used by mathematicians – the most succinct formulation that contains everything needed for a proof – implying that the more complex representations of problems take up less space, not more. Casti (1980) supports this with his statement that a complex system is functionally indecomposable while a simple system may contain weak parts or relationships which, when removed, do not really change the system. Csurgay (1988) directly argues for this view of the spatial aspect of complexity:

> "Scientists consider the simplest theory to be the best one, and that if a theory is too *ad hoc*, it is useless. The simpler the theory, the shorter the program, thus the information grasped by the string can be squeezed into a smaller space. The relevance of this notion to information processing, storage and retrieval is obvious: if a binary sequence can be described by a short program, then this means that the information content of it can be 'squeezed' into a small space. Information-based complexity has this spatial character."

Thus there are arguments supporting both the idea that short computer programs are more complex than longer programs, and vice-versa. This is a very interesting debate, but it is not very useful for developing a practical metric of complexity. Problem Solving Systems built for use in the real world tend to be more "long" than short, for two reasons. (1) Modern programming techniques allow preliminary modules to be implemented quickly, with new additions resulting from continual use and testing. Rapid prototyping and user-assisted development processes anticipate that systems will grow, as users find new needs and request better features. Many successful systems continue to develop for years, far beyond the scope of their initial designs. (2) High-level, easy-to-use software development packages allow programmers to write useful systems, but the results are not succinct or "elegant" in terms of the code. Such software requires extremely large amounts of overhead and only under stationary, or static, conditions. For dynamic systems, the number of relationships an element can have with other elements increases, because they do not all have to happen at once, but in sequence over a period of time. In an increasingly large system given enough time, all elements could interact with all other elements. Luhmann says: "Time compensates for the disadvantages of size."

Casti's first three characteristics of complex systems shown in Table 2 (i.e., unpredictable behavior, large numbers of interacting variables, and decentralized control) would seem to characterize the complex systems of Luhmann's definition. All of these characteristics require more time to operate than their more simple counterparts (i.e., predictable behavior, few interactions, one locus of control).

Complex systems require more time to function than do simple systems. They take "longer," in some measurable sense. Unlike the conflicting complexity theories regarding space, there does not seem to be an opposing theory that argues that more complex systems take less time than less complex ones.

## Logic and Efficiency

**Efficiency:** Lopez, Meseguer and Plaza (1989) agree that complexity measures must include measurements of time and space, and they add the concept of efficiency: "Complexity measures ... concern the depth and length of reasoning ... (as well as) efficiency of the reasoning." Their term efficiency refers to (1) the number of steps needed to reach a conclusion, (2) the quantity and relevancy of information required, and (3) the order in which information is requested. Lopez *et.al.* cite Shapiro's (1984) work using proof trees and Tao *et al.*'s (1987) development of a mathematical function. Both of these approaches attempt to determine the length complexity and the breadth complexity of code in knowledge-based systems.

**Logic:** An approach regarding the logic of paradigms is found in Rivest (1977). Studying systems whose objectives could be satisfactorily met both with paradigms that used feedback loops and others that did not, he used gate count (i.e., the logical gates *AND, OR, NAND, NOR*) as a measure of complexity. In assigning values of relative complexity he found that feedback paradigms were more complex than loop-free paradigms with respect to gate count, but that the *systems* that used the feedback (i.e., the more complex) paradigms were themselves less complex. This suggests a hierarchical nature to complexity.

## Hierarchy, Organization and Structure

**Hierarchy:** Casti (1979) lists these aspects of static complexity (as opposed to the complexity of dynamic systems):

- Hierarchical structure
- Variety of components
- Connective patterns
- Strength of interactions

Imagining the last three aspects in a non-hierarchical or "flat" system is quite possible. But one intuitively accepts that even small numbers of components, connective patterns, and interactions can be better understood and managed (i.e., maintained, controlled, and tested) if organized in some way that does not require them to always be treated all at once or as equals. Except in the conceptually simplest of systems, it is usually the case that certain components are accessed more often, that certain paths are traversed more often, and that certain interactions happen more often than others. Forming hierarchical structures to organize this "unequalness" is the efficient response in both nature and computer programming. Casti comments: "By some accounts, the single most overriding consideration in assessing a system's complexity is its hierarchical organization. ... Assuming the validity of this proposition, it follows that the number of hierarchical levels in a given system represents a rough measure of its complexity."

**Structure:** Describing the organization of a system is one way of describing its structure. Bliss, Feld & Hayes (1986) measure structural complexity as part of an evaluation of decision-aiding methodologies. Their measure is: $(N-1)/S$, where $N$ is the number of decisions in a function's structure and $S$ is the number of linkages between decisions. The methodology uses basic decision trees to graph the decision making process, with the $N$'s the nodes and the $S$'s the connections between nodes. A complex structure has a lower score than a simple structure. The authors note: "A simple structure is considered more desirable because it is less confusing to the user and will be more easily programmed. A simple structure is one in which there is a single path to accomplish a function." Simple structures may of course be desirable, but experience shows that it is not long before a simple Problem Solving System turns into something much more complex.

For finite systems where all potential interactions can be known in advance, it is possible to calculate a potential number of interactions using the number of elements in the system. This number of course could be very large. Elements of most systems in the real world (e.g., biological, social, computer) may have potentially unlimited numbers of interactions with other elements, but they still are constrained by time and other physical principles and thus can never realize even a fraction of those states. In practical terms there is no difference between a system for which the number of potential interactions is calculable but so huge that they could never all be attained, and a theoretical system for which the number of potential interactions is infinite.

Only in a very few small systems would each new individual added have even a theoretical chance of interacting with all of the other previously existing individuals. Most systems are organized in a selective way to prevent that from happening. Luhmann (1978):

> "Complex systems are characterized by the fact that they cannot realize the mathematically possible. Their capacity to link up with other elements is limited. Consequently, complex systems must constrain themselves to using only a fraction of mathematically possible relations. They have to proceed selectively."

Luhmann's idea of the time aspect of complexity expands upon this interaction selectiveness: the more time allowed for a system to function, the more interactions that might occur. Selective organization can thus be seen as a hierarchy and having the dimension of a time horizon as well: past, present and future. More complex systems can take more time, and thus can have more points along the time horizon where unique events can occur, as allowed by their structure.

But once a system is complex it is very difficult to make it significantly *more* complex, if it is well structured. Adding just one element or a few rules will not make a PSS more complex, but adding several more interdependent modules probably will. Regarding validation, once a system is complex, it may not matter *how* complex it is. The possibility of conclusively validating a system may end at the point where a system can be said to be complex, by some acceptable measure. Thus the challenge is to find the demarcation line past which Problem Solving Systems can truly be said to be complex. The following Table summarizes the various methods of measuring complexity, along with brief comments about some practical limitations in trying to use these measures.

**Table 3. MEASUREMENT METHODS for COMPLEX SYSTEMS**

| METHOD | Description: What is more complex? | Limitations to Practical Application |
|---|---|---|
| SPACE (Large) | Longer code, more lines | Some computer languages are more cryptic than others; bad programming can produce more code |
| SPACE (Small) | Shorter code, more succinct/"elegant" | Except in specific, directly comparable mathematical formulations, it is intuitive that more code "does" more |
| TIME | More steps, longer execution time | Computers operate at different speeds; Human users greatly influence system operation times |
| EFFICIENCY | Proof trees & mathematical functions | Impractical for large systems; Too complicated for use in business |
| LOGIC | High gate count re feedback loops | Cannot compare unlike systems on this basis |
| HIERARCHY | More modules | Modular structure depends primarily on a programmer's style; Later additions to system may be modular or not |
| STRUCTURE | More branching (e.g., decision trees); Multiple paths to one answer | Complete set of decision-making paths taken by system rarely documented (or understood) |

### 3.1.2 Practical Considerations

The objective is to find a measure of Technical Complexity that can be applied to computerized Problem Solving Systems now in use in business and research. This measure must be:

• Expressible in terms that are understandable to business people, nonscientists and other ordinary PSS users. The measure must not be so mathematical or theoretical that users of the systems will not understand the measure, or why it is being applied.

- Possible to apply during the normal operation of the system.
- Possible to apply using normal resources (e.g., staff time) offered by the organization that owns and/or uses the PSS.
- Possible to apply in a way that preserves confidentiality and respects the proprietary nature of systems to the extent required by the owner of each PSS.

Even in treatments that produce quantitative measures of complexity (e.g., the decision structure formulation by Bliss *et.al.*, 1986), these measures are practical only for very small systems. For large systems, the mathematics gets unwieldy very quickly. Lopez *et.al.* (1984) commenting on the use of proof trees, state: "... the measures ... are defined as the upper bound of all possible interpretations (solutions) of a problem and their proofs. This renders such measures impractical when the KBS is large." And Gerardy (1981) says: "There does not appear to be any widely accepted criterion as to what exactly constitutes a good complexity measure."

Complexity is the bane of validation. If systems were not complex, we would have little (or, certainly, less) problem validating them. None of the complexity measures reported in the literature appear helpful for the practical problem of stating just how complex a system really is (Gerardy 1981). But the problem will not go away. It is commonly assumed that today's systems are more complex than yesterday's, and tomorrow's systems will be more complex than today's. Complexity is and will remain an issue for those who want to try to determine to what extent a system can be relied upon. Therefore, it is important to continue to address the issue of measuring complexity.

Complexity is not a very meaningful idea when assigned to one system alone. Complexity is a relative term. Establishing a "measure of technical complexity" implies the discovery of a metric for comparing technically complex systems.

Some individual factors that clearly contribute to the complexity of a system are not easily or directly measurable. For example, it is reasonable to say that recursive programming structures are generally more complex than non-recursive algorithmic programs. But there is no consensus for practical measure of recursion. More rules, more logical operators, larger data bases or arrays – all certainly make for more complex systems. None of these measures alone, however, satisfactorily represent complexity. And none of the measures are applicable to all systems, because systems are built using many different languages and techniques.

One possible measure of the *total* complexity of systems is to use the notion of "state space" – the total number of different possible states that a system can have. This can be thought of either as the number of (1) internal states of the system, i.e., states occurring during program execution; or (2) output states, i.e., how many unique "answers" the system can produce. For all but the simplest of systems, any number representing the potential state space of the system can be huge; many systems could be said to have an infinite number of potential states. A more realistic problem is that such a measurement would be prohibitively difficult to calculate.

Although it is desirable but virtually impossible to count the number of potential states of a system directly, this idea is nevertheless useful if a proxy measure can be found. Because no one measurement adequately represents complexity, the proxy cannot be one-dimensional, but must be a combination of measures. Using the criterion that the measure should "make sense" to people, the following is a list of features that can realistically be measured in most systems:

- Lines of program (source) code
- Size of system (in megabytes), including data structures and files
- Number of main modules
- Number of input items
- Size of user documentation (in pages)
- Training period for beginning users
- Amount of output generated (pages and screens)
- Length of time to develop system
- Cost of developing system
- Cost of maintaining system (per year)
- Number of separate installations of system

Taken together, this list of features should facilitate the successful characterization of systems along a range of complexity. The Itemized Rating Scale technique was used (Davis and Cosenza, 1985), since it was not necessary that exact numerical values be assigned to these measures, but rather that they were ordered along a continuum. General ranges of values are enough to retain ordinal information, which will allow a ranking of complexity.

## 3.2 Human Involvement

Verification methodologies originated to demonstrate that computer code was correct. Verification depends primarily on mathematical (logical) and statistical techniques. Validation, on the other hand, is rooted in the behavioral sciences. Many of the concepts concerning validity and much of validation methodology have been developed by social scientists in the context of making observations on the real world according to an experimental design

(Finlay and Wilson 1992). Humans, not measuring instruments, ultimately make the decision on whether or not a system is valid. Human Involvement, at many different stages and in many differing roles, clearly is an important factor in determining the extent to which systems are said to be valid.

In contrast to the rich literature on Technical Complexity and Verification, the literature on issues related to Human Involvement is minuscule. Few references exist that elaborate on the influence of humans on computer design, implementation, and use (beyond the obvious fact that humans perform these activities).

Since Turing's discussion of the possibility of an "Intelligent Machine" (Turing, 1950), much has been written about how human thought produces models of reality. Simon (1977) has written extensively on decision making and problem solving. Platt (1964) has proposed how humans can be more effective in developing scientific theories. And Kugel (1986) is one of many writers comparing the human thought process to computing, an idea strongly ridiculed by Dreyfus (1972, 1988).

The influence of humans on the origination of computer models of reality (i.e., programs) is unquestioned. Weizenbaum (1976) maintained that all computer code is a model of reality, and that model originated nowhere else but in a human mind. Computer programs reflect a reality as it was perceived, at least at one time, by a real person. The theoretical approach to this activity is to show that the computer code is a physical (i.e., encoded) model of a person's mental model (i.e., a theory), which resulted from analysis of reality by an intelligent mind (Weizenbaum 1976).

But these theoretical treatments do not go far enough in demonstrating that there is a significant human factor to Validation. Humans influence Problem Solving Systems not only at the original point of the great idea, but throughout the entire existences of these systems, and can greatly affect the success of the systems at all stages. Not only with the original theory or design, but through all stages of development, implementation and performance, it probably should be said that humans are the most important factor in whether a PSS succeeds or fails.

### 3.2.1 Measuring Human Involvement

Respondents to the questionnaire were asked to indicate the numbers of people in 21 categories (3 roles people play times 7 phases). This is basically an open-ended rating scale (Davis and Cosenza, 1985). They were also asked for the grand total of people involved in the system, a number not necessarily the total of the 21 categories, since one person can play more than one role and be involved in more than one phase.

In the past, two distinct groups of people were associated with complex computer systems: programmers and users. Now it is more common to refer to these three groups:

- Experts
- Developers
- Users

It is increasingly common for these groups to overlap – for an expert to also function as a developer, for example. It is not uncommon for one person to play all three roles; many small spreadsheets and database programs are written for personal use by people who know best what they need to accomplish. Others groups exist such as sponsors, clients, and project managers. All of these influence whether or not the goal of a PSS project will be attained: successful, continued use of the system. The very fact that a PSS has been implemented and remains in use can be said to demonstrate that the system is valid.

Computer systems projects typically progress through these stages:

- Problem Definition
- Design
- Development/Programming
- Testing
- Implementation
- Routine Use.

These steps overlap and are iterative. It is increasingly common for all groups to affect all of the above-listed stages of a project, to varying degrees.

The final aspect of human involvement with systems is that humans usually *interpret* the output of systems. Some computerized decision-making systems operate automatically with little human intervention, except in emergencies (e.g., automatic temperature controlling systems, autopilots). But most systems produce results or recommendations that humans must act upon. Humans must decide whether or not to use these results. The success of such human interpretation of system output is crucial to whether or not systems are accepted as accurate representations of reality (i.e., "valid").

## 3.3 Observability

Recall Popper's notion that a hypothesis can never be proven to be true. No matter how many times a hypothesis appears to be true, there is always the possibility that it could be falsified in the future. Just one counterexample is enough to falsify. However, people are likely to be persuaded that a hypothesis is true if it repeatedly turns out to be correct and is never, in their experience, demonstrated to be false. The key feature here is repetition: the number of times the hypothesis can be tested by comparing predictions to reality. This is the basis of observability. The ability to observe frequent occurrences of real events clearly facilitate the validation of the model of the system.

A Problem Solving System is written to mimic an event, or solve a problem, in the real world. Problems or events can be said to have lifecycles, i.e., specific lengths of time from first identification or appearance of the problem to its resolution. Computerized PSS also have lifecycles; their purpose is to represent and run selected aspects of the problem in a shortened amount of time. Testing a PSS depends not only on how often the PSS can be run, but also on how frequently the actual problem occurs and can be resolved, or results obtained.

If the real system is so rare, so dangerous, or its lifecycle so long that it essentially cannot be observed, then a model of that system (a PSS) cannot be tested by comparing the output of the model to reality. Forest management models commonly predict future yield after 50-150 years of growth (e.g., Sterba et al. 1994), a time horizon that exceeds not only the length of a career but a lifetime as well (Monserud 1989). Although such forest simulation models take only seconds to run, the actual event is so long as to be unobservable. Another class of PSS that cannot be directly tested are certain military strategy systems (e.g, nuclear war). Harwell's (1984) predictions of nuclear winter are taken seriously by scientists even though the actual event, hopefully, will never be observed.

### 3.3.1 Measuring Observability

Finding a measure of the Observability of Problem Solving Systems is quite straightforward. Two questions must be answered:

- **How long** does it take to run a PSS, from initial input to producing usable results?

- **How often** does the real situation that the PSS models occur?

As in measuring Technical Complexity, the Itemized Rating Scale technique was used (Davis and Cosenza, 1985). Answers to the questions were given in ordered units of time; the time units are not all the same. For example, a PSS that advises on granting credit to bank customers may take only a few minutes to run (most likely the longest part of a process such as this is data input by a person), but the number of applications for credit is normally reported by the bank as a number per day. Large models that take hours or days to run typically model situations that occur infrequently, or are reported at regular, longer time intervals (e.g., air pollution prediction systems that predict by days or weeks).

It will be possible to measure the following attributes of PSS Observability:

- Frequency of the real problem or event
- Length of occurrence of the real event
- Control humans have in making the event occur
- How frequently the PSS can be run
- Limitations to running the PSS (cost, data, staff, etc.)

Obviously a PSS that runs very quickly and represents a real situation that happens very frequently can be extensively tested. If results are good after many runs, users will believe that it will continue to give good results. They will conclude that the system is built properly and can be relied upon to correctly represent reality. Although this is not formal validation, the users behave as if the PSS has been validated.

# CHAPTER 4. Research Methods

## 4.1 Rationale

This study has the practical goal of providing useful advice about validating Problem Solving Systems to business people, researchers, and other software developers. Thorough examination of the literature failed to uncover any coherent theory or methodology for the validation of complex computer models. Additional examination revealed that the numerous validation methods could be logically organized along a continuous Validation Spectrum, ranging from technical validation methods through semi-technical, to methods relying increasingly on human judgment. While this Spectrum is a useful tool for organizing validation methods, it nevertheless did not reveal any patterns or trends in successful model validation. With the failure to find useful general results after mining the literature, this study moved into the next phase: collecting data directly from users of such systems.

Clearly humans make the final decision about the validity of a system. Every programmer knows that the ultimate test of the program is what the users think. Either they accept it, or they don't. User acceptance is usually based on confidence that the system (1) is constructed properly, (2) was designed and built by people knowledgeable about the problem being solved, (3) has been well-tested, and (4) can be used frequently enough that users will be able to determine if it does its job. Even if a system has been tested objectively with a variety of automated testing tools, the ultimate judgment will be subjective. If an experienced user, or group of users, says that a system appears to be, say, 70% valid, the system is for all practical purposes 70% valid. It matters not if prior "V&V" testing techniques declared the validation of the system to be higher.

All subsequent analysis of validation methods and system features performed herein are done in relation to how valid people think their systems are. Thus a tool was needed to collect data not only on (1) system features and (2) validation methods, but also on (3) reported levels of validation.

## 4.2 Data Collection Tool – The Questionnaire

A questionnaire was designed to register information regarding all of the main groups of Validation Methods discussed in Chapter 2 and the important features of Problem Solving Systems in Chapter 3. The complete six-page questionnaire is listed in Appendix C. The questionnaire was divided into 6 parts, as follows:

Part 1. Description of the System

Part 2. Technical Complexity of the System

Part 3. Involvement of People in the System

Part 4. Observability of the System

Part 5. Reported Verification and Validation of the System

Part 6. Methods Used to Test the System

**Part 1.** Description and background information was collected, including:

- A description the system
- Name (frequently an acronym)
- Year system design/development began
- Year system was put into use
- If system is in use at present time
- Countries in which system is used.

Parts 2, 3 and 4 result directly from the discussion of measurable PSS features in Chapter 3.

**Part 2.** The 11 measures of Technical Complexity indicated in Chapter 3 were investigated:

- Lines of program (source) code
- Size of system (in megabytes), including data structures and files
- Number of main modules
- Number of input items
- Size of user documentation (in pages)
- Training period for beginning users
- Amount of output generated (pages and screens)
- Length of time to develop system
- Cost of developing system
- Cost of maintaining system (per year)
- Number of separate installations of the system.

**Part 3.** Seven phases in the life cycles of systems were listed, and respondents were asked to indicate the number of experts, developers (programmers), and users that had been involved in each phase. The 7 phases are:

- Problem Definition
- Design
- Development / Programming
- Testing
- Implementation
- Routine Use
- Interpretation of Output.

Respondents were asked to indicate their role in the system (e.g., expert, developer, user, client, sponsor, manager) and also were asked for a total number of people who have worked on and used the system.

**Part 4.** Data was collected on the following points that affect the Observability of a System:

- Frequency of occurrence of the real event
- Length of occurrence of real event
- Length of computer run that models the real event
- Can the real event can be made to happen (controllable)?
- Factors that limit repeated runs of the computer system:
  - Cost
  - Data availability
  - Time required to collect data
  - Availability of trained staff
  - Time required to run system
  - Other limiting factors.

**Part 5.** Respondents were asked to indicate, on two scales from 0% to 100%, the level of their confidence that their system was (1) verified and (2) validated. They were also asked whether their system ever gave false results, and, if yes, to write about what happened.

**Part 6.** Twenty-seven Validation methods were listed, in the order in which they are presented in Chapter 2. Respondents were asked to check the methods that had been used to validate their systems in one or more of three major phases of the system lifecycle: (1) Design phase, (2) Programming & Debugging phase, and (3) User phase. Note that these three phases are a combination of the seven phases in Part 3, above.

## 4.3 Data Collection – Population Sampled

Because the goal of this study is to give practical advice to people who construct computer models for practical application, it follows that the population to be sampled was these same model developers. This insured that the samples represent the Problem Solving Systems that are the focus of this study.

The resulting data set represents a wide and interesting variety of systems in many different fields in business, scientific research, and government. People were contacted with systems in very diverse applications — from banking to forestry, from education to county elections. It was intentional to have a broad representation of applications and complexity, and for that reason systems of all sizes and levels of sophistication are found in the data set. An additional feature of the data set that gives it more generality is its international nature: many of the systems are used in many countries, with Europe and North America well represented. The set of systems represented by the data is a realistic representation of systems that are commonly used throughout the developed world.

Ninety potential respondents were asked to fill out the research questionnaire. Forty completed questionnaires were received. Contacting potential respondents and receiving completed questionnaires was accomplished over approximately a 6-month period. In all, 89 data items were requested; not all respondents were able to answer every one of the questions. Virtually every questionnaire has at least a small number of data items missing. Only one questionnaire was discarded because it contained so few answers that it could not be used.

## 4.4 Data

Data were collected on 40 systems. They are grouped into three general areas: Business, Research, and Government. Each system has an identifying number. The following three tables display the domain, countries in which it operates, and the year placed in service for each system:

**Table 4a: Data Collection – Systems used in Business**

Domains, countries, & year first used.

| ID # | DOMAIN | COUNTRIES | YEAR |
|---|---|---|---|
| 7 | Wood Products Industry | Austria, Germany | 1989 |
| 12 | Blast Furnace Process Diagnosis | Austria | 1994 |
| 13 | Software Development | Austria, Germany | 1994 |
| 17 | Sales Analysis (Printing) | USA | 1984 |
| 24 | Inventory Forecasting (Forestry) | USA, Canada | 1977 |
| 25 | Financial Information (Banking) | USA | 1990 |
| 26 | Manufacturing (MRP) | USA | 1993 |
| 27 | Product Allocation & Shipping (Manufacturing) | USA | 1987 |
| 28 | Electronics | USA, Canada, Vietnam, New Zealand | 1995 |
| 29 | Optimal Forest Stand Management | USA, Brazil, Argentina, Greece | 1987 |
| 30 | Electronics Design (CAD) | USA, UK, Holland, Canada, Japan, Germany | 1991 |
| 31 | Sports Equipment (Retail, Wholesale & Catalog) | USA | 1981 |
| 32 | Office Management | USA, Canada, UK, Indonesia | 1985 |
| 37 | Hospital Management | USA | 1993 |
| 38 | Bicycle Racing | USA, Canada, Australia | 1991 |
| 39 | Circuit Board Analysis | USA, UK, Canada, Europe, Japan & other Pacific Rim, South America | 1988 |

**Table 4b: Data Collection – Systems used in Research**

Domains, countries, & year first used.

| ID # | DOMAIN | COUNTRIES | YEAR |
|---|---|---|---|
| 2 | Environmental Impact Assessment | –none– (Demo) | 1990 |
| 4 | Forest Growth Modeling | Austria, Switzerland, Germany | 1989 |
| 6 | Agricultural Policy Modeling | Austria, UK, USA, India | 1985 |
| 8 | Forest Growth Simulation | Austria | 1994 |
| 15 | Global Change Modeling | USA | 1994 |
| 18 | Control Engineering | Austria | 1990 |
| 23 | Research (Education) | USA | 1995 |

**Table 4c: Data Collection – Systems used in Government**

Domains, countries, & year first used.

| ID # | DOMAIN | COUNTRIES | YEAR |
|---|---|---|---|
| 1 | Reforestation | USA | 1990 |
| 3 | Water Quality, Rivers | UK, India, Mexico, Italy | 1984 |
| 5 | Environmental Impact Assessment | Thailand | 1990 |
| 9 | Warehouse Management | Austria | 1993 |
| 10 | Ecology Control | Italy, Austria | 1990 |
| 11 | Forest Management (Forecasting) | Austria | 1995 |
| 14 | Environmental DSS | Netherlands | 1986 |
| 16 | Environmental Assessment/Reporting | Thailand, Kenya | 1993 |
| 19 | Customer Tracking (Social Service) | USA | 1993 |
| 20 | Watershed Management & Control | Austria | 1995 |
| 21 | Records Management (Social Service) | USA | 1993 |
| 22 | Managing Research Data (Education) | USA | 1993 |
| 33 | Patient Tracking (Medical Clinic) | USA | 1987 |
| 34 | Elections (County Government) | USA | 1993 |
| 35 | Forest Regeneration & Management | Finland | 1995 |
| 36 | Equipment Maintenance | USA, UK, Canada | 1985 |
| 40 | Forest Management/Growth Forecasting | Denmark | 1974 |

### 4.4.1 Data: Technical Complexity

Measures of Technical Complexity are most useful in a relative sense (e.g., to compare "small" systems to "large" ones). The data represent systems with a broad range of technical measures. Among the "small" systems are a spreadsheet used in a bank to review performance of existing loans, a database system that helps manage and reorder inventory in a warehouse, and a front end expert system that queries a large Fortran database to find the genetically best seeds for replanting pine trees in Arizona and New Mexico. The "large" systems include a corporate Materials Resource Planning (MRP) system, a hospital management and billing system, and a tool that performs thermal analysis of printed circuit boards. Of the technical information requested on the questionnaire, this is the range of responses:

**Table 5: Technical Complexity** – Range of Responses

| TECHNICAL MEASURE | RANGE of DATA |
|---|---|
| Lines of program (source) code | 1,000 to 400,000 |
| Size of system (bytes) | under 600 kb to over 2 gb |
| Number of main modules | 1 to approx 1000 |
| Number of input items | 0 (automated system) to over 300 |
| Size of user documentation (pages) | no documentation to over 1000 |
| Training period for beginning users | 0 (developed with users) to 90 days |
| Amount of output generated (pages) | 0 to as many as requested |
| Amount of output generated (screens) | 0 to over 100 |
| Length of time to develop system | <1 month to >20 years |
| Cost of developing system | less than \$100 to \$1.5 million |
| Annual Cost of maintaining system | 0 to over \$100,000 |
| Separate installations of the system | 0 (still in development) to over 250. |

The sample data represent a wide range of conditions for every item. For purposes of analysis it was important to combine them into natural groups. It was common for users not to know exact numerical answers to the questions, but to know the approximate range into which their answers fell. Since all of the Technical Complexity questions on the Questionnaire (Appendix C.2) required answers that ranged from small (or zero) to large, a 6-point scale was created to collate the data into usable groupings.

Next, a Technical total for each observation was calculated as the sum of all of the 12 Technical scores. It was hypothesized that systems could be ranked according to overall Technical Complexity with this total score, and that it might bear some relationship to reported levels of validation (and verification).

### 4.4.2 Data: Human Involvement

The data includes systems that are used by from 2 to over 500 people. The systems with the fewest users are the inventory management system mentioned above, the bike racing system (2 users each) and the control engineering research tool in Austria (3 users); the users of these systems were also the systems' designers. In this sample, the largest numbers of people work with systems developed in electronics manufacturing, because these systems are sold as products to large numbers of clients (e.g., CAD design of wire harnesses, thermal analysis of printed circuit boards, communications device for electrical power relays). All are reported to have from 100 to over 200 users, and from 10 to 20 designers/experts.

There are 21 separate Human Involvement scores, each indicating how many experts, developers, and users were involved in each of seven phases of the life cycle of systems (Appendix C.3). Respondents were also asked for the total number of people associated with the system (because this was not necessarily the sum of the 21 separate scores), and the respondent's role in the system (Expert, Developer, User, Client, Sponsor, Manager).

### 4.4.3 Data: Observability

Four measures of observability were taken (Appendix C.4): (1) frequency of occurrence of the real event, (2) length of occurrence of the real event, (3) how often can the computer system be run, and (4) if the real event be made to happen, thus creating test situations that allow more opportunities for the computer model predictions to be compared to reality. The last item was given a score of 1 for a 'yes' and 0 for a 'no'.

The answers to the first three items were scored on a scale of 0 to 7; a higher level of observability indicates more possibility to test the system (and for it to fail). The measurement scale is as follows (continuous frequency is scored 7):

**Table 6: Observability Scores.**

| Obs. SCORE | Observability MEASUREMENT | | |
|---|---|---|---|
| | FREQUENCY - Event occurs 1-10 times per: | LENGTH - Event is timed in: | REPETITION - Computer run takes: |
| 7 | Second | Seconds | Seconds |
| 6 | Minute | Minutes | Minutes |
| 5 | Hour | Hours | Hours |
| 4 | Day | Days | Days |
| 3 | Week | Weeks | Weeks |
| 2 | Month | Months | Months |
| 1 | Year | Years | Years |
| 0 | Many Years | Many years, (e.g., centuries) | Many years |

For example, a frequency value given by the respondent as 3100/year was given the Observability score of 4 because:

3100/year = 258/month = 60/week = 8.5/day is between 1-10 times per day.

Just over half the systems represent real events that occur infrequently (i.e, in years or months). The longest event was a century (optimal forest management). The length of occurrence of the real event is quite evenly distributed through the responses, as is the length of time required to run the computer model of the event. Three-fourths of the events can be induced to occur (i.e., controlled), which may offset – at least for testing purposes – the fact that the real events occur infrequently.

Respondents were also asked to indicate if any of the following five limiting factors impede running their systems to produce output that can be compared to reality. Each of the following was given a score of 1 if the respondent checked it as a limitation, and 0 if it was not a limitation.

- Cost to run system
- Data availability
- Time required to collect data
- Availability of trained staff
- Time required to run system

A total Observability score was created by adding the scores for the first four items, and subtracting the values of the limitation factors.

### 4.4.4 Data: Validation Methods

Questions on 27 different validation methods were asked in the Questionnaire (Appendix C.6) and scored on a simplified Likert scale (Davis and Cosenza, 1985). These methods correspond exactly with the 27 categories in the original organization of validation methods in the literature review in Chapter 2. Respondents were asked to check all those that were used in each of three phases of the system's lifecycle:

- Initial Design Phase
- Programming / Debugging Phase
- Initial User Phase

This resulted in 81 variables describing validation methods. Each method was given a score of 1 if the respondent checked it as a method used, and 0 if it was not checked. A total Validation Score was calculated as the total number of methods checked. Responses were in the range of 9 to 51 methods. It was hypothesized that this total might show a relationship to reported levels of verification and/or validation. For example, a higher Validation Score would mean that more validation methods were used to test a system, which could result in higher levels of trust in the system.

### 4.4.5 Data: Verification & Validation Scores

The final judgment on whether a system can be trusted is made by the people who know it and work with it. Respondents in this study were asked to state the extent to which they considered their system to be verified and valid (Appendix C.5), in terms of percentages along a continuum from 0% to 100% (graphical scale according to Davis and Cosenza, 1985). Seventy percent of respondents consider their systems to be at least 90% verified. Only one respondent insisted that his system is 0% verified.[5] The other 11 respondents judge their systems to be between 50% and 85% verified.

Seventy-seven percent of respondents consider their systems to be at least 70% valid. A teaching system that has been used hundreds of times is still considered by its developer to be only 5% valid, and one system still in development is considered 0% valid. The remaining seven respondents consider

[5]Two examples of the opposite definitions of verification and validation as used by Americans and Europeans, noted in Chapter 2, occurred in responses from Europe. Several quick e-mail exchanges cleared up the confusion in both cases. The American who listed the verification of his system at 0% was also contacted. He did not misunderstand the definitions, and stands by this judgment.

their systems to be between 20% and 65% valid.

These verification and validation percentages were used as the dependent variables in the analysis. Reliably predicting these percentages from measurable features of the system would meet the overall research goal.

A major contribution of this study is that structuring the data collection as described allowed independent estimates to be made for verification and validation. Because respondents were asked for both the verification and validation levels of their systems (see the Questionnaire, Appendix C.5.), independent tests of hypotheses for both verification and validation could be made using the same input data.

People design computer-based Problem Solving Systems to do a specific task. They test the system to see if it does that task well. In most cases they do not test specifically with verification or validation in mind; they just want to know that the system does its job. If the system fails a test, this is information on something that must be improved. They may test the system in as many ways as seem reasonable and that the company can afford, until they have enough confidence in the system to use it. If the system does not fail in testing it is put into use. The process of clients using a system can be viewed as a continuing attempt to invalidate it. Although we do not usually think of using a reliable system as a continuing validation test, this is actually the case.

With this study's Questionnaire we sliced through 40 different systems at one point in each system's lifecycle. At these points the respondents stated their confidence, expressed as a percentage, in the verification and validation levels of their systems. Because these two questions were asked separately, separate hypotheses could be entertained for verification and validation using exactly the same reported test methods and system features. Even though respondents may themselves never have distinguished between the processes of verification and validation, the results obtained here do. The balance and overall view that this gives of people's confidence in their systems is invaluable.

## 4.5 Hypotheses

It was hypothesized that a linear relationship can be found between levels of verification and validation reported by people familiar with systems and certain, measurable features of those systems. Those features describe:

- the Technical Complexity of the system
- the Involvement of Humans in the system
- the Observability of the system, and
- the Methods used to Test the system.

Over two dozen specific linear hypotheses were made and tested in the effort to find these relationships. For convenience, three broad categories of hypotheses were considered:

- System Features
- Validation Methods
- Combinations of System Features and Validation Methods.

System factors are Technical Complexity, Observability, and Human Involvement. Each hypothesis was separately analyzed for the levels of verification and validation reported on the questionnaires. The general form of all hypotheses is:

"The extent to which a Problem Solving System
can be said to be validated (or verified)
is found to depend linearly on ......."

For convenience, each specific hypothesis was rephrased in this shortened form:

"Validation (or verification) depends on ......"

Not all of the hypotheses originally proposed are listed here. Some were briefly considered and discarded. Only those that seemed realistic or hopeful enough early on were carried throughout the many stages of the analysis. For this reason the hypothesis labels are not all sequential or complete. Note that "Verification" was also substituted for "Validation" in all hypotheses, doubling the number of hypotheses tested. The following is the set of hypotheses in each category.

**Table 7: Hypotheses**

**Systems Features Hypotheses:**

| | |
|---|---|
| $H_5$: | Validation depends on the **technical complexity** of a system. |
| $H_6$: | Validation depends on the level of **human involvement** in a system. |
| $H_9$: | Validation depends on the degree to which a system is **observable.** |
| $H_{10}$: | Validation depends on the combination of **complexity, human involvement & observability** of a system. |

**Validation Methods Hypotheses:**

| | |
|---|---|
| $H_7$: | Validation depends on the **methods used to test a system.** |
| $H_{8a}$: | Validation depends on the **technical methods** used to test a system. |
| $H_{8b}$: | Validation depends on the **semi-technical methods** used to test a system. |
| $H_{8c}$: | Validation depends on the **human judgment methods** used to test a system. |
| $H_{11}$: | Validation depends on groupings of **test methods** used in systems. |

**Combined Hypotheses:**

| | |
|---|---|
| $H_{12}$: | Validation depends on summary scores (totals) for **technical complexity, human involvement, observability** & the **test methods** used in a system. |
| $H_{14}$: | Validation depends on the **complexity, human involvement & observability** factors of a system combined with the groups of **test methods** used. |
| $H_4$: | Validation depends on the **complexity, human involvement & observability** factors of a system combined with the individual **test methods** used. |

## 4.6 Analysis

The fundamental task of science is to explain natural phenomena (Pedhazur 1983). This is best done by studying the relationships among relevant variables in the system. If strong theory (e.g., the laws of thermodynamics in physics) is not available for a discipline, then progress can nevertheless be made by empirically discovering useful relations between an important variable of interest (the dependent variable) and other measurable factors (the independent or predictor variables). Regression analysis is an efficient and general method for estimating and quantifying these relationships.

Broadly speaking, regression analysis is a method of analyzing the variability of a dependent variable by using information available on one or more independent variables (Pedhazur 1983). It provides a quantitative answer to the question: what are the expected changes in the dependent variable as a result of changes in the independent variables? At the same time it also is an efficient vehicle for hypothesis testing. If a hypothesis that the relation between a dependent variable (e.g., the validation percentage) and certain observable independent variables is linear, then linear regression can also test the statistical significance of each of the estimated parameters. Thus, a linear hypothesis can be tested, the parameters estimated, and the corresponding linear model determined, all with one efficient procedure. Discussing validation of marketing models in the business environment, Naert & Leefang (1978) and Coates *et al.* (1991) consider such established statistical techniques invaluable for objectively determining model validity.

All hypotheses in this study are assumed linear, and the corresponding model is of the general form:

$$Y = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + ... + \epsilon$$

where $Y$ is the dependent variable, $\beta_0$ is the intercept, the $\beta_i$ are the parameters relating $X_i$ to $Y$, and $\epsilon$ is the random error component. Thus, a unit change in $X_1$ corresponds to a change in $Y$ of $\beta_1$ units, the slope of the regression line. The estimated parameters are called regression coefficients, and are denoted by $b_i$ for the $i^{th}$ predictor variable. The term *regression* is a misnomer taken from an anthropology paper by Sir Francis Galton in 1885, but the term has stuck and is now universally used (Draper and Smith 1981).

For hypothesis testing, the errors $\epsilon$ are assumed to be Normally distributed with mean zero and constant variance. This assumption is quite robust under most situations because of the broad applicability of the Central Limit Theorem (Draper and Smith 1981). Sample sizes of 30 are usually sufficient for the Central Limit Theorem to yield an adequate approximation (Brunk, 1965; Mendenhall and Schaeffer, 1973). This is reassuring, for sample sizes for the hypotheses tested in this study are between 30-40 observations. Note that the independent variables $X_i$ are assumed fixed or measured without error; only the error term $\epsilon$ is assumed normally distributed. Significance of the regression coefficients is judged by a standard $t$-test to determine if they are different from zero. A hypothesis was "accepted" (technically, "not rejected") if all variables in the model are significant at the standard $\alpha = 0.05$ probability level, corresponding to a 95% confidence level. Basically, this means that there is a 1-in-20 chance that a hypothesis that is accepted as true is in fact false. The percentage of variation explained by the regression model is called $R^2$. It is a useful measure of how good the model is and how closely it can predict the dependent variable.

To estimate the parameters, regression uses the least squares criterion invented by Carl Friedrich Gauss at the end of the eighteenth century (Draper and Smith 1981). Given a set of data relating $X$ and $Y$, the method of least squares finds that unique line that minimizes the sum of the squared differences between the observed $Y$ and the predicted $\hat{Y}$ (see Figure 2, next page).

Hypotheses were tested and parameters estimated using the regression procedures in Statistical Analysis System (SAS, 1985). This is a widely used statistical analysis package, with numerous features for examining the data and the residuals.

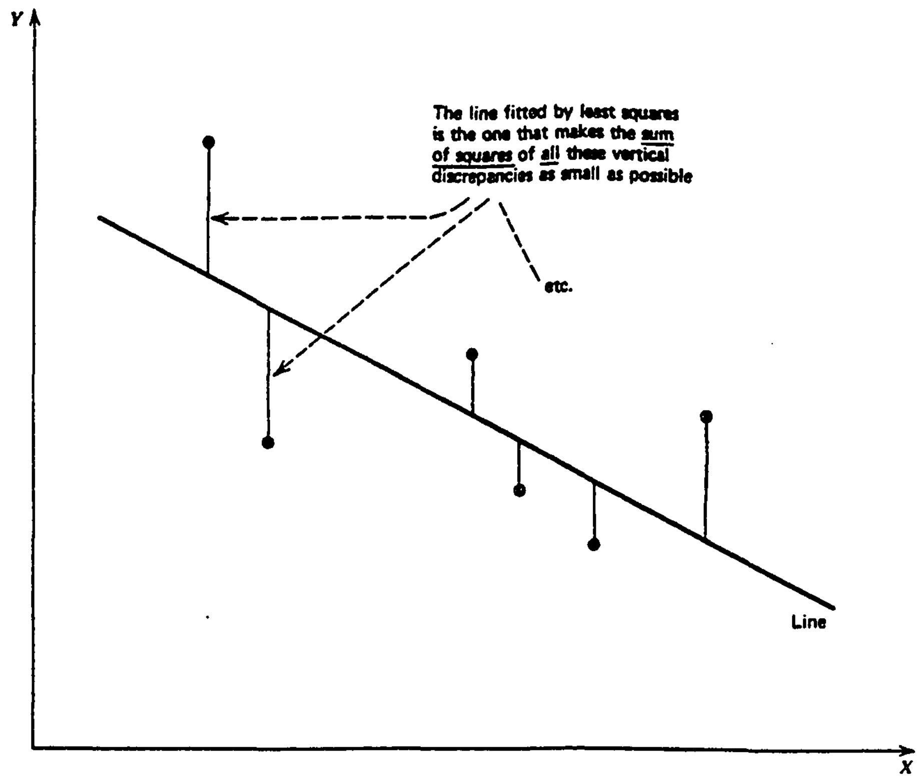

Y
The line fitted by least squares
is the one that makes the sum
of squares of all these vertical
discrepancies as small as possible
etc.
Line
X

**Figure 2:** The vertical deviations whose sum of squares is minimized in the least squares regression procedure (Draper and Smith 1981).

Twelve hypotheses were tested iteratively using regression analysis. Initially, two general hypotheses were proposed: (1) Validation (or verification) depends on the three features of problem solving systems: Technical Complexity, Observability, and Human Involvement; and (2) Validation (or verification) depends on the methods ("V&V") used to test systems. A typical respondent answered most but not all of the questions. These missing values effectively reduced the number of observations available for testing a given hypothesis. The greater the number of variables included in a hypothesis, then the fewer number of observations available to test the hypothesis. Thus, tests of very broad hypotheses that included a large number of variables were the weakest of all, for they excluded the most observations. The solution to this dilemma was to formulate and test a series of more precise hypotheses that used smaller, logical groups of variables. Tests of these more precise hypotheses were much stronger because they used more of the observations. An additional advantage of using smaller subsets of all available variables was that the very useful all-possible-regressions procedure (RSQUARE) could be used in SAS (1985) if the number of variables was less than the number of observations. Clearly, this was impossible with the most general hypothesis with 165 possible variables and only 40 observations.

Each hypothesis test consisted of first performing a thorough stepwise regression (Proc MAXR in SAS) that found the combination of variables that explained the most variation for a given number of variables. These "best models" by definition passed two tests: (1) all variables in the equation were significant at the $\alpha = 0.05$ level, and (2) any equations with a specific number of variables (1, 2, 3, 4, or 5 variables) had a higher $R^2$ value than any other significant equation resulting from the same hypothesis with the same number of variables.

These equations were then reviewed to see if they made sense. With 24 hypotheses, 165 possible independent variables, and a good representative sample of data, it was fairly straightforward to find many models that explained a large and significant amount of variation using regression analysis. The purpose of this study, however, is to find models that can be used to make clear and sensible recommendations on allocating resources and establishing environments to improve the process of building solid, useful Problem Solving Systems. Thus, the significant models had to pass a third test: common sense. When a model found by MAXR did not pass the common sense test, exhaustive regression (Proc RSQUARE in SAS) was run to find all possible models, in decreasing order of $R^2$, until a meaningful model could be found. The significance of this model and its coefficients was then determined with Proc REG in SAS (1985), the standard regression procedure.

Standard statistical procedures were used to examine the estimated models. The most fundamental test is an examination of the residuals (the observed minus predicted values). Graphical tests are best, for any pattern in the residuals that can be detected provides key information on the form of an alternative model (hypothesis). For example, if the residuals form a simple curved pattern, then a quadratic (squared) term is suggested.

# CHAPTER 5: Results

Results are presented for each of the hypothesis tests, in the order in which they were presented in Chapter 4. Results for both verification and validation are grouped together for each hypothesis. In the discussion in this Chapter, the variables follow these general naming conventions:

**Table 8: Variable Naming Conventions & Interpretation**

| PREFIX | SUFFIX | Interpretation |
|---|---|---|
| T | | Technical Complexity system factor |
| H | | Human Involvement system factor |
| O | | Observability system factor |
| VT | | Technical Validation method |
| VS | | Semi-Technical Validation method |
| VH | | Human Judgment Validation methods |
| | CAT | Category (used with Tech Compl. or Observ.) |
| | DEFN | Problem Definition phase |
| | DES | Design phase |
| | PRG | Programming phase |
| | TEST | Testing phase |
| | IMPL | Implementation phase |
| | USR | User phase |
| | INTR | Interpretation phase |

See Appendix D for the full list of variable names and descriptions. A complete listing of the data represented by these variables is found in Appendix E.

## 5.1 Results of Hypothesis Tests

| | |
|---|---|
| | Best MODEL: |
| | 2-variables;$R^2$=42% |
| | 4-variables;$R^2$=59% |
| | $n = 26$ |
| $H_5$: Verification depends on the **technical complexity** of a system. | ACCEPTED |

Two versions of this hypothesis were run: (1) one version with 8 Technical Complexity variables, and (2) one version with 13 variables.[6] The best results reported here reflect the combination of the largest sample size (indicated as '$n$' in the heading of each section) with the highest $R^2$ value for each hypothesis.

The best 2-variable model uses T7A_CAT, representing the number of pages of printed output, and T10_CAT, representing the maintenance cost of the system. The best 4-variable model includes the above variables, and adds two more: T2_CAT, representing the amount of hard disk space required, and T6_CAT, representing training days. This equation explains 59% of the variability in the data. Another model from a larger sample size ($n$=36) contains one significant variable, T1_CAT, that represents the number of lines of program code; however, its $R^2$ value is only 16%.

The formulas for the models described above are as follows:[7]

$$VER\% = 102.81 + 5.3 * T7A\_CAT - 10.5 * T10\_CAT \quad (1)$$

$$VER\% = 105.66 + 4.3 * T7A\_CAT - 11.6 * T10\_CAT + 6.2 * T2\_CAT - 9.2 * T6\_CAT \quad (2)$$

The conclusion from testing this hypothesis is that reported levels of verification are positively associated with higher number of output pages, lower maintenance costs, larger systems, and fewer training days.

**Example**($H_{5ver}$): Observation #31, a system that tracks sales, inventory, purchasing and receiving for a sports equipment retail and catalog sales company, is a good example of the predictions of this hypothesis. The actual

[6]There are two versions each for hypotheses $H_5$, $H_6$ and $H_9$ because of missing values on the questionnaires. The more restrictive subset of variables results in a stronger test because sample size is larger.

[7]For readability, intercepts and slope coefficients are written to one decimal place, even though SAS calculates to the seventh decimal.

reported verification of this system is 92%; the 2-variable model predicts verification to be 92.6% and the 4-variable model predicts 91.5%. Overall, both models predict verification for at least one- half of the observations extremely well (within 10%); the models do not predict well for the one observation (#39) with a reported verification level of 0%.

| | |
|---|---|
| $H_5$: Validation depends on the **technical complexity** of a system. | Best MODEL: 2-variables; $R^2$=42% $n = 26$ ACCEPTED |

The best model contains two variables: T7A_CAT, the variable for amount of printed output, and T7B_CAT, representing the amount of screen output. Together they explain 42% of the variation. The hypothesis does not produce any other models with more variables that are significant.

The formula for the model described above is as follows:

$$VAL\% = 79.7 + 6.0 * T7A\_CAT - 5.4 * T7B\_CAT \quad (3)$$

The conclusion from testing this hypothesis is that reported levels of validation are positively associated with more printed output and less screen output.

**Example**($H_{5val}$): Observation #22, a system to manage data records for a longitudinal educational research study (reported validation=70%) and observation #20, a runoff analysis and risk assessment simulation modeling system for alpine watersheds (reported validation=80%) are both predicted with an error ¡ 1% by the one model resulting from this hypothesis. (Say what predictions are HERE.) One-third of the observations are predicted quite well by this model. The largest error is 50%, for an observation with a very low reported validation level of 20%.

| | |
|---|---|
| $H_6$: Verification depends on the level of **human involvement** in a system. | Best MODEL: 3-variables;$R^2$=61% 4-variables;$R^2$=67% $n = 36$ ACCEPTED |

The best model contains 3 variables that represent the number of developers involved in the design phase (H2D_DES) and the programming phase (H3D_PRG), as well as the total number of people involved in the system (H8_TOTAL). This model's $R^2$ value is 61%. Interestingly, H8_TOTAL shows up in a 1-variable model for a slightly larger sample size ($n$=38) explaining almost one-third (32%) of the variation all by itself. The model is slightly improved by the addition of H9_EXPRT, which indicates whether the person answering the questionnaire is an expert on the PSS. H8_TOTAL and H9_EXPRT both appear in several models resulting from testing hypotheses in this study.

The formulas for the models described above are as follows:

$$VER\% = 89.5 + 6.2 * H2D\_DES - 3.2 * H3D\_PRG - 0.15 * H8\_TOTAL \quad (4)$$
$$VER\% = 96.9 + 7.6 * H2D\_DES - 3.9 * H3D\_PRG - 0.17 * H8\_TOTAL - 11.1 * H9\_EXPRT \quad (5)$$

The conclusion from testing this hypothesis is that reported levels of verification are positively associated with having more developers (programmers) take part in the design phase, fewer developers (programmers) doing the actual programming, and fewer people involved in the system overall. The negative coefficient for H9_EXPRT indicates that verification was reported to be lower when the expert provided the information on the survey. This probably means that the expert is more aware of the limitation of the system than users, for whom the system may work quite well, or programmers, who will only admit to doing a great job.

**Example**($H_{6ver}$): Observation #14, an interactive environmental information and decision support system, is a good example of improvement in prediction as variables are added to test an hypothesis. The reported verification for this system is 95%. The above 3-variable model predicts verification of 87.6%, and the 4-variable model improves the prediction to an essentially perfect 94.8%. The verification percentage for observation #4, a forest stand growth simulation model with verification reported as 95%, is predicted by the 3-variable model to be 95.2%, and 93.2% in the 4-variable model. Both predictions are excellent. Both models predict verification for over half of the observations within 10%, and predict all but one of the rest within 20%.

| | |
|---|---|
| $H_6$: Validation depends on the level of **human involvement** in a system. | Best MODEL:<br>2-variables;$R^2$=32%<br>3-variables;$R^2$=43%<br>4-variables;$R^2$=53%<br>5-variables;$R^2$=61%<br>$n = 32$<br>ACCEPTED |

This hypothesis yielded a wealth of results. The best 2-variable model contains variables for the number of developers involved in designing the system (H1D_DEFN) and in system implementation (H5D_IMPL). The best 3-variable model adds H4U_TEST, the number of users involved in the testing phase. The best 4-variable model adds H9_EXPRT, indicating whether the person answering the questionnaire is an expert involved with the system. The best 5-variable model takes out H1D_DEFN, and adds two variables regarding users of the system: H2U_DES, the number of users involved in the design phase, and H3U_PRG, the number of users involved in the programming phase. The models explain between 32% and 61% of the variation in the system.

The formulas for the models described above are as follows:

$$VAL\% = 70.3 + 9.7 * H1D\_DEFN - 6.8 * H5D\_IMPL \quad (6)$$

$$VAL\% = 77.2 + 9.9 * H1D\_DEFN - 7.5 * H5D\_IMPL - 1.9 * H4U\_TEST \quad (7)$$

$$VAL\% = 90.4 + 9.7 * H1D\_DEFN - 7.9 * H5D\_IMPL - 2.1 * H4U\_TEST - 17.9 * H9\_EXPRT \quad (8)$$

$$\begin{aligned} VAL\% = {} & 102.5 + 14.2 * H2U\_DES - 7.8 * H5D\_IMPL - 3.1 * H4U\_TEST - 16.0 * H9\_EXPRT \\ & -10.2 * H3U\_PRG \quad (9) \end{aligned}$$

The conclusion from testing this hypothesis is that reported levels of validation are positively associated with having more developers involved in the problem definition phase, fewer developers involved in the implementation phase, and fewer users involved in the testing phase. In the last equation, there is strong positive influence from more users being involved in the design phase. As in the previous hypothesis, obtaining the information about the system from an expert is likely to have a negative effect on the reported level of validation.

**Example**($H_{6val}$): Observation #7, an economic and market model of the Austrian forest sector, is predicted well by all four models resulting from this hypothesis. The models predict between 73.2% and 79.6% validation; reported validation for this system is 80%. Interestingly, these models predict quite well for systems with very low reported levels of validation (observation #13, validation=0% and observation #35, validation=5%) but are poor predictors for some—but not all—systems with high validation levels. Predictions for observations with high validation improve dramatically as variables are added to the model.

| | |
|---|---|
| $H_9$: Verification depends on the degree to which a system is **observable.** | Best MODEL: *–none–* $n = 40$ REJECTED |

The hypothesis failed to produce a model in which any variable is significant. Every model examined (from one to five variables) contained insignificant variables. Also, each model's $R^2$ was very low. For example, the best 1-variable model explained only 6% of the variability in the data, and that one variable was not significant at the 0.05 probability level. Thus, the hypothesis is rejected.

| | |
|---|---|
| $H_9$: Validation depends on the degree to which a system is **observable.** | Best MODEL: 1-variable; $R^2$=13% $n = 40$ REJECTED |

The only model accepted was the 1-variable model with O5_STAFF, a variable that has the value '1' if lack of trained staff people is a limitation on how often a system is run, and '0' if this is not a problem. The $R^2$ here is only 13%.

This hypothesis is a good example of not choosing a model that has significant variables and a higher $R^2$ because the model cannot pass the common sense criterion. The best 2-variable model contained O5_STAFF (above) and O5_COST. Both variables represent limitations on running a PSS; a '0' value for O5_COST indicates that cost is not a limitation to repeated observations, and a '0' value for O5_STAFF indicates that staff to run the model are available. If either of these (cost or staff) *is* a limitation, the value of the variable is '1'. The coefficients of the model variables are roughly equivalent but with signs in opposite directions; the coefficient for O5_COST is positive (+39), and coefficient for O5_STAFF is negative (-34). It is clear that if O5_STAFF is '1', the effect on the reported level of validation is in the right direction (i.e., negative). But if the *COST* of running a system is seen as a limitation, it doesn't make sense that it should have a positive effect on validation; high costs are likely to result in running a system less often, and thus there would be less opportunity to establish confidence in the system. Thus the 2-variable model, with $R^2$=24%, is not accepted as a meaningful model.

The formula for the one significant model described above is as follows:

$$VAL\% \quad = \quad 79.8 - 24.0 * O5\_STAFF$$

Although this equation in statistically significant, it explains so little variation (%13) that it is simply not useful. For that reason, hypothesis $H_9$ for validation is rejected. A maximum level of $R^2$=25% was required for an equation to be accepted in addition to statistical significance. Three other hypotheses, below ($H_{8a}$ for both verification and validation, and $H_{8b}$ for validation), are also rejected for this reason.

| | | Best MODEL: |
|---|---|---|
| | | 1-variable; $R^2$=52% |
| | | 3-variable; $R^2$=67% |
| | | 4-variable; $R^2$=72% |
| $H_{10}$: | Verification depends on the combination of **complexity, human involvement & observability** of a system. | $n = 35$ |
| | | ACCEPTED |

The best 1-variable model resulting from testing this hypothesis is exceptionally strong. The variable H8_TOTAL, the total number of people involved in the system, explains 52% of the variation. A 2-variable model combines H8_TOTAL with O1_CAT, representing the frequency with which modeled event occurs, but the significance level of O1_CAT is just over .05, so the model is not accepted. The best 3-variable model combines H8_TOTAL with variables representing the number of developers involved in the design phase (H3D_DES) and the programming phase (H3D_PRG). The addition of H9_EXPRT, indicating whether the respondent to the questionnaire is an expert, creates the best 4-variable model and the model with the highest $R^2$. 72%. Note that the only significant variables in this model are for Human Involvement, even though this hypothesis includes variables for Technical Complexity and Observability as well.

The formulas for the models described above are as follows:

$$VER\% = 92.9 - 0.15 * H8\_TOTAL \tag{10}$$

$$VER\% = 90.6 - 0.15 * H8\_TOTAL + 6.6 * H2D\_DES - 3.4 * H3D\_PRG \tag{11}$$

$$VER\% = 97.1 - 0.17 * H8\_TOTAL + 7.7 * H2D\_DES - 4.1 * H3D\_PRG - 9.9 * H9\_EXPRT \tag{12}$$

The conclusion from testing this hypothesis is exactly the same as that found above in $H_6$ for verification: reported levels of verification are positively associated with more developers (programmers) in the design phase, fewer developers (programmers) doing the actual programming, and fewer people involved in the system overall. Again, H9_EXPRT is an influential negative factor on reported verification.

**Example**($H_{10_{ver}}$): Observation #23, a data input verification program used in statistical analyses, is one of many examples of very good verification predictions by the three significant equations resulting from this hypothesis. Each equation predicts over one-half of the observations with an error less than 10%. Even the very unusual observation #39, with 510 people (variable

H8_TOTAL) and 0% reported verification, is predicted well; the 3 models predict 8.9%, 13.5% and 16.4% — all within a respectable range of prediction.

| | |
|---|---|
| $H_{10}$: Validation depends on the combination of **complexity, human involvement & observability** of a system. | Best MODEL: 2-variable; $R^2$=38% 3-variable; $R^2$=56% 4-variable; $R^2$=62% 5-variable; $R^2$=69% $n = 35$ ACCEPTED |

The best 2-variable model found contains H2D_DES, indicating that developers are involved in designing the system, and T1_CAT, which represents the number of lines of program code. The best 3-variable model combines T3_CAT, representing the number of modules in the system, with the above two variables. The addition of H9_EXPRT results in the best 4-variable model. And the addition of H3D_PRG (number of developers in the design phase) creates the best 5-variable model, which explains 69% of the variation in the data.

The formulas for the models described above are as follows:

$$VAL\% = 110.8 + 11.9 H2D\_DES - 15.7 * T1\_CAT \quad (13)$$

$$VAL\% = 107.0 + 13.7 * H2D\_DES - 12.4 * T1\_CAT - 6.2 * T3\_CAT \quad (14)$$

$$VAL\% = 115.3 + 13.7 * H2D\_DES - 12.5 * T1\_CAT - 6.2 * T3\_CAT - 12.2 * H9\_EXPRT \quad (15)$$

$$\begin{aligned} VAL\% = {} & 108.5 + 16.8 * H2D\_DES - 8.9 * T1\_CAT - 5.5 * T3\_CAT - 16.2 * H9\_EXPRT \\ & -3.7 * H3D\_PRG \quad (16) \end{aligned}$$

Testing this hypothesis gives advice about programmers and system size. Reported levels of validation are positively associated with more developers (programmers) in the design phase and fewer developers (programmers) doing the actual programming. Smaller programs (less lines of code and fewer modules) also influence higher levels of reported validation. Again, H9_EXPRT is an influential negative factor.

**Example**($H_{10val}$): Observation #16, an environmental assessment and reporting system used in Thailand and Kenya, has a reported verification of 75%. The predictions of the four models in this hypothesis range from 86.5% (an error of +11.5%) to 71.80% (an error of -3.2%), including the prediction of the 5-variable model, a nearly perfect 75.1%. Validation levels for one-half of the observations were predicted within 10% by all four models; the other half were predicted within 30%.

| | |
|---|---|
| $H_7$: Verification depends on the **methods used to test** a system. | Best MODEL:<br>2-variable; $R^2$=44%<br>3-variable; $R^2$=53%<br>4-variable; $R^2$=65%<br>5-variable; $R^2$=72%<br>$n = 34$<br>ACCEPTED |

The variable VT11_USR has the value 1 if "benchmarking" (client's formal tests for acceptance) was used, and a value of 0 if not. It appears in every 2-, 3-, 4- and 5-variable model resulting from testing this hypothesis. It is the variable that is most consistently evident in the results of not only testing this hypothesis but also $H_{8a}$, $H_{11}$, $H_{14}$ and $H_4$ regarding verification. It also shows up in the result of one hypothesis regarding validation, $H_4$.

VT11_USR is combined with VH3_DES, testing by comparing the system to expert sources during the design phase, to produce the best 2- variable model. The best 3-variable model adds the variable VT1_USR, indicating that the program compiles without crashing in the user phase. The best 4-variable model then adds the variable VS9_DES, indicating that the design phase determined that the program can be continually improved. The last addition to create the best 5-variable model is VS8_DES, representing that the system was improved by comparing its results to real events. The $R^2$ of this last model is quite high: 72%.

The formulas for the models described above are as follows:

$$VER\% = 81.5 + 15.7 * VT11\_USR - 26.8 * VH3\_DES \quad (17)$$

$$VER\% = 89.1 + 18.1 * VT11\_USR - 32.8 * VH3\_DES - 12.9 * VT1\_USR \quad (18)$$

$$VER\% = 95.3 + 21.8 * VT11\_USR - 29.9 * VH3\_DES - 18.5 * VT1\_USR - 15.7 * VS9\_DES \quad (19)$$

$$VER\% = 97.0 + 21.1 * VT11\_USR - 36.8 * VH3\_DES - 19.5 * VT1\_USR - 20.6 * VS9\_DES + 21.6 * VS8\_DES \quad (20)$$

The conclusion from testing this hypothesis is that certain methods of testing, at certain phases in the development of the system, influence reported levels of verification. The most influential positive type of testing is VT11_USR, i.e., having the system pass formal tests that the client requires for the system to be accepted. The variable indicates that this testing, also known as benchmarking, is performed during the initial user phase. VT11_USR appears in the results of several of the following hypotheses. Testing by comparing the system to expert sources during the design phase is not recommended; this type of testing is best done in the programming/debugging stage.

Determining that the system under development can be improved (once) as a result of testing during the design phase also is positive. But determining during the design phase that the system can be *continually* improved and refined is negative, possibly because either (1) the idea is overwhelming or (2) most designers just don't report it happening.[8] Another puzzle is why

[8]Notice that the coefficients for VS8_DES and VS9_DES offset each other. Perhaps respondents are most likely to answer 'yes' to both VS8_DES and VS9_DES, or 'no' to both, depending on their style and attitude regarding the nature of system design.

VT1_USR is negative. This variable represents the fact that a program compiles and runs without crashing; one would think it is an obvious first test, particularly from the users' point of view. The fact that it is negative here probably means that respondents did not consider it a test during the user phase. The systems in this study are all working systems; systems that crash would never get into users' hands, and thus would not be represented by this data.

**Example**($H_{7ver}$): Observation #29, a program that determines optimal management of forest stands, is predicted very well by all four of the above models for this hypothesis. Its reported verification level is 98%. Both the 4- and 5-variable equations predict 98.6%, the 2-variable prediction is 97.2%, and the 3-variable prediction is 94.3%. The 5-variable model predicts verification for all observations within 21% or less; the 4- variable predictions are all within 31%.

| | Best MODEL: |
|---|---|
| | 2-variable; $R^2$=40% |
| | 3-variable; $R^2$=55% |
| | 4-variable; $R^2$=69% |
| | 5-variable; $R^2$=76% |
| | $n = 34$ |
| $H_7$: Validation depends on the **methods used to test** a system. | ACCEPTED |

The best 2-variable model contains VS4_PRG, testing a system with historical data during the programming phase, and VH6_USR, client acceptance and/or purchase of the system during the user phase. The best 3-variable model adds VT1_USR, compiling correctly in the user phase. The best 4-variable model adds VS3_PRG, a determination that variables are complete, unbiased and accurate in the programming phase. The best

5-variable model adds VH4_DES, indicating that users participated at some level during the design phase. These models explain 40% to 76% of the variability in the system.

Note that this hypothesis — that all types of testing methods can be used to predict reported levels of validation — results in an equation that includes one Technical validation method (VT1_USR), two Semi-Technical methods (VS4_PRG, VS3_PRG) and two Human Judgment methods (VH6_USR and VH4_DES).

The formulas for the models described above are as follows:

$$VAL\% = 43.2 + 26.7 * VS4\_PRG + 24.1 * VH6\_USR \quad (21)$$

$$VAL\% = 50.7 + 26.6 * VS4\_PRG + 33.8 * VH6\_USR - 22.1 * VT1\_USR \quad (22)$$

$$VAL\% = 56.2 + 28.2 * VS4\_PRG + 37.5 * VH6\_USR - 26.6 * VT1\_USR - 21.2 * VS3\_PRG \quad (23)$$

$$VAL\% = 51.7 + 23.3 * VS4\_PRG + 37.7 * VH6\_USR - 20.7 * VT1\_USR - 24.9 * VS3\_PRG + 17.4 * VH4\_DES \quad (24)$$

The conclusion from testing this hypothesis is that certain methods of testing, at certain phases in the development of the system, influence reported levels of validation. There are three recommended methods: (1) VS4_PRG, running the system with historical data and comparing the system's output with actual data from the past, during programming; (2) VH6_USR, deciding that a system is valid because the client likes, accepts or pays for it at the user stage; and (3) having the people who will use the system participate in its design.

It is not recommended that VS3_PRG, determining that variables are complete, unbiased and accurate, be done during programming. This should be done in the design phase. Also VT1_USR appears again (again with a negative value); this is discussed above in $H_7$ for verification.

**Example**($H_{7val}$): The validation level of 95% reported for observation #30, electronic design automation software for the design of electrical wire harnesses, is predicted very well. The 4-variable equation gives the best prediction, 95.2%, and the 3-variable equation makes the lowest prediction, 89%. The 5- and 2-variable models also come very close, with predictions of 92% and 94% respectively. All four models for this hypothesis predict the reported validation levels of one-half or more of the observations within 10%.

| | Best MODEL: 1-variable; $R^2$=21% $n = 34$ |
|---|---|
| $H_{8a}$: Verification depends on the **technical methods** used to test a system. | REJECTED |

There is only one good model, a 1-variable model with VT11_USR. As noted before, this variable indicates that benchmarking was used as the test during the user phase, and was the basis on which the client accepted the system. VT11_USR is significant every time Technical testing methods are included in a hypothesis. The $R^2$ of 21% is not particularly high for an entire model compared to other models found in this study, but it is impressive for one variable, especially when there are 81 other variables in the hypothesis.

The formula for the model described above is as follows:

$$VER\% = 76.5 + 17.8 * VT11\_USR$$

The conclusion from testing this hypothesis is that one technical method of testing, VT11_USR, is quite important in influencing reported levels of verification. But because the $R^2$ of this equation is less than 25%, this hypothesis is rejected.

| | |
|---|---|
| $H_{8a}$: Validation depends on the **technical methods** used to test a system. | Best MODEL: 1-variable; $R^2$=19% *n = 34* REJECTED |

The only acceptable model is a 1-variable model with VT5_DES, indicating that the system was tested with representative inputs during the design phase. The $R^2$ for this model is 19%. No models with larger numbers of variables have all variables significant.

The formula for the model described above is as follows:

$$VAL\% = 83.2 - 24.1 * VT5\_DES$$

The conclusion from testing this hypothesis is that one technical method of testing, tracing representative inputs, their interactions and results (VT5_DES), cannot be accomplished at the design phase. However, the $R^2$ of this equation is less than 25%; therefore this hypothesis is rejected.

| | |
|---|---|
| $H_{8b}$: Verification depends on the **semi-technical methods** used to test a system. | Best MODEL: *–none–* *n = 34* REJECTED |

No models containing significant variables can be found. The hypothesis is rejected regarding verification.

| | |
|---|---|
| $H_{8b}$: Validation depends on the **semi-technical methods** used to test a system. | Best MODEL: 1-variable; $R^2$=19% *n = 34* REJECTED |

The only acceptable model is a 1-variable model containing VS4_PRG, which indicates that the system was tested with historical data during the programming phase. No larger models were found with significant variables.

The formula for the model described above is as follows:

$$VAL\% = 61.3 + 22.8 * VS4\_PRG$$

The conclusion from testing this hypothesis is that one semi-technical method of testing, running the PSS with past data, influences reported levels of validation. This variable also occurred in the hypothesis results for $H_7$ for validation. Because the $R^2$ of this equation is less than 25%, this hypothesis is rejected.

| | Best MODEL: 1-variable; $R^2$=28% 4-variable; $R^2$=49% $n = 34$ |
|---|---|
| $H_{8c}$: Verification depends on the **human judgment methods** used to test a system. | ACCEPTED |

The best 1-variable model contains VH3_DES, indicating that the system was compared to expert sources during the design phase, and explains 28% of the variability in the data. There is no 2- or 3-variable model that contains all significant variables. VH3_DES is included in a best 4-variable model, which also includes the variables for favorable first impressions by users at the programming phase (VH1_PRG), users participating in the system in some manner during the design phase (VH4_DES), and payment or other form of client acceptance during the user phase (VH7_PRG). This best 4-variable model explains one-half of the variation in the data.

The formulas for the models described above are as follows:

$$VER\% = 90.2 - 29.2 * VH3\_DES \qquad (25)$$

$$VER\% = 83.4 - 34.9 * VH3\_DES + 15.9 * VH1\_PRG + 13.6 * VH4\_USR - 12.0 * VH7\_USR \qquad (26)$$

The conclusion from testing this hypothesis is that several human judgment methods of testing influence reported levels of verification. Verification is higher when users form positive first impressions during the programming phase (VH1_PRG). This can occur with demonstrations, asking users their opinions, and generally good public relations efforts by the development staff. This is closely associated with VH4_USR, which indicates that users have been requested to participate throughout the entire development process. Recall that VH4_DES was a positive variable in the results of $H_7$ for validation; it is also in the results of $H_4$ and $H_{8c}$ for validation, below.

Negative coefficients indicate that it is not helpful to try compare the results of the PSS to expert sources during the design phase. It is also appears negative for verification that users would pay for the PSS, or the system, generates a positive cash flow (VH7_USR). This is the only equation in which this variable occurs; it may be a reflection of the considerable number of research and government systems included in this study and also the fact that user appreciation and acceptance is more likely a validation, rather than verification, of a PSS.

**Example**($H_{8c-ver}$): Observation #21, an information and referral database system that helps customers of a non-profit service agency, is one of four examples whose verification levels are predicted with 5% by both equations resulting from the above hypothesis. The reported level of verification for observation #21 is 90%; the 1-variable model prediction is

90.2% and the 4-variable prediction is 87.3%.

| | Best MODEL: 2-variable; $R^2$=34% 3-variable; $R^2$=45% $n = 34$ |
|---|---|
| $H_{8c}$: Validation depends on the **human judgment methods** used to test a system. | ACCEPTED |

The best 2-variable model contains VH4_DES, indicating that users participated in some way during the design phase, and VH6_USR, indicating that the client likes or accepts or pays for the system during the user phase. The best 3-variable model adds to these two the variable VH5_DES, which says that the process of modeling clarified the understanding of the real problem during the design phase. These models explain 34% and 45% of the variability in the data respectively.

The formulas for the models described above are as follows:

$$VAL\% = 52.8 + 25.1 * VH4\_DES + 24.6 * VH6\_USR \quad (27)$$

$$VAL\% = 55.9 + 25.9 * VH4\_DES + 25.1 * VH6\_USR - 20.8 * VH5\_DES \quad (28)$$

The conclusion from testing this hypothesis is that three human judgment methods of testing influence reported levels of validation. VH4_DES, getting users involved in various stages of the system development (here, in the design phase) has been mentioned several times in other models above. VH6_USR, the client accepting/purchasing the system at the user phase, has also been mentioned above. VH5_DES means that the modeling process clarified details, yielding a better understanding of the real problem; even though this seems to be a very good idea, its negative value may indicate that, in practice, it just doesn't happen.

**Example**($H_{8c-val}$): Observation #5, an environmental impact assessment program with validation reported as 80%, is predicted to be 77.9% by the 2-variable model above, and to be 81.8% by the 3-variable model above. The reported validation levels of one-third of the observations are predicted within 10% by these two models.

| | |
|---|---|
| $H_{11}$: Verification depends on groupings of **test methods** used in systems. | Best MODEL: 2-variable; $R^2$=35% 5-variable; $R^2$=55% $n$ = *34* ACCEPTED |

This hypothesis states that the testing methods themselves, without regard to when they occur (design phase, programming phase, initial user phase) can predict levels of verification. There are 27 testing methods, and thus 27 groups.

The best 2-variable model includes VT11_GRP, indicating testing by benchmarking at any phase. (This is probably because VT11_USR is so strong, as mentioned above under $H_7$.) Also included in the best 2-variable model is VH3_GRP, testing by comparing the system to expert sources. This equation explains one-third of the variability in the data ($R^2$=35%). There is no 3- or 4-variable model containing all significant variables. The best 5-variable model contains the above two variables, plus these three: VT1_GRP, compiling at the user phase; VS8_GRP, improving the PSS by comparing it to results of real events; and VS6_GRP, using the PSS to predict future events and judging the final results when they happen. The $R^2$ of this 5-variable model is 55%.

Note that this hypothesis — that all groups of testing methods predicts verification — includes two Technical groups (VT11_USR, VT1_GRP), two Semi-Technical (VS8_GRP, VS6_GRP) and one Human Judgment (VH3_GRP) in the best 5-variable model.

The formulas for the models described above are as follows:

$$VER\% = 85.4 + 10.0 * VT11\_GRP - 9.1 * VH3\_GRP \quad (29)$$

$$VER\% = 104.6 + 12.2 * VT11\_GRP - 11.2 * VH3\_GRP - 7.8 * VT1\_GRP + 7.7 * VS6\_GRP - 6.7 * VS8\_GRP \quad (30)$$

The conclusion from testing this hypothesis is that groups of test methods do have an influence on reported levels of verification. The positive influence by VT11_GRP is no doubt due to the strength of VT11_USR, discussed above. VS6_GRP, testing a PSS by predicting future events and waiting for them to occur, has not occurred in another "best model"; therefore its positive influence here must indicate this test method can be used different stages.

The negative effect of VT1_GRP comes from VT1_USR, explained above. VS8_DES is the only member of VS8_GRP, and is discussed in $H_{11}$ for verification. VH3_GRP contains VH3_DES which appears in $H_{8c}$ and $H_7$, also regarding verification.

**Example**($H_{11_{ver}}$): Observation #32, an office management system with a verification level reported of 95%, is the best example of both the 2-variable and 5-variable models resulting from this hypothesis. The verification predicted by the 2-variable model is 95.4% and the 5-variable model predicts 95.6%. Both models predict verification for over one-half of the observations within 10% or less.

| | |
|---|---|
| $H_{11}$: Validation depends on groupings of **test methods** used in systems. | Best MODEL:<br>2-variable; $R^2$=32%<br>3-variable; $R^2$=40%<br>4-variable; $R^2$=51%<br>$n = 34$<br><br>ACCEPTED |

The best 2-variable model contains VS4_GRP, testing the PSS with historical data, and VT5_GRP, testing representative inputs and relationships. VT8_GRP, comparing the PSS with other existing methods of solving the problem, is added to form the best 3-variable model. The best 4- variable model adds VH1_GRP, favorable first impressions by users. These four groups of testing methods explain half of the variation in the data.

Note that the best model from this hypothesis includes two Technical groups (VT5_USR, VT8_GRP), one Semi-Technical (VS4_GRP) and one Human Judgment (VH1_GRP) variable. Also note that an entirely different set of groups appears in the best models resulting from applying this hypothesis ($H_{11}$, above) to verification.

The formulas for the models described above are as follows:

$$VAL\% = 82.4 + 10.3 * VS4\_GRP - 12.1 * VT5\_GRP \quad (31)$$

$$VAL\% = 85.7 + 14.1 * VS4\_GRP - 12.3 * VT5\_GRP - 8.4 * VT8\_GRP \quad (32)$$

$$VAL\% = 72.0 + 16.2 * VS4\_GRP - 12.8 * VT5\_GRP - 10.9 * VT8\_GRP + 11.8 * VH1\_GRP \quad (33)$$

The conclusion from testing this hypothesis is that groups of test methods do have an influence on reported levels of validation. The positive ones are VS4_GRP and VH1_GRP. VS4_PRG is discussed above in $H_7$ and $H_{8b}$, both for validation. VH1_PRG is discussed above in $H_{8c}$ for validation.

**Example**($H_{11val}$): Observation #8, a simulation program for teaching forestry management, is one of four examples whose validation values are predicted within 7.5% by all three equations resulting from hypothesis $H_{11}$. The reported validation level for this observation is 80%; it is predicted to be

78.8%, 80.9% and 79.7% by the 2-, 3- and 4- variable models above.

| | | Best MODEL: 1-variable ;$R^2$=34% $n$ = *33* ACCEPTED |
|---|---|---|
| $H_{12}$: | Verification depends on summary scores (totals) for **technical complexity, human involvement, observability** & the **test methods** used in a system. | |

This hypothesis includes four variables. They are the summary scores calculated for Technical Complexity, (TEC_NUM), Observability (O6_TOTAL) and Test Methods (VAL_NUM). The summary score for Human Involvement, (H8_TOTAL) is the total number of people involved in the system, as reported by respondents to the questionnaire.

A model was found with one significant variable, H8_TOTAL ($R^2$=34%). Initial runs of subsets of the final set of observations produced no model with any significant variables at all; the appearance of H8_TOTAL from a larger sample size could be due to the fact that later observations had very high numbers of people involved, thus expanding the sample to cover a broader range of conditions.

The formula for the model described above as follows:

$$VER\% \quad = \quad 92.0 + -0.11 * H8\_TOTAL \tag{34}$$

The conclusion from testing this hypothesis has also been found in other hypotheses: larger numbers of people involved in the system lessen the extent to which people feel their systems have been verified. Note that the coefficient for H8_TOTAL, while negative, is very small. This means that this effect is greater with very large numbers of people than in systems with few people.

**Example**($H_{12ver}$): Observation #27, a shipping and inventory system used in manufacturing, is the best example of the one equation that results from the above hypothesis. The reported verification of this system is 90%; this 1-variable model predicts the verification to be 89.03%. Hypothesis $H_{12}$ is based on four scores that attempt to characterize the observations in a very general way. Only one score, H8_TOTAL (number of people involved in a system) was significant. It is interesting to note that, using this one score, an equation was found that predicts the verification levels of 56% of the observations within 10% of actual values, and another 38% between 10% and 30% of actual. This accounts for all but the following three observations: The two observations with the highest H8_TOTAL values (obs.#39 with 510 people and 0% reported verification, and obs.#37 with 300 people and 95% verification) are each predicted with a difference of 36% to actual. The remaining observation, obs.#13 (15 people and 50% verification) is predicted with a difference of 40%. This is a very robust result for a simple equation with one variable that takes on values from 2 to 510.

| | | |
|---|---|---|
| $H_{12}$: | Validation depends on summary scores (totals) for **technical complexity, human involvement, observability** & the **test methods** used in a system. | Best MODEL: *–none–* $n = 33$ REJECTED |

Not one significant equation was found. The four summary variables do not predict validation. The hypothesis is rejected.

| | | |
|---|---|---|
| $H_{14}$: | Verification depends on the **complexity, human involvement & observability** factors of a system combined with the *groups* of **test methods** used. | Best MODEL: 2-variable; $R^2$=66% 3-variable; $R^2$=71% 4-variable; $R^2$=79% 5-variable; $R^2$=85% $n = 30$ ACCEPTED |

The best 2-variable result contains VT11_GRP (benchmarking), mentioned before in $H_{11}$ for verification, and H8_TOTAL (total people involved), also significant in several other models. The 3-variable model is formed with the addition of O1_CAT, the variable representing how frequently the real event or problem being modeled occurs in the real world. The 4-variable model removes O1_CAT and adds H3D_PRG, the developer being involved in the programming stage and H2D_DES, the developer being involved in the design stage. H3D_PRG has appeared in three previous models, and H2D_DES has appeared previously in two models. The 5-variable model adds H9_EXPRT, which has appeared in three models above. The high $R^2$ values, from 66% to 85%, indicate that this is a rich and fruitful hypothesis.

The formulas for the models described above are as follows:

$$VER\% = 85.2 - 0.14 * H8\_TOTAL + 7.7 * VT11\_GRP \quad (35)$$

$$VER\% = 79.3 - 0.14 * H8\_TOTAL + 7.4 * VT11\_GRP + 2.5 * O1\_CAT \quad (36)$$

$$VER\% = 82.7 - 0.15 * H8\_TOTAL + 7.2 * VT11\_GRP + 6.5 * H2D\_DES - 3.2 * H3D\_PGMG \quad (37)$$

$$VER\% = 89.6 - 0.17 * H8\_TOTAL + 7.1 * VT11\_GRP + 8.3 * H2D\_DES - 4.1 * H3D\_PGMG - 11.3 * H9\_EXPRT \quad (38)$$

All of the variables in these formulas have been discussed above.

**Example**($H_{14_{ver}}$): Verification for observation #14, another interactive environmental information and decision support system, is predicted very well by all four models resulting from the hypothesis above. The reported verification for this system is 95%; the predicted verification levels range from 87.8% (a difference of 7.2%, the largest difference) predicted by the 3-variable model to a near-perfect 95.2% predicted by the 5-variable model. All four models predict at least 22 of 36 observations used in this hypothesis within

10% of actual values.

| | | Best MODEL: 2-variable; $R^2$=45% 3-variable; $R^2$=55% 4-variable; $R^2$=65% $n = 30$ |
|---|---|---|
| $H_{14}$: | Validation depends on the **complexity, human involvement & observability** factors of a system combined with the *groups* of **test methods** used. | ACCEPTED |

This hypothesis, which combines system features with *groups* of test methods, has no variables representing groups in its results. The best 2-variable model contains T3_CAT, representing the number of modules in a system, and H2D_DES, the variable indicating that developers are involved in the design phase. The variable T1_CAT is added, representing the number of lines of program code, to form the best significant 3-variable model. The best 4-variable model drops T1_CAT and adds H3D_PRG, the number of developers involved in programming, and H9_EXPRT, indicating if it was an expert who answered the questionnaire. Variables H2D_DES, H3D_PRG and H9_EXPRT are referred to several times above for other hypotheses. The $R^2$ values of these formulas explain from almost one-half to almost two- thirds of the variability in the data.

The formulas for the models described above are as follows:

$$VAL\% = 80.4 + 7.6 * H2D\_DES - 8.4 * T3\_CAT \quad (39)$$

$$VAL\% = 101.0 + 11.9 * H2D\_DES - 7.1 * T3\_CAT - 8.8 * T1\_CAT \quad (40)$$

$$VAL\% = 89.4 + 13.7 * H2D\_DES - 6.8 * T3\_CAT - 4.5 * H3D\_PGMG - 17.7 * H9\_EXPRT \quad (41)$$

All of the variables in these formulas have been discussed above.

**Example**($H_{14val}$): Observation #19, a system that coordinates data about consumers of services for a non-profit organization, has a reported validation of 85%. Its predicted validation levels are 87.2%, 91.3% and 83.3%

given by the 2-, 3- and 4-variable models, respectively. Observation #27, a product allocation and shipping program for manufacturing, is also interesting. Its reported validation is 80%, and its predicted validation is a very close 79.4% in both the 2- and 3-variable models. The prediction of the 5-variable model is not quite as good (74.1%), but the error is only 6%. Over one-third of the validation levels for the 35 observations in the data set for this hypothesis are predicted within 10% by these models.

| | |
|---|---|
| $H_4$: Verification depends on the **complexity, human involvement & observability** factors of a system combined with the *individual* **test methods** used. | Best MODEL:<br>2-variable; $R^2$=72%<br>3-variable; $R^2$=78%<br>4-variable; $R^2$=84%<br>5-variable; $R^2$=86%<br>*n = 30*<br>ACCEPTED |

The best 2-variable result contains VT11_USR, mentioned before as appearing many times (in five models above), and H8_TOTAL, also mentioned before as showing up several times (in four models above). This model explains almost three-fourths of the variability with only two variables. The best 3- variable model is formed by adding H9_DEV, indicating that the person answering the questionnaire is a system developer. The 4-variable model adds O1_CAT, representing the frequency with which the real event or problem occurs in the real world. The best 5-variable model is formed by adding VH3_PRG, testing the PSS by comparing it to expert sources during the programming phase. Notice that the $R^2$ values for these four models are quite similar – just a bit larger – the $R^2$ values for verification hypothesis $H_{14}$.

The formulas for the models described above are as follows:

$$VER\% = 82.6 + 17.6 * VT11\_USR - 0.14 * H8\_TOTAL \qquad (42)$$

$$VER\% = 74.1 + 20.2 * VT11\_USR - 0.14 * H8\_TOTAL + 10.6 * H9\_DEV \tag{43}$$

$$VER\% = 66.2 + 19.6 * VT11\_USR - 0.15 * H8\_TOTAL + 12.8 * H9\_DEV + 2.8 * O1\_CAT \tag{44}$$

$$VER\% = 67.8 + 19.6 * VT11\_USR - 0.15 * H8\_TOTAL + 14.1 * H9\_DEV + 2.5 * O1\_CAT - 8.0 * VH3\_PRG \tag{45}$$

A variable appearing for the first time in results from testing hypotheses is H9_DEV, the fact that the person answering the questionnaire is a system developer. The value of this term in the equation is positive, indicating that system developers (programmers) report higher levels of trust in their systems. VH3_PRG also has not shown up before; the negative coefficient indicates that comparing a PSS to expert sources is not done in the programming phase. (Recall the same recommendation for this type of testing in the design phase, VH3_DES, several times.) The other three variables (VT11_USR, H8_TOTAL, and O1_CAT) have been discussed above.

**Example**($H_{4ver}$): Observation #33, a patient administration and scheduling system for a county medical clinic, has a reported verification level of 98%. The four equations above predict the verification for this system in the range from 104% (by the 5-variable model) to a perfect 97.90% (by the 3-variable model). Of course a prediction exceeding 100% would be set at the maximum, 100verification level, and a moderate number of people involved in the system (50). Interestingly, the only observation for which the equations consistently predict poorly is observation #37, which also has a high verification (95%) and a large number of people (300). The four models predict extremely well for the observation with the largest number of people (obs.#39, with 510 people involved) but a verification level of 0%. Of the 9 other observations with high verification levels (95%–100%), all but one are predicted well (i.e., 8 with a difference less than 13% and one with a difference of 26%).

| | Best MODEL: 2-variable; $R^2$=45% 3-variable; $R^2$=55% 4-variable; $R^2$=62% 5-variable; $R^2$=69% $n = 30$ |
|---|---|
| $H_4$: Validation depends on the **complexity, human involvement & observability** factors of a system combined with the *individual* **test methods** used. | ACCEPTED |

The best 2-variable and 3-variable models are exactly the same as those for the validation hypothesis $H_{14}$, above. The 4-variable model adds the important VT11_USR, benchmarking in the user phase. The 5-variable model removes VT11_USR and adds VH4_DES, user participation in the design phase, and VS3_PRG, determining that variables are complete, accurate and unbiased during the programming phase. Notice that the $R^2$ values for the first three formulas are virtually the same as those for validation hypothesis $H_{14}$, and that the first two formulas in each set of results are identical.

The formulas for the models described above are as follows:

$$VAL\% = 80.4 + 7.6 * H2D\_DES - 8.4 * T3\_CAT \quad (46)$$

$$VAL\% = 101.0 + 11.9 * H2D\_DES - 7.1 * T3\_CAT - 8.8 * T1\_CAT \quad (47)$$

$$VAL\% = 93.7 + 12.1 * H2D\_DES - 7.3 * T3\_CAT - 8.7 * T1\_CAT + 12.1 * VT11\_USR \quad (48)$$

$$VAL\% = 101.8 + 12.0 * H2D\_DES - 5.4 * T3\_CAT - 10.6 * T1\_CAT + 18.2 * VH4\_DES - 16.5 * VS3\_PRG \quad (49)$$

All of these variables have been previously discussed, and their behaviors explained. H2D_DES is the variable that appears most often in the above formulas. It appears six times; interestingly in three verification models (($H_6$, $H_{10}$ and $H_{14}$) and in three validation models ($H_{10}$, $H_{14}$ and here, $H_4$).[9] T1_CAT and T3_CAT both are in the results of $H_{10}$ and $H_{14}$ for validation. See $H_7$ (validation) for the discussion of VH4_DES and VS3_PRG.

[9]H2D_DES is usually paired with H3D_PRG. This is the only one of the six results in which H3D_PRG is not also a factor.

**Example**($H_{4val}$): Observation #20, a runoff analysis and risk assessment simulation model for alpine watersheds, is reported to be 80% validated; the four equations predict from 79.6% to 72.4% (a difference of only 7.6%). These four models predict the validation levels of two-thirds of the observations within 12% accuracy. Furthermore, the 4- and 5-variable models have a maximum error of 30%.

## 5.2 Summary of Hypotheses and Variables

The reader is referred to the following Appendices for a summary of the results presented in this Chapter:

- Appendix D: Names & Descriptions of All Variables used in All Hypotheses
- Appendix E: Listing of the data.
- Appendix F: Equations Resulting from Accepted Hypotheses
- Appendix G: All Variables Occurring in Equations Resulting from Accepted Hypotheses.

# CHAPTER 6: Discussion & Conclusion

## 6.1 Review

The study began with an examination of the Verification and Validation literature, focusing on the last forty years of developing methods to test complex computer-based Problem Solving Systems. A general hypothesis was developed stating that there are certain measurable features of systems that are related to the best testing methods for those systems. Combining the literature review with the general hypothesis led directly to a survey of systems currently operating in over a dozen countries. The data collection was structured specifically to explore relationships between test methods, system features, and the degrees to which people reported their systems to be valid. Analyzing the data with a series of smaller, more specific hypotheses found that most of these hypotheses are useful in explaining how certain system features and certain test methods are related to reported levels of confidence in a wide variety of systems. Because of the specific nature of the hypotheses, they correspond to linear equations that can be estimated and tested for significance using regression analysis. Interpretation of significant equations leads directly to useful recommendations regarding system features and types of validation methods that are significantly associated with the verification and validation of complex models. Statistical significance was judged with a standard $t$-test at the usual $\alpha = 0.05$ probability level. Resulting recommendations bear strongly on proper resource allocation for testing of new and existing computer-based Problem Solving Systems.

The strategy was to sort through the wealth of existing validation methods to determine which, if any, of them can be said to be significantly related to the levels of confidence reported by those who design, program, and use the systems. The goal was not to test yet another validation method on a particular system, but instead to show if and how a large number of validation methods are related to the stated validity of a large number of systems. Quantifiable features of Problem Solving Systems were also added to the analysis.

Forty systems were surveyed, with data collected on 165 variables for each system. Twelve hypotheses were proposed, and each one was tested separately regarding verification and validation levels as reported by respondents in the survey, for a total of 24 hypotheses tested. The results (Chapter 5) yield a list of 53 statistically significant equations that model many ways to test Problem Solving Systems, and interpretations of the practical uses of these models. See Appendix F for a concise list of these equations.

## 6.2 Discussion

Of these 24 hypotheses, three were rejected because no equations could be found that contained significant variables at the 0.05 significance level. An additional four equations have $R^2$ values below 25%. All of these are 1-variable models. These models, while significant, explain so little variation (less than 25%) in the observations that they are not useful. Since each is the only model resulting from testing their respective hypotheses, those four hypotheses were also rejected (viz., *H8a*-verification and *H8a*-validation; *H8b*-verification, and *H9*-validation).

This left a large group of hypotheses that were accepted (17), resulting in 49 models, and 37 significant variables of interest. The question now is how to use this information to give good advice regarding allocating resources and establishing fruitful testing environments for the development of computer-based Problem Solving Systems that people will trust. Attention should be paid both to the variables themselves and also to the special combination of variables, the equations. First we will discuss the variables, and then the models.

Just over one-third of the variables in the formulas have positive coefficients (15); finding reasonable explanations for these is quite a bit more straightforward than for those with negative coefficients. An unexpected but encouraging result is that the sign for a given variable remains constant across all equations in which it is significant. Thus, if a variable is positive in one formula it is positive everywhere it appears; if a variable is negative, it remains negative. Such consistency increases confidence that the variables are measuring an important feature of the validation of Problem Solving Systems. See Appendix D for a list of significant variables, the signs of their coefficients, and the hypothesis from which they emerged. The following is an interpretation of the most important variables (those that appear in more than one equation) with positive coefficients:

**Table 9a: Most Important Variables with Positive Coefficients**

| VARIABLE | INTERPRETATION |
|---|---|
| T7A_CAT | People trust systems with large output; perhaps they feel more confident that the system will give them exactly what they want. |
| H2D_DES | People are more confident in a system in which more developers are involved in designing it. |
| O1_CAT | People tend to trust systems more if the real-world event that the system models happens frequently; it is hard to observe rare events, & thus less possible to make a judgment on whether the system that models the event is correct. |
| VT11_USR (& VT11_GRP) | Systems are trusted when they meet predefined, formal standards set by the client; this happens during the initial user phase. |
| VS4_PRG (& VS4_GRP) | Systems are trusted when they are tested with data from past events, during the programming phase. |
| VH1_PRG (& VH1_GRP) | Users' first impressions make a big difference in whether or not the system will be accepted. Even at the programming stage, they can be kept informed of progress. |
| VH4_DES | Users should be involved in designing the system. |
| VH6_USR | Systems are trusted when clients speak well of them, or are willing to pay money for them. |

The following is an interpretation of the most important variables (those that appear in more than one equation) with negative coefficients:

**Table 9b: Most Important Variables with Negative Coefficients**

| VARIABLE | INTERPRETATION |
|---|---|
| T1_CAT | People don't trust systems with huge amounts of program code. |
| T3_CAT | People don't trust systems with many separate modules. |
| H3D_PRG | Multiple programmers do not inspire trust in systems. |
| H8_TOTAL | People don't trust systems that have too many people involved. |
| H9_EXPRT | The system expert always knows the faults and limitations of the system. |
| VT1_USR (& VT1_GRP) | That a system compiles and runs without crashing is necessary, but not sufficient, for system validation. |
| VS3_PRG | Determining if variables are complete, unbiased and accurate is not a good testing method during the programming phase; this is better done at the design stage. |
| VH3_PRG, VH3_DES (& VH3_GRP) | Comparing the results of a PSS to other expert sources, such as published literature, is not persuasive of the system's validity. |

Histograms of the frequency distribution of each of these key variables are listed in Appendix H. Recall that the independent (predictor) variables are assumed known in regression. No assumption (such as Normality) is made on the distribution of these predictor variables. The most desirable property is that the full range of conditions is represented by the sample for each predictor variable. This is generally true in this sample (see Appendix H). A good example is illustrated by O1_CAT, the frequency of occurrence of the event (Appendix H). The values for O1_CAT are the integers 0 to 7; all eight are represented in the sample, with at least two observations per category. Alternately, if the full range of conditions cannot be sampled, then it is important that the extremes of a variable are sampled well. This property is illustrated by the histogram of T3_CAT (number of modules), Appendix H. The only important variable that is not sampled well at both extremes is H8_TOTAL, the total number of people involved in the system (Appendix H). Only six systems were found with more than 100 people involved, and only one with more than 500. Such large systems are the hardest to sample, because they are almost all business systems that are highly proprietary.

The 37 variables listed in Appendix F are all significant in equations that explain from 28% to 86% of the variation in the 17 accepted hypotheses.[10] All the information that we can analyze in the data is contained in this variation. Therefore, it is important that managers understand not only how to interpret the individual variables, based on their content and coefficients, but also how the variables are combined in the models. For example, having developers involved in programming, H3D_PRG, (which carries a negative coefficient) is always accompanied in formulas with H2D_DES, developers being involved in the design phase (with a positive coefficient). H3D_PRG has been interpreted throughout Chapter 5 to mean that it is undesirable to have too many programmers working on the same program. But when combined with the knowledge that it appears desirable to have *more* of the *same* people — i.e, programmers — involved in *designing* the system, this could be a good guideline regarding the allocation of developers' time.

Another use of models is to compare the relative size of the coefficients in the formulas. Although it is the combination of all significant variables in an equation that is important for prediction, the relative importance of variables can be used to examine features of the testing environment. The software

[10]Sixteen of the variables are described in Tables 9a. and 9b. as "most important variables" because they appear in two or more of the equations; these are marked with an asterisk in Appendix F. The remaining variables each appear in one of the significant equations.

development environment in which one works can be informally analyzed in this way to determine strengths, weaknesses and possible areas of improvement. For example, the best 2-variable model for hypothesis $H_{14}$ is:

$$VER\% = 85.2 - 0.14 * H8\_TOTAL + 7.7 * VT11\_GRP$$

It predicts that the verification percentage is positively associated with "benchmarking" and negatively associated with large numbers of people involved in the system. A simple recommendation based on this hypothesis is to set and meet benchmark standards, but avoid getting too many people involved.

### 6.2.1 Regression Analysis

Standard statistical procedures were used to examine all models. The most fundamental test is an examination of the residuals (the observed minus predicted values). The residuals carry important information concerning the appropriateness of the model assumptions (Cook and Weisburg, 1982). Graphical tests are quite useful, since any pattern in the residuals that can be detected provides key information on the form of an alternative model or hypothesis (Draper and Smith, 1981). For example, if the residuals form a simple curved pattern when displayed versus variable $x$, then a quadratic (squared) term in $x^2$ is suggested. Specifically, no pattern (such as quadratic) was detected in the residual plots for any of the models corresponding to accepted hypotheses in this study. This meant that no evidence was found to reject linear models in favor of quadratic or logarithmic models, for example.

Histograms of the residuals for the 49 accepted models are displayed in Appendix I. The assumption in regression is that the errors $\epsilon$ are Normally distributed. For testing purposes, this assumption of Normality is quite robust to departures (Draper and Smith, 1981), so that valid conclusions can be drawn from the $t$-tests even when the distribution of residuals is not exactly symmetrical, for example. A quick examination of the 49 histograms in Appendix I reveals that the residuals of each model indeed approximate the Normal distribution quite well.

An outlier is a residual that is at least 3-4 standard deviations from the mean of the residuals. Outliers were not a serious problem in the analysis of the residuals. Obviously, such a residual indicates something rather unusual about the observation and the model itself. Although some researchers blindly reject outliers and reanalyze the data, this is not a wise course of action. Draper and Smith (1981) caution that sometimes an outlier arises from an unusual combination of circumstances that may be of vital interest to the study. Basically, the outliers are the only observations that can provide information that can improve the model; discarding them is thus the last thing that should be considered. As a general rule, outliers should be rejected out of hand only if they can be traced to errors in measuring or recording the observation (Draper and Smith, 1981). When questions arose regarding the accuracy of a recorded value on the survey questionnaire, the original respondent was contacted and the matter cleared up. In the set of 40 observations there are two or three that exhibit somewhat unusual circumstances in comparison with the others but definitely are not considered bad data. On the contrary, is it thought that these are representative of similar observations that would be present in a larger data set. As a result, no observations were discarded because of suspicious data values.

An unintended but interesting feature of the analysis was that it was incrementally repeated over time. Human subjects do not respond all on cue as the researcher would like. Instead, questionnaires trickled in slowly from around the world over the course of many months. The analysis was first conducted when approximately 20 questionnaires arrived. It was repeated when 30 had arrived, and repeated a final time when the $40^{th}$ respondent answered. As sample size increased, certain generalities appeared in the most important hypotheses. The most important variables in a given model remained significant as sample size increased. Furthermore, the sign of their estimated slope coefficient consistently remained positive or negative (see Tables 9a and 9b, and Appendix G). Marginally significant variables might have been replaced by another more significant variable as sample size increased, but the most important variables remained stable in a given model. This unintended extra analysis led increased credibility to the final models.

In additional to traditional multiple regression analysis, cluster analysis (SAS, 1985; Romesburg, 1984) was also tried. Cluster analysis is a generic name for a wide variety of mathematical methods for finding out which objects in a set are similar (Romesburg, 1984). For each hypothesis tested with multiple regression (Table 7), a corresponding hypothesis was tested using clustering. Standard clustering algorithms in SAS (1985) indeed produced clusters of "similar" observations, but these clusters had no predictive ability. In no instance was even 20% of the variation explained. The limitation of cluster analysis was that, although clusters of the data could easily be found, so many variables were involved that it was impossible to characterize the clusters in any way meaningful to this study. Although cluster analysis is a useful tool for problems involving classification, the goal here is not amenable to a classification paradigm. Rather, multiple linear regression was the appropriate analysis tool, simultaneously testing both the significance of a linear hypothesis and estimating the corresponding coefficients in one efficient procedure.

An additional advantage of linear regression is the ease in interpretation of results. Managers skeptical of inscrutable analyses and obscure modeling methods can easily understand a linear equation, as well as the trade-off between two variables with opposite signs. Likewise, the least squares criterion is not difficult to picture, and is quite intuitive. Cluster analysis requires the unusual ability to picture "similar" observations in a space that in this case is from 10 to 80 dimensions. Furthermore, the cluster analysis algorithms are rather intimidating, if not totally opaque to the layman. The linear model is the one that has received the greatest attention both in theory and in practice. From the theoretical point of view it is mathematically tractable, and in practical applications of wide variety it has shown itself to be of great value (Searle, 1971).

### 6.2.2 Recommendation for Managers

The best way for managers and sponsors of software development projects to benefit from this research is to systematically assess either current or recently completed projects in terms of both the equations and variables found here to be meaningful. It is strongly recommended that managers undertake their own research project to see how these results might be helpful in their circumstances. The project would be structured approximately as follows:

**1.** Identify a reasonably-sized group of people who are familiar with recent and current software development projects in the company or department.

**2.** Have the group first familiarize themselves with the 37 individual variables reported here and listed in Appendix G — definitions, category ('T'-Technical, 'H'-Human Involvement, 'O'-Observability and 'V'-Test Methods) and sign of their coefficients. Determine in group discussion if people generally agree that these variables can be important in influencing the reported validity of systems. This should take two scheduled group meetings, about one week apart.

**3.** Have the group familiarize themselves with how the variables combine to form the significant equations. The emphasis here should be on which variables typically occur together, and their relationships in the equations (shown by their signs and relative weights of the size of their coefficients). Three sub-groups could each study the equations in one set of accepted hypotheses (System Features, Validation Methods and Combination; see Appendix F). This should take one or two meetings for the groups to understand their responsibilities and report back. Discussion should be held to determine if these relationships are clear and make sense to everyone.

**4.** Have the group select one software development project (preferably a completed one) to analyze first. Have the group members individually record their opinions on the verification and validation levels for the system developed in this project. Collect and compile but do not publish these opinions at this time.

**5.** Using everyone's familiarity with the project, carefully evaluate each variable as to whether or not it was a characteristic of the project. Establish values for each appropriate variable (e.g., size of system in megabytes, number of people involved, which test methods were used). Score these values as was done for this study. (See the questionnaire, Appendix C, pages 2,3,4 and 6.) This can be done in meetings or on a checklist circulated among all parties. Publish the final list for all members of the group.

**6.** Record all variables representing the project under consideration on a matrix format similar to that in Appendix G. Discuss whether this project had a considerable number of "positive" and/or "negative" variables. Mark up the matrix in any way (colors, arrows, etc.) to note any patterns, weights or relationships among the variables that illustrate the group's experience with these factors in software development.

**7.** Select a few of the 17 hypotheses in Appendix F that the group agrees might be most meaningful in terms of the selected project. As accurately as possible, enter values for the variables in the equations that resulted from the hypotheses. Calculate the verification and validation percentages from the equations.

**8.** Calculate the verification and validation percentages predicted by all 49 equations, as in **7.**, above.

**9.** Compile the predicted V&V scores and the group members' reported V&V scores collected above in step 4. Present these at a group meeting and discuss the results.

**10.** Repeat steps 4-9 for all other appropriate projects. This process will become faster as the group gains skill in evaluating the variables and running the equations. (Surely someone will volunteer to computerize the routine parts of the process.)

**11.** Conclude the study by reaching agreement on the following points:

- The extent to which the equations in this study model your company's success in designing verified and valid systems.
- Characteristics of your company's development environment, policies and procedures that enhance the likelihood of producing highly verified and valid systems.
- Characteristics of your company's development environment, policies, and procedures that should be changed to improve the likelihood of producing highly verified and valid systems.
- Suggest ways to implement these changes.

It is important to remember that the tangible results of this study (the equations) do not predict only *high* reported levels of verification and validation. Rather, these particular equations are shown to quite accurately predict *any* level of V&V, depending on the values of the variables the represent real conditions that exist. This means that users of these equations can experiment with combinations of factors to discover, specifically, under what conditions high levels of verification or validation are predicted. This should be done early in the planning process; undertaking some "what-if" experimenting on paper is certainly less expensive and time- consuming than shifting people and resources after the project is underway.

## 6.3 Personal Reflections

This author has worked in the information systems design field for nearly 20 years. I have spent thousands of hours designing and programming computer systems, training and cajoling users to use my systems, and finding and fixing the inevitable bugs. I realized early on how important people are in this context: you can't have a "computer system" without real people to build it, promote it, test it, swear at it at times, depend on it and make it a part of their lives – or, at least, part of their work.

What I did not realize until I did this study is the extent to which verification and validation are also dependent – as are so many other system concepts – upon people. One would think that how "valid" a system is, or to what degree a system has been "verified," are objective concepts. This certainly is the image presented in the literature. In this study, since I was interested in how to improve validation, it was of foremost importance that I could find out how valid systems are in the first place. I contacted over 100 people to ask them about their Problem Solving Systems. I interviewed over

50 people in depth about what their systems do, and how humans interact with the systems through the many stages in the system lifecycle. Still it was not possible for me to test any of these systems myself. From a practical point of view, I do not know those systems—and the problems they are designed to solve—well enough to determine how valid they are. Even theoretically, it is not conceivable that the particular set of 40 systems that ended up in my study could be tested in any standardized, objective manner that would allow me, the researcher, to accurately assign validation scores to them. So what data did I use for each system's validation (and verification) score? I used what the respondent for each system told me. How did these respondents get their verification and validation "numbers"? Very frankly, in as many different ways as there are systems. I know from my personal contact with the respondents that some had run very detailed, technical tests on their systems which they reported, and that others reported their "gut feel". Is this uniform? Is it objective? No. Is it legitimate as a measure of how well people believe their systems have been tested? Absolutely. This is why throughout this study the "0%–100%" scores for the results of testing systems are referred to as the *"reported levels of .."* verification and validation.

This initial realization – that people not only do the work of verifying and validating systems, but they also make all final judgments – holds throughout the study. The results found here reflect and re-emphasize the importance of people in making (or breaking) any system's success. Not one of the results of the study can be explained without reference to the impact of that feature of the system on the people who use it. Take, for example, the three Technical Complexity system features that are found to be significant in this study: (1) amount of output, (2) amount of program code and (3) number of modules. In absolute terms, it means nothing to say that more or less of any of these should indicate that a system is more or less valid. Only when we understand that *people* associated with the systems respond to these features positively or negatively is it clear why they are important. People like lots of output—it provides information that can convince users that the system works as intended. They dislike large programs and complicated modular structures. Systems that meet these criteria are more likely to be considered valid than those that do not. Again, it is evident that the importance of these findings is their interpretation in the human context.

The one Observability system feature included in the list of the primary results demonstrates again the essential part people play in whether or not a system is valid. People tend to trust systems more if the real-world event that the system models happens frequently; it is hard to observe rare events, and thus it is harder for people to make judgments on whether systems model rare events correctly. Theoretically, this would also be true in an imagined context without humans. Rare events are experienced less often and therefore can be examined less often and therefore may not be able to be compared to the systems created to model them enough times to warrant conclusions of high reliability. But how will the experiencing, examining and comparing be accomplished? In a practical and very realistic sense, only people can do these tasks and make these judgments.

The Human Involvement system features show that, more often than not, the fewer people involved in a system, the better. Only having more designers involved in the system design phase is seen as a plus, perhaps reflecting a belief that information from a wide range of knowledgeable sources is necessary to solve complex problems. Having many programmers coding the system and having a large number of people involved in the system as a whole are negative influences on people's perception of system reliability (the too-many-cooks phenomenon).

All of the Testing Methods that are positively correlated with reported levels of system validation depend on roles humans play in system development or standards people set to judge them. Systems are preferred that (1) had users included in the design of the system; (2) elicited good first impressions from users; (3) met clients' standards and therefore clients accepted the systems by speaking highly of them and paying for them; and (4) were tested with data from past events or experience. The methods negatively correlated are noticeably less directly associated with human activity or influence on systems; it appears from this study that checking compilation, variables and published expertise do not convince people that they will be comfortable with a system.

My professional experience in the information systems field is entirely supportive of what this research has found. Allow me to give some personal examples on the following points:

• **User Involvement:** I have *always* found it essential to involve users at every point possible in the system lifecycle. The insights and information gathered in this way allow me to work much faster and result in a better product. Particularly in years past, I have met with some resistance, from not being permitted to ask users' advice on the initial design of a system, to including the people who will actually use a system in beta-testing. Either this is changing (thankfully) or I have progressed to the point that I don't ask permission for this anymore; I simply proceed to employ users' opinions and skills wherever I can.

• **Too many Programmers:** I personally do not enjoy working on a team of programmers as much as creating and seeing a program through to the finish myself; therefore I am glad to find that this contributes to a good perception of my work. Clearly there are many circumstances in which a small team project is appropriate and efficient. I have worked on teams that were productive and, unfortunately, on one team that fell apart. I think that the strength of this particular result of the study occurs because most people have a horror story about wasted efforts in team projects that never finished, or were passed from programmer to programmer; programmers themselves are the most critical of others' "spaghetti code".

• **Client acceptance:** There is no greater joy for a system designer like me than when a client writes the check. These days, of course, we usually don't wait until the end of a project to get paid in full, but the symbolism is the same: The work has value, and the value is recognized. Equally important is the fact that others know that the client's standards have been met; this confidence inspires confidence in others. Acceptance of the system by the client is obviously important to the system developer, but it is bit of a surprise to learn from this study that it is such an important factor in influencing the opinions of many people about whether or not a system is considered to be valid.

• **Lots of Output:** Users are frustrated when they know that they've put volumes of data into a system, but then can't get it out. Systems that provide numerous ways to retrieve data not only help workers do their jobs more efficiently, but also satisfy users' curiosity about whether or not the system is recording, manipulating and reporting "their" data accurately.

• **Productive Testing:** The best type of routine system testing according to the results of this study is comparing system output with historical data. I have never developed a system where I did not ask users to provide a variety of real examples that I carefully reproduced in the system, and then checked the results. Using familiar real data makes sense to users; seeing that my system mimics real results makes me confident of my design.

- **Unproductive Testing:** Of the dozens of validation methods included in this study, three rather technical ones turn out to be negatively associated with reported levels of validation. One can conclude that checking the accuracy of variables and checking system output against other published results, at least during the programming phase, are not a good use of time. It is quite surprising that the third technical method — that the system compiles and runs without crashing — shows up as a negative. My interpretation here is that when asked about this as a validation method, people's response was, so what? It does show some progress in our sophistication about validation that people today regard successful compilation and absence of catastrophic program errors as "givens" — necessary but not sufficient for saying anything about the degree to which a system is really valid.

- **The Expert Knows:** One fascinating important result of the study is that getting experts' opinions about systems has a negative influence on reported levels of validation. Why should this be? Any system is a model, an abstraction, of reality, and the expert is by definition the person who best understands that reality. Thus the expert knows where the system fails, or is limited, in representing all facets of the real problem and its solution. We can truly say that the expert is never satisfied. Contrast this to another result that shows that programmers' opinions are positively associated with reported levels of validation. Not surprisingly, programmers are more easily satisfied with their systems than are the experts.

## 6.4 Areas for Future Research

The contributions to the field of Verification and Validation made by this study are considerable. The two primary contributions are:

- Compilation and organization of more validation methods than have ever before appeared together in the literature: the Spectrum of Validation Methodologies (Appendix A).

- Identification, through scientifically replicable data collection and analysis techniques, of a set of system features and validation methods that have the most impact on reported levels of system validation.

The goal of science is to create new knowledge, a necessarily never-ending process. Certainly more work can be done that builds on this study and its results. Following are suggestions for future areas of research building on the results of the current study.

- **Issue #1:** Are systems necessarily harder to validate as they become more technically complex?

This hypothesis depends to some extent on the degree of structure in a given problem. In a financial analysis system for business, the problem is highly structured. The nature of this particular system may lead to *increasing* (not decreasing) trust in the validity of the Problem Solving System as the complexity—the number of necessary financial details and their relationships—increases. Contrast this to the very unstructured problem domain of Executive Decision Support. The number of relevant factors a successful manager must consider increases geometrically as the complexity of the management environment increases.

**Recommendation #1:** (1) Split the data into two groups, "High" and "Low" complexity, and repeat the study separately for each group. (2) Augment the questionnaire to determine whether the problem domain is well or poorly structured and include this factor in the analysis.

• **Issue #2:** Is there a difference in how successful validation can be enhanced that depends on domain?

This is a fruitful hypothesis for future testing, because the three major domain groups have mutually exclusive goals. Generally, the goal of Business is to make money, the goal of Research is to discover new knowledge, and the goal of Government is to provide services. Because of the disparate nature of these goals, it is possible that successful validation will vary with the degree of complexity, for example, needed for a system to meet the given goal in its specific domain. Proper testing of this hypothesis requires at least three times more data than available in the current study.

**Recommendation #2:** Using the three domain areas of Business, Research and Government, expand the study to obtain enough data to repeat the analysis separately for each group.

• **Issue #3:** What importance should be attached to the fact that people consistently interchange the terms "verification" and "validation"?

Based on etymology, the confusion is not surprising. Science works by precisely defining a concept and not changing that definition. It does not follow that those definitions are actually being followed, however. Additional information pointing to the internal nature of verification and the external nature of validation could be elicited from respondents to help distinguish between these two concepts.

**Recommendation #3:** Create a study to determine how well this study's respondents understand the distinction between verification and validation and compare to the levels they report for their systems.

• **Issue #4:** Do developers and users validate their systems in ways not represented in this study?

The Research Questionnaire was a direct outgrowth of published methods for testing Problem Solving Systems. Proprietary by nature, business systems are often not reported outside their companies, much less published in the literature. Thus there could exist successful methods for testing business systems that have never been published. Unless it violates company policy, business respondents can be quite forthcoming if they are assured that confidentiality will be maintained and the proprietary nature of the system not compromised. Two ideas for business "rule of thumb" validation methods that have come up in personal conversations about this study are the following:

(1) A highly regarded system is one that returns the dollar amount invested in it as quickly as possible.

(2) A system is more likely to be accepted and trusted the shorter the period that exists from the time a need is recognized to the time a system is implemented to meet that need. If managers or sponsors drag out the decision-making and development time, enthusiasm for the system wanes, and it becomes increasingly likely that skepticism that the problem will ever be handled well can be overcome.

**Recommendation #4:** Canvas respondents (and others) for additional types of validation methods, and add these to the survey.

• **Issue #5:** How closely are the concepts of people's "confidence", "trust" and "belief" in systems related to verification and validation?

Semantics plays an important role in our ability to measure abstract concepts relating to validation. Indeed, even with the definitions of Verification and Validation stated in the Questionnaire, some European respondents still switched the standard definitions. Although definitions for "confidence", "trust" and "belief" could be provided, it is likely that respondents will answer based on a very personal understanding of these concepts.

**Recommendation #5:** Expand the section of the questionnaire that asks people to rate verification and validation from 0% to 100%, using the same scale to score their trust, confidence, and belief in their systems. Repeat the analysis, using these new dependent variables, and compare to the results for verification and validation.

• **Issue #6:** What is the relationship to verification and/or validation of concepts such as "credibility", "reliability", "correctness", "usefulness" and similar terms describing desirable outcomes of system testing?

Again, this area of semantics is interesting and important to V&V. Some authors have used these terms interchangeably, referring to the results of certain types of verification or validation methods. Others make a point to use these words instead of "verification" or "validation" in an effort to avoid the confusion already pointed out concerning these two broad terms. In the future this might help focus V&V efforts, both practically and theoretically.

**Recommendation #6:** First, search the literature to find definitions and examples of the use of these terms and relate them to specific categories or types of V&V testing (using or enhancing the Spectrum of Validation Methodologies developed here). Second, question system developers and users to determine if these concepts are more easily investigated and/or understood than the broad definitions of V&V. Try to determine if substituting these concepts and terms helps or hinders the overall goal of producing systems that people find credible, reliable and correct.

## 6.5 Conclusion

The series of hypotheses tested in this study provide useful information about what should be expected of the validation process, and where resources are best spent. Hypotheses that are considered acceptable and therefore useful explain between 28% and 86% of the total variation. The concept of useful hypotheses corresponds directly to the concept of useful models. While absolute validation of models can never be attained, *usefulness* is a very important criterion. Fedra *et al.* (1987) emphasize that it is far more important that users have confidence in the utility of a system than it is to focus on the impossible task of proving total system validity. Mankin *et al.* (1975) appear to concede that all models are invalid, but immediately point out that there exists a large class of "invalid but useful" models and that it is most fruitful to focus on how useful a model is, and how to make it more so. System managers and developers know they are not working in a perfect world. But neither are they adrift at sea without any knowledge at all of the validity and usefulness of their systems. They can determine a starting point, and a reasonable path to take to improve confidence in a given system (Landry *et al.*, 1983; Fedra *et al.*, 1987). A system that is reported to be, say, 50% valid (usually with a higher percentage for verification) is clearly a useful system in most contexts, especially if such a characterization of the system is viewed as one stage in a dynamic process that leads to continued improvement (Miller 1989; Sacerdoti, 1991).

Although much from this analysis can be concluded about system features and testing methods that are associated with reported validation percentages, caution should be exercised. This analysis is not about cause and effect; that could only be established by undertaking carefully controlled experiments, a difficult if not impossible task. Rather, this study is concerned with relationships to and influences on validation. There do exist useful combinations of variables that are significantly related to people's perceptions of the validity of Problem Solving Systems. These variables are not just abstract constructs but instead correspond directly to system features and testing methods of systems in use. They correspond to conditions and methods that are *related* to higher, or lower, reported levels of people's confidence in those systems. Most of these factors are under the control of the developers, managers, and clients; many could be specified as requirements at the appropriate stages of the system lifecycle.

Because "truth" has been traditionally viewed as an absolute albeit unobtainable attribute, it is not surprising that many have been skeptical about the possibility of validating models (Mitroff, 1969; House, 1975; Shechter & Lucas, 1980; Gass, 1983; Pappas & Remer, 1984; O'Keefe, 1987; Fetzer, 1988; Finlay & Wilson, 1991; Kleijnen, 1995). Although this is a philosophically safe position, it is not particularly helpful to those whose jobs depend on creating and using software. Popper's (1963) dictum that nothing can be proven to be true could easily be used as an excuse to ignore the practical problem of model validation. However, it is far more productive to use it as an upper bound on our knowledge of a system — knowledge that can be steadily improved with sequential hypothesis testing (Platt, 1964; Mankin *et al.*, 1975; Loehle, 1983) or iterative improvement of the Problem Solving System (Gass, 1983; Miller 1989; Sacerdoti, 1991). The view taken here is that model validation is a process, not an event (Mankin *et al.*, 1975). Results of this study help us focus our efforts in the process toward the goal of at least partial validation.

This study accomplished another useful thing: the winnowing out of a great many system features and test methods that apparently are not associated with reported levels of validation. Managers and developers have hundreds of decisions to make; this study has defined a relatively small set of the most important considerations. Requiring certain features in development environments or certain types of testing that were not found to be significant in this study may only have the effect of wasting time and money. At the minimum, this study has determined specific acceptable hypotheses, meaningful models, and significant variables as a place to start.

The main contribution of this study is that it identifies broadly applicable ways to create environments and procedures for enhancing successful validation, not only in specific circumstances, but for large numbers of systems. Thus these results help to efficiently focus managerial efforts and corporate resources toward the goal of increasing the acceptance and therefore the usefulness of computer-based Problem Solving Systems.

# LITERATURE CITED

**Bachant, J. and J.McDermott** (1984) "R1 Revisited: Four Years in the Trenches", *The AI Magazine*, Fall 1984:21-32.

**Baglow, R.L.** (1977) "Models and Managers Experience in the Introduction of a Mathematical Inventory Model", *Infor*, 15(2):264-271.

**Balci, O.** (1994) "Validation, Verification, and Testing Techniques Throughout the Life Cycle of a Simulation Study", *Proceedings of the 1994 Winter Simulation Conference*; Tew, Manivannan, Sadoeski and Seila, *eds.*, pgs. 215-220.

**Balci, O. and R.G.Sargent** (1983) "Validation of Multivariate- Response Trace-Driven Simulation Models", *Performance '83*, A.K. Agrawala and S.K. Tripathi, *eds.*, North-Holland, New York, pgs. 309-323.

**Balci, O. and R.G.Sargent** (1984) "Validation of Simulation Models Via Simultaneous Confidence Intervals", *American Journal of Mathematical and Management Sciences*, 4(3&4):375-406.

**Barlas, Y.** (1989) " Multiple Tests for Validation of System Dynamics Type of Simulation Models", *European Journal of Operations Research*, 42(1989):59-87.

**Baroudi, J.J. and W.J.Orlikowski** (1989) "The Problem of Statistical Power in MIS Research", *MIS Quarterly*, March, 1989, 13(1):87-106.

**Baughman, M.L.** (1980) "Reflections on the Model Assessment Process: A Modeler's Perspective", *Validation and Assessment Issues of Energy Models*, S.I. Gass, *ed.*, Proceedings of a Workshop (NBS Special Publication, 569), pgs.197-203.

**Bell, P.C.** (1985) "Visual Interactive Modeling in Operational Research: Successes and Opportunities", *Journal of the Operations Research Society*, 36(11):975-982.

**Bell, P.C.** (1989) "Stochastic Visual Interactive Simulation Models", *Journal of the Operations Research Society*, 40(7):615-624.

**Bliss, J.R., P.Feld and R.E.Hayes** (1986) "Decision Aid Measurement and Evaluation (DAME)", *Proc. of the 1986 IEEE, International Conference on Systems, Man and Cybernetics*, pgs.126-130.

**Brunk, H.D.** (1965) *An Introduction to Mathematical Statistics*, Blaisdell Publishing Company, New York, 429 pages.

**Buchanan, B.G. and E.H. Shortliffe** *eds.* (1984) *Rule-Based Expert Systems: The MYCIN Experiments of the Stanford Heuristic Programming Project*, Addison-Wesley Publishing Company.

**Campbell, D.T. and D.W. Fiske** (1959) "Convergent and Discriminant Validation by the Multitrait-Multimethod Matrix", *Psychological Bulletin*, Vol. 56, March 1959, pgs.81-105.

**Carter, G. and E.Ignall** (1970) "A Simulation Model of Fire Department operations", *IEEE Transactions, Systems Science & Cybernetics*, vol.6, pgs.282-293.

**Casti, J.L.** (1980) "On System Complexity: Identification, Measurement, and Management (Chapter 6)", *Complexity, Language and Life: Mathematical Approaches*, J.L. Casti & A. Karlqvist, *eds.*, Springer-Verlag, pgs.146-173.

**Casti. J.L. and A.Karlqvist** (1980) "Introduction: Complexity and the Evolution of Living Systems", *Complexity, Language and Life: Mathematical Approaches*, J.L. Casti & A. Karlqvist, *eds.*, Springer-Verlag, pgs.xi–xiii.

**Castore, G.** (1987) "A Formal Approach to Validation and Verification for Knowledge-Based Control Systems", *First Annual Workshop on Space Operations Automation and Robotics, NASA*, (SOAR 87):197-202.

**Caswell, H.** (1976) "The Validation Problem", *Systems Analysis and Simulation in Ecology*, B.C. Patten *ed.*, Academic Press, New York, pgs.313-325.

**Cazalet, E.G.** (1980) "A Decision Analyst's View of Model Assessment", *Validation and Assessment Issues of Energy Models*, S.I. Gass, *ed.*, Proceedings of a Workshop (NBS Special Publication, 569), pgs.325-336.

**Chaiken, J.M.** (1971) "The Number of Emergency Units Busy at Alarms Which Require Multiple Servers", R-531, The Rand Corporation, Santa Monica, CA.

**Charniak, E. and D.McDermott** (1987) *Introduction to Artificial Intelligence*, Addison-Wesley Series in Computer Science.

**Chlebus, E.** (1991) "Validation of Models for a Telephone Network Employing Routing Over Transit Paths", *Applied Mathematical Modelling*, August, 1991, vol.15:440-443.

**Churchman, C.W.** (1971) *The Design of Inquiring Systems*, Basic Books, Inc., New York, London, 277 pages.

**Churchman, C.W.** (1979) *The Systems Approach*, Dell Publishing Company, New York, 243 pages.

**Coates, D., Finlay, P. and J.Wilson** (1991) "Validation in Marketing Models", *Journal of the Market Research Society*, 33(2):83-90.

**Cobham, A.** (1954) "Priority Assignments in Waiting Line Problems", *Operations Research*, vol.2, pgs.70-76; correction *ibid.*, vol.3 pg.547.

**Cochran, E.L. and B.L.Hutchins** (1987) "Testing, Verifying and Releasing an Expert System — The Case History of Mentor", *Proceedings of the Third Conference on Artificial Intelligence Applications*, Kissimmee, FL, March, 1987. IEEE Computer Society press, Washington, D.C., 163-167.

**Cohen, J.** (1960) "A Coefficient of Agreement for Nominal Scales", *Educational and Psychological Measurement*, 20:37-46.

**Cohen, J.** (1968) "Weighted Kappa: Nominal Scale Agreement with Provision for Scaled Disagreement or Partial Credit", *Psychological Bulletin*, 70(4):213-220.

**Cohen, J.** (1977) *Statistical Power for the Behavioral Sciences, Revised Edition*, Academic Press, New York.

**Cook, R.D. and S. Weisberg** (1982) *Residuals and Influence in Regression*, Chapman and Hall, New York, 230 pages.

**Cook, T.D. and D.T.Campbell** (1979) *Quasi-Experimentation: Design and Analytical Issues for Field Settings*, Rand McNally, Chicago, Il.

**Cronbach, L.J.** (1951) "Coefficient Alpha and the Internal Consistency of Tests", *Psychometrika*, vol.16, September 1951, pgs.297-334.

**Cronbach, L.J.** (1971) "Test Validation", *Educational Measurement*, 2nd Edition, R.L. Thorndike *ed.*, American Council on Education, Washington, D.C., pgs. 443-507.

**Cronbach, L.J. and P.E. Meehl** (1955) "Construct Validity in Psychological Tests", *Psychological Bulletin*, vol. 52, pgs. 281-302.

**Csurgay, A.** (1988) "Information-Based and Computational Complexity of Physically Realizable Machines", *Beyond Number Crunching*, Austrian-Hungarian Conference, September 14-16, 1988, published by the John von Neumann Society for Computing Sciences, V. Haase & E. Knuth, *eds.*, pgs.9-13.

**Cumming, P.D., Kendall, K.E., Pegels, C.C., Seagle, J.P. and Shubsda, J.F.** (1976) "A Collections Planning Model for Regional Blood Suppliers: Description and Validation", *Management Science*, May, 1976, 22(9):962-971.

**Cutler, M.M** (1980) "Validation Proofs of Discrete Event Simulation Programs", Summer Computer Simulation Conference, Seattle, WA, Simulation Council, La Jolla, CA, pgs.240-245.

**Davis, D. and R.M. Cosenza** (1985) *Business Research for Decision Making*, Kent Publishing Company, Boston, Mass., pgs.172-175.

**Davis, M.J.** (1989) "Applying Expert Systems Technology to Communications Software Validation", *IEEE Proceedings, Fifth Conference on Artificial Intelligence Applications*, pgs.209-213.

**Derkinderen, F.G.J. and R.L.Crum** (1988) "The Development and Empirical Validation of Strategic Decision Models", *International Studies of Management and Organization*, XVIII(2):29-59.

**Draper, N.R. and H.Smith** (1981) *Applied Regression Analysis, 2nd Edition*, John Wiley & Sons, New York.

**Dreyfus, H.** (1972) *What Computers Can't Do: A Critique of Artificial Reason*, Harper & Row, New York.

**Dreyfus, H.L. and S.E.Dreyfus** (1988) "Making a Mind Versus Modeling the Brain: Artificial Intelligence Back at a Branchpoint", *Deadalus*, Winter, 1988, pgs.15-43.

**Dörner, D.** (1987) "On the Difficulties People Have in Dealing With Complexity", *New Technology and Human Error*, J. Rasmussen, K. Duncan & J. Leplat, *eds.*, John Wiley & Sons, pgs.97-109.

**Eckenrode, R.T.** (1965) "Weighing Multiple Criteria", *Management Science*, vol. 12, pgs.180-191.

**Eilon, S.** (1974) "Mathematical Modeling for Management", *Interfaces*, 4(2):32-38.

**Elleby, P., H.E.Fargher and T.R.Addis** (1987) "Evaluating a Knowledge Based Scheduling System", *IEE Colloquium on Testing Expert Systems*, London, UK, 3 Dec. 1987. IEE 1987 Digest No. 110, 2/1-4.

**Emshoff, J.R. and R.L.Sisson** (1970) "Design and Use of Computer Simulation Models", Macmillan, New York, 1970

**Enand, R., G.S.Kahn and R.A.Mills** (1990) "A Methodology for Validating Large Knowledge Bases", *International Journal of Man-Machine Studies*, vol.33:361-371.

**Falk, H. and L.A.Gordon** (1978) "Assessing Industry Risk by Ratio Analysis: Validation", *The Accounting Review*, Jan., 1978, vol.LIII(1):216-227.

**Fedra, K.** (1988) "Information and Decision Support Systems for Risk Analysis", *Proceedings of the IFAC Symposium*, Fulda, FRG, 9-11 November, 1988, pgs.53-59.

**Fedra, K., E.Weigkricht and L.Winkelbauer** (1987) "Intelligent Information-and Decision-Support Systems: Hazardous Substances and Industrial Risk Management", Advanced Computer Applications Project, IIASA, Laxenburg, Austria, pgs.1-8.

**Fetzer, J.H.** (1988) "Program Verification: The Very Idea", *Communications of the ACM*, September, 1988, 31(9):1048-1063.

**Finlay, P.N., Forsey, G.J. and J.M.Wilson** (1988) "The Validation of Expert Systems—Contrasts with Traditional Methods", *Journal Operational Research Society*, 39(10):933-938.

**Finlay, P.N. and J.M.Wilson** (1991) "Validation for Decision Support Systems: Resent Developments and Findings", *Systems Practice*, 4(6):599-610.

**Finlay, P. and J.Wilson** (1992) "Validity of Decision Support Systems: Towards a Contingent Validation Methodology", working paper 1992:10, Loughborough University Management Research Series; referenced with permission of the senior author.

**Fischer, G.W.** (1977) "Convergent Validation of Decomposed Multi-Attribute Utility Assessment Procedures for Risky and Riskless Decisions.", *Organizational Behavior and Human Performance*, vol.18, pgs.295-315.

**Fishman, G.S. and P.J.Kiviat** (1967) "The Analysis of Simulation Generated Time Series", *Management Science*, vol.13, pgs.525-557.

**Fishman, G.S.** (1973) *Concepts and Methods in Discrete Event Digital Simulation*, Wiley, New York.

**Forman, Leonard** (1978) "The New York Times Newspaper Planning Model", *Corporate Planning Models*, Thomas Naylor, *ed.*, Addison-Wesley Publishing Co., Menlo Park, California, pgs.305-316.

**Fox, B.** (1978) "Estimation and Simulation", *Management Science*, 24(8), pgs.860-861.

**Gale, B.T.** (1978) "Cross-Sectional Analysis: The New Frontier in Planning", *Planning Review*, March, 1978, 6(2):17-20.

**Gale, G.** (1979) *Theory of Science: An Introduction to the History, Logic, and Philosophy of Science*, McGraw-Hill Book Company, New York, 298 pages.

**Gaschnig, J., P.Klahr, H.Pople, E.Shortliffe, and A.Terry** (1983) "Evaluation of Expert Systems", *Building Expert Systems*, F. Hayes-Roth, D.A. Waterman, D.B. Lenat *eds.*, Addison-Wesley, Reading, MA.

**Gass, S.I.** (1983) "Decision-Aiding Models: Validation, Assessment and Related Issues for Policy Analysis", *Operations Research*, July-August, 1983, 31(4):603-631.

**Gerardy, R.** (1981) "On the Complexity of Paradigms", *Applied Systems and Cybernetics, Proceedings of the International Congress on Applied Systems Research and Cybernetics*, G.E. Lasker, *ed.*, Volume II: Systems Concepts, Models and Methodology, Pergamon Press.

**Gödel, K.** (1931) "Über formal unentscheidbare Sätze der Principia Mathematica und verwandter Systeme I.", *Monatshefte für Mathematik und Physik*, vol. 38 (1931), 173-198. Authorized translation by John van Heijenoort in *From Frege to Gödel* (see below).

**Green, L. and P.Kolesar** (1984) "A Comparison of the Multiple Dispatch and M/M/c Priority Queueing Models", *Management Science*, vol.30 pgs.665-670.

**Green, L. and P.Kolesar** (1989) "Testing the Validity of a Queueing Model of Police Patrol", *Management Science*, February, 1989, 35(2):127-148.

**Greenberg, J.H. and M.Ruhlen** (1992) "Linguistic Origins of Native Americans", *Scientific American*, November, 1992, pgs.94-99.

**Groner, R., M.Groner and W.F.Bischof** (1983) *Methods of Heuristics*, Lawrence Erlbaum Associates, Pubs., London, 416 pages.

**Groundwater, E.H.** *et al.* (1987) (Science Applications) "Approaches to the Verification and Validation of Expert Systems for Nuclear Power Plants", Electric Power Research Institute, EPRI NP 5236, July 1987.

**Gruhl, J.** (1982) "Model Credibility and Independent Evaluation: Three case Studies", *Omega* 10(5):525-537.

**Guion, R.M.** (1965) "Synthetic Validity in a Small Company: A Demonstration", *Personnel Psychology*, vol.18, pgs.49-63.

**Gutting, G.,** *ed.* (1980) *Paradigms & Revolutions: Applications and Appraisals of Thomas Kuhn's Philosophy of Science*, University of Notre Dame Press, Notre Dame, IN., 339 pages.

**Hall, I.O.** (1986) "Towards Authenticating a Multi-purpose Expert System", *Proceedings of NAFIPS '86—1986 Conference of the North American Fuzzy Information Processing Society*, New Orleans, LA, 2-4 June 1986, pgs.160-168.

**Hall, L.O., M.Friedman and A.Kandel** (1989) "On the Validation and Testing of Fuzzy Expert Systems", *IEEE Transactions on Systems, Man and Cybernetics*, 18(6):1023-1027.

**Harwell, M.A.** (1984) *Nuclear Winter: The Human and Environmental Consequences of Nuclear War*, Springer-Verlag, New York.

**Hashimoto, H.** (1983) "A World Iron and Steel Economy Model: A Projection for 1980-95", *Journal of Policy Modeling*, 5(3):379-396.

**Hayes-Roth, F.** (1985) "Rule-Based Systems", *Communications of the ACM*, 28(9):921-932.

**Helman, D.H. and J.L.Bennett** (1988) "Theories of Explanation: Expert Systems and Simulation", *Human Factors in Management Information Systems*, J.Carey, *ed.*, Chapter 10:167-180.

**Henderson, J.C. and P.C.McNutt** (1980) "The Influence of Decision Style on Decision-Making Behavior", *Management Science*, 26(4):371-386.

**Hermann, C.F.** (1967) "Critique and Comment: Validation Problems in Games and Simulation Models of International Politics", *Behavioral Science*, 12(3):216-231.

**Hickam, D.H., E.H.Shortliffe, M.B.Bischoff, A.C.Scott, and C.D.Jacobs** (1985) "The Treatment Advice of a Computer-based Cancer Chemotherapy Protocol Advisor", *Annals of Internal Medicine*, vol. 103, pgs.928-936.

**Ho Kang, B., W.Gambetta and P.Compton** (1996) "Verification and validation with Ripple-Down Rules", *International Journal of Human-Computer Studies*, 44(2):257-269.

**Hoepfl, R.T. and G.P. Huber** (1970) "A Study of Self-explicated Utility Models", *Behavioral Science*, vol. 15, pages 408-414.

**Hoffman, E., S.J.Varghese, A.B.Whinston and J.R.Marsden** (1987) "Development, Use and Verification of Expert Systems in Modelling Microeconomic Systems", NATO ASI Series, *Decision Support Systems: Theory and Applications*, vol.F31:411-428.

**Hollenbeck, J.R. and E.M.hitener** (1988) "Criterion-Related Validation for Small Sample Contexts: An Integrated Approach to Synthetic Validity", *Journal of Applied Psychology*, August, 1988, 73(3):536-544.

**House, P.W.** (1975) "Diogenes Revisited - The Search for a Valid Model", *Simulation*, 23(4):117-125.

**Hudson, D.L.** *et al.* (1984) "Prospective Analysis of Emerge: an Expert

System for Chest Pain Analysis", *IEEE Computers in Cardiology*, IEEE Service Center, Piscataway, N.J., pgs.19-24.

**Ignall, E.J. and R.Urbach** (1975) "The Relationship between Fire Fighting Unit Availability and the Number of Units Dispatched", P-5420, The Rand Corporation, Santa Monica, CA.

**Ignall, E.J., P.Kolesar and W.E.Walker** (1978) "Using Simulation to Develop and Validate Analytic Models: Some Case Studies", *Operations Research*, March-April, 1978, 26(2):237-253.

**Johnson, S.C.** (1988) "Validation of Highly Reliable, Real-Time Knowledge-Based Systems", *NASA 2nd Annual Workshop on Space Operations Automation and Robotics* (SOAR 1988), pgs.123-129.

**Kang, Y. and T.Bahill** (1990) "A Tool for Detecting Expert- System Errors", *AI Expert*, Feb. 1990 47-51.

**Kennedy, H.A.** (1971) "Judged Importance of the Dimensions of a Loan", University of Washington (Seattle), Department of Psychology.

**Kerlinger, F.** (1973) *Foundations of Behavioral Research*, Holt, Rinehart & Winston, New York.

**Kerr, R.M. and R.V.Ebsary** (1988) "Implementation of an Expert System for Production Scheduling", *European Journal of Operational Research*, North-Holland, vol.33, pgs.17-29.

**Kirk, R.E.** (1968) *Experimental Design: Procedures for the Behavioral Sciences*, Brooks/Cole Publishing Compnay, Belmont, California.

**Kheir, N.A. and W.M.Holmes** (1978) "On Validating Simulation Models of Missile Systems", *Simulation*, April, 1978, pgs.117-128.

**Kleijnen, J.P.C.** (1995) "Verification and Validation of Simulation Models", *European Journal of Operational Research*, Elsevier Science B.V.

**Kors, J.A., A.C.Sittig and J.H.van Bemmel** (1990) "The Delphi Method to Validate Diagnostic Knowledge in Computerized ECG Interpretation", *Methods of Information in Medicine*, 29(1990):44-50.

**Kugel, P.** (1986) "Thinking may be more than Computing", *Cognition*, 22(1986):137-198.

**Landry, M., J.-L.Malouin, and O.Muhittin** (1983) "Model Validation in Operations Research", *European Journal of Operations Research*, 14(1983):207-220.

**Langlotz, C.P., E.H.Shortliffe and L.M. Fagan** (1986) "Using Decision Theory to Justify Heuristics", *Proc. AAAI '86*, Menlo Park, California, 1986, pgs.215-219.

**Larson, R.C.** (1972) *Urban Police Patrol Analysis*, The MIT Press.

**Law, A.M. and M.G.McComas** (1990) "Secrets of Successful Simulation Studies", *Industrial Engineering*, May, 1990, 2:47-53.

**Light, R.J.** (1971) "Measures of Response Agreement for Qualitative Data: Some Generalizations and Alternatives", *Psychological Bulletin*, 76(3):363-377.

**Linstone, H.A.** (1984) *Multiple Perspectives for Decision Making*, Elseview Science Publishing, New York, 413 pages.

**Loehle, C.** (1983) "Evaluation of Theories and Calculation Tools in Ecology", *Ecological Modelling 19*, Elsevier Science Publishers, Amsterdam, pgs.239-247.

**Long, J.A. and I.Neale** (1990) "Validating and Testing with Multiple Experts in KBS", DBSAD, City University, London, EC1V OHB, England.

**Lopez, B., P.Meseguer and E.Plaza** (1989) "Validation of Knowledge Based Systems: An*(sic)* State of the Art", Grup de Reserca en Intelligència Artificia i Lògicia, Centre d'Estudis Avançats de Balnes, CSIC, 17300 Blanes, Catalunya, Spain, June, 1989.

**Luck, H.O. and U.Schlossarek** (1987) "Software Controlled Fire Detection Systems", *Fire Science and Technology*, Nov.2, 1987, vol.7 pgs.53-60.

**Luhmann, N.** (1978) "Temporalization of Complexity", *Sociocybernetics*, vol.2, R.F. Geyer and J. van der Zouwen, *eds.*, Martinus Nijhoff Social Sciences Division, Leiden/Boston/London.

**Mankin, J.B., R.V.O'Neill, H.H.Shugart, and B.W.Rust** (1975) "The Importance of Validation in Ecosystem Analysis", *New Directions in the Analysis of Ecological Systems*, G.S. Innis, *ed.*, Part I. Simulation Council, La Jolla, California, 5(1):63-71

**Mayer, L.S.** (1980) "On a Perspective for Energy Model Validation", *Validation and Assessment Issues of Energy Models*, S.I. Gass, *ed.*, Proceedings of a Workshop (NBS Special Publication 569, 1980), pgs.477-495.

**McCall, M.W. and M.M.Lombardo** (1982) "Using Simulation for Leadership and Management Research: Through the Looking Glass", *Management Science*, 28(5):533-549.

**McCormick E.J., A.S.DeNisi and J.B.Shaw** (1979) "Use of the Position Analysis Questionnaire for Establishing the Job Component Validity of Tests", *Journal of Applied Psychology*, vol.64, pgs.51-56.

**Mendenhall, W. and R.L. Scheaffer** (1973) *Mathematical Statistics with Applications*, Duxbury Press, North Scituate, Massachusetts, 561 pages.

**Mihram. G.A.** (1972) *Simulation: Statistical Foundations and Methodology*, Academic Press, New York, pgs.250-254.

**Miller, L.W., R.J. Kaplan and W. Edwards** (1967) "Judge: A Value-Judgment-Based Tactical Command System" *Organizational Behavior and Human Performance*, vol. 2, pgs.329-374.

**Miller, M.J.** (1989) "Verification and Validation of Decision Support Expert Systems: Chemical Process Risk Management in International Operations", *American Chemical Society Symposium Series* (1989), v.408:126-146.

**Mitchell, T.R.** (1985) "An Evaluation of the Validity of Correlational Research Conducted in Organizations", *Academy of Management Review*, (10:2), pgs.192-205.

**Mitroff, I.I.** (1969) "Fundamental Issues in the Simulation of Human Behavior: A Case Study in the Strategy of Behavioral Science", *Management Science*, August, 1969, 15(12):B635-B649.

**Monserud, R.A.** (1989) "Optimizing Single-tree Simulators", *Artificial Intelligence and Growth Models for Forest Management Decisions*, H.E. Burkhart, H.M. Rauscher & J. Johann, *eds.* School of Forestry and Wildlife Resources, Virginia Polytechnic Institute and State University, Publication No. FWS-1-89:308-321.

**Monserud, R.A., O.V.Denissenko and N.M.Tchebakova** (1993) "Comparison of Siberian Paleovegetation to Current and Future Vegetation Under Climate Change", *Climate Research*, vol.3, pgs.143-159.

**Moore, P.G.** (1983) *The Business of Risk*, Cambridge.

**Morehead, L.A.** (1990) "Verification and Validation of Expert Systems", manuscript on file at Portland State University, Systems Science Ph.D. Program, Portland, Oregon. 40 pages.

**Naylor, T.** (1978) "PIMS - Through a Different Looking Glass", *Planning Review*, March, 1978, vol.6, page 15+.

**Naylor, T., J.Balintfy, D.Burdick and K.Chu** (1966) *Computer Simulation Techniques*, Wiley, New York.

**Naert, P.A. and P.S.H.Leefang** (1978) "Building Implementable Models", Leiden/Boston: Martinus/Neijhoff.

**Newman, J.R.** (1975) "Assessing the Reliability and Validity of Multi-Attribute Procedures: An Application of the Theory of Generalizability", *SSRI Research Report 75-7*, Social Science Research Institute, University of Southern California.

**Norlen, U.** (1976) *Simulation Model Building: a Statistical Approach to Modeling in the Social Sciences with the Simulation Method*, Halsted Press, New York.

**O'Connor, M.F.** "The Application of Multi-Attribute Scaling Techniques to the Development of Indices of Water Quality", Doctoral Dissertation, The University of Michigan.

**O'Keefe, R.M., O.Balci and E.Smith** (1987) "Validating Expert System Performance" *IEEE Expert*, vol.2(1987):81-90.

**O'Leary, D.E.** (1987) "Validation of Expert Systems—with Application to Auditing and Accounting Expert Systems", *Decision Sciences*, vol.18:468-486.

**O'Leary, D.E. and N.A.Kandelin** (1988) "Validating the Weights in Rule-Based Expert Systems: A Statistical Approach", *International Journal of Expert Systems*, 1(3):253-279.

**Orchard-Hayes, W.** (1976) "On Complexity in the Use of Computers and Some Recent Experience in Mastering It", *Survey of Mathematical Programming, Proceedings of the 9th International Mathematical Programming Symposium*, Budapest, August 23-27, 1976, vol.3, A. Prékopa, *ed.*, pgs.399-409.

**Orden, A.** (1979) "The Effectiveness of Linear Programming Models in Operations Planning", Paper presented at the 10th International Symposium on Mathematical Programming, Montreal, August 27-31, 1979.

**Oxman, S.W.** (1991) "Reporting Chemical Spills: An Expert Solution", *AI Expert*, May, 1991, pgs.50-51.

**Packard, N.H.** (1985) "Complexity of Growing Patterns in Cellular Automata", *Dynamical Systems and Cellular Automata*, Academic Press, London, pgs.123-137.

**Pai, G.K., D.H. Gustafson and G.W. Kiner** (1971) "Comparison of Three Non-risk Methods for Determining a Preference Function", University of Wisconsin, January, 1971.

**Pappas, R.A. and D.S.Remer** (1984) "Status of Corporate Planning Models", *Managerial Planning*, March/April, 1984, 32(5):4-16.

**Percy, L.** (1981) "Using Qualitative Focus Groups in Generating Hypotheses for Subsequent Quantitative Validation and Strategy Development", pgs.57-61.

**Pedhazur, E.J.** (1982) *Multiple regression in Behavioral Research: Explanation and Prediction, 2nd edition*, Holt, Rinehart & Winston.

**Plant, R. and A.Preece** (1996) "Editorial: Special Issue on Verification and Validation", *International Journal of Human-Computer Studies*, 44(2):123-125.

**Platt, J.R.** (1964) "Strong Inference" *Science*, 16 October, 1964, 146(3642):347-353.

**Preece, Alun D.** (1990) "Towards a methodology for evaluating expert systems", *Expert Systems*, 7(4):215-222.

**Primoff, E.S.** (1959) "Empirical Validation of the J-Coefficient", *Personnel Psychology*, vol 12, pgs.413-418.

**Pollack,I.** (1964) "Action Selection and Yntema-Torgerson Worth Function", *Information Science and Engineering: Proceedings of the First Congress of the Information Systems Sciences*, New York: McGraw-Hill.

**Popper, K.R.** (1963) *Conjectures and Refutations*, Routledge and Kegan Paul, London, 413 pages.

**Popper, K.R.** (1977) *The Logic of Scientific Discovery*, Hutchinson of London, 480 pages.

**Popper, K.R.** (1978) *Objective Knowledge*, Oxford University Press, Oxford, 395 pages.

**Recknagel, F. and J.Benndorf** (1982) "Validation of the Ecological Simulation Model "SALMO"", *International Revue ges. Hydrobiology*, 67(1):113-125.

**Rich, E.** (1983) *Artificial Intelligence*, McGraw-Hill, Inc., New York, 436 pages.

**Richardson, J.M., Jr.** (1982) "Global Modeling–2: Where to now?", *Futures*, 10(6):476-491.

**Richer, M.H. and W.J.Clancey** (1985) "Guidon-Watch: A Graphic Interface for Viewing a Knowledge Based System", *IEEE Computer Graphics and Applications*, November, 1985, pgs.51-64.

**Romesburg, H.C.** (1984) *Cluster Analysis for Researchers*, Lifetime Learning Publications, Belmont, California, 334 pages.

**Rosen, R.** (1980) "On Information and Complexity (Chapter 7)", *Complexity, Language and Life: Mathematical Approaches*, J.L. Casti & A. Karlqvist, *eds.*, Springer-Verlag, pgs.174-196.

**SAS Institute Inc.** (1985) *SAS User's Guide: Statistics, Version 5 Edition*, Cary, NC: SAS Institute Inc., 956 pages.

**Sacerdoti, E.D.** (1991) "Managing Expert-System Development", *AI Expert*, May, 1991, pgs.26-33.

**Saunders, C.** (1985) "Model Validation: The Missing Process", *Journal of Systems Management*, August, 1985, 36(8):26-29.

**Scarl, E.A., J.R.Jamieson and C.I.Delaune** (1987), "Diagnosis and Sensor Validation through Knowledge of Structure and Function", *IEEE Transactions on Systems, Man & Cybernetics*, SMC-17(3):360.

**Schruben, L.W.** (1980) "Establishing Credibility of Simulations", *Simulation*, March, 1980, pgs.101-105.

**Searle, S.R.** (1971) *Linear Models*, John Wiley & Sons, Inc., New York, 532 pages.

**Sell, P.S.** (1985) *Expert Systems: A Practical Introduction*, "Chapter 6: Validation", MacMillan Publishers Ltd.

**Shannon, R.E.** (1975) *Systems Simulation, The Art and Science*, Prentice-Hall, Englewood Cliffs, NJ.

**Shapiro, E.** (9184) "Alternation and the Computational Complexity of Logic Programs", *Journal of Logic Programming*, vol.1, pgs.19-33.

**Shechter, M. and R.C.Lucas** "Validating a Large Scale Simulation Model of Wilderness Recreational Travel", *Interfaces*, October, 1980, 10(5):11-18

**Shooman, M.L.** (1983) *Software Engineering*, McGraw-Hill, New York.

**Simon, H. A.** (1977) "The Logic of Heuristic Decision Making", *Models of Discovery and Other Topics in the Methods of Science*, D. Redel Publishing Company, Boston.

**Simon, Herbert A.** (1982) *Models of Bounded Rationality: Behavioral Economics and Business Organization (Vol.2)*, MIT Press, Cambridge, MA.

**Stauffer, H. Jr.** (1980) "Developing, Improving and Assessing the ICF Coal and Electric Utilities Model", *Validation and Assessment Issues of Energy Models*, Proceedings of a Workshop (NSB Special Publication, 569, 1980), S.I. Gass. *ed.*, pgs.141-151.

**Straub, D.W. and C.L.Carlson** (1989) "Validating Instruments in MIS Research", *MIS Quarterly*, June, 1989:147-169.

**Sterba, H., M.Moser and R.Monserud** (1995) "Prognaus — Ein Waldwachstumssimulator für Rein– und Mischbestände", *Österreichische Forstzeitung*, 103(5):19-20.

**Stewart, D.A. and B.W.Surgenor** (1987) "Simulator Validation of an Expert System for Process Fault Diagnosis", J.Q.B. Chou *ed.*, Society for Computer Simulation, 19th Annual Summer Computer Simulation Conference, 27-30 July, 1987, Montreal, Canada, pgs.663-667.

**Tao, Y., H.Zhijun and Y.Ruizhao** (1987) "Performance Evaluation of the Inference Structure in Expert Systems", *Natural Language, Proceedings, IJCAI 1987*, International Joint Conference on Artificial Intelligence, pgs.945-950.

**Tchebakova, N.M., R.A.Monserud and D.I.Nazimova** (1994) "A Siberian Vegetation Model Based on Climatic Parameters", *Canadian Journal of Forestry Research*, vol.24, pgs.1597-1607.

**Turing, A.M.** (1950) "Computing Machinery and Intelligence", *Mind LIX*, no.2236, Oct, 1950, Oxford University Press; reprinted with permission in *The Philosophy of Artificial Intelligence*, M.A. Boden, *ed.*, Oxford University Press, 1990, pgs.41-66.

**Turing, A.M.** (1963) "Computing Machinery and Intelligence", *Computers and Thought*, E.A. Feigenbaum & J. Feldman, *eds.*, McGraw-Hill, New York.

**Valluy, L., P.Nykanen, J.Rantanen, J.Salmela, P.Gronroos and J.Lumio** (1989) "An Approach to Combine Knowledge Validation & Evaluation with Development in Medical AI", Proceedings of the Scandinavian Conference on Artificial Intelligence (SCAI '89), pgs.753-764.

**van Heijenoort, J.** (1966) *From Frege to Gödel, Source Book on Mathematical Logic, 1879-1931*, Cambridge, Mass.

**van Heijenoort, J.** (1972) "Gödel's Theorem", *The Encyclopedia of Philosophy*, P. Edwards, *ed.*, pgs.348-357.

**van Horn, R.L.** (1971) "Validation of Simulation Results", *Management Science*, 17(5), pgs.247-258.

**Vick, S. and K.Lindenmayer** (1988) "Verification and Validation of Rulebased Systems for Hubble Space Telescope Ground Support", *Telematics and Informatics*, 5(3):313-323.

**von Winterfeldt, D.** (1971) *Multi-attribute Utility Theory: Theoretical Background and Experimental Validation*, The University of Michigan.

**Weizenbaum, J.** (1976) *Computer Power and Human Reason*, W.H. Freeman and Company, San Francisco, 300 pages.

**Whitehead, A.M. and B.A.W.Russel** (1910-1913) *Principia Mathematica (3 vols.)*, Cambridge University Press, UK.

**Wittgenstein, L.** (1922) *Tractatus logico-philosophicus Logisch-Philosophische Abhandlung*, Edition Suhrkamp (1980), Basil Blackwell, Oxford, 114 pages.

**Wittgenstein, L.** (1953) *Philosophische Unterschungen*, G.E.M. Anscombe, R. Rhees, G.H. von Wright, *eds.*, Oxford (in German with English translation on facing pages).

**Yntema, D.B. and L. Klem** (1965) "Telling a Computer How to Evaluate Multi-Dimensional Situations", *IEEE Transactions on human Factors in Electronics*, HFE-6, pgs. 3-13.

**Yu, V., L.Fagan, S.Wraith, W.Clancey, A.Scott, J.Hannigan, R.Blum, B.Buchanan, and S.Cohen** (1979) "Antimicrobial Selection by Computer: A Blinded Evaluation by Infectious Disease Experts" *Journal of the American Medical Association*, vol. 242, pgs.1279-1282.

**Ziegler, B.P.** (1976) *Theory of Modelling and Simulation*, John Wiley & Sons, New York, 435 pages.

# APPENDICES

## Appendix A: Spectrum of Validation Methods

### App. A.1: Technical Validation Methods

| Validation METHODS | Definition of Validation METHODS | DOMAIN |
|---|---|---|
| **T-1. TRADITIONAL VERIFICATION** | | |
| COMPILATION & EXECUTION | *Initial test to insure that the model can be made to run on demand* | |
| VERIFICATION | *Determine if formal (computer) model is constructed as intended, with all pertinent variables and relationships* | |
| VERIFICATION | *Demonstrating that a computer program "runs as intended"* | |
| VERIFICATION | *Comparing program code with verbal/ written program specifications and formulas* | |
| VERIFICATION | *Extensive effort to assure that the equations defining system performance & their implementation via a computer program are correct* | Police Patrols Green & Kolesar 1989 |
| STATIC ANALYSIS | *Determine consistency & completeness of a model; e.g. in a knowledge base, check structure, rules and attributes* | Medicine Valluy *et al.*, 1989 |

| Validation METHODS | Definition of Validation METHODS | DOMAIN |
|---|---|---|
| **T-2. MATHEMATICAL TREATMENTS** | | |
| MATHEMATICAL Validity | *Determine if mathematical and numerical calculations are correct & accurate; analyse logical flow of data; check for missing variables & relationships–e.g., VERIFICATION* | |
| FORMAL PROOFS of CORRECTNESS | *Proving that the expected results are as required for all valid inputs* | Defense: Aircraft Communications SW Davis, 1989 |
| CHECKING OPERATIONAL VARIABLES | *Confirming that given values of input produce desired outputs when the model is run* | |
| LOGIC MODEL Validation | *Includes: CONSISTENCY: Showing that a system produces similar answers to similar questions; i.e., that the same inputs result in the same outputs COMPLETENESS: Show that all feasible outcomes can be derived, & all acceptable input will produce an outcome; i.e., everything that is true is derivable SOUNDNESS: Show that everything that is derivable is true ACCURACY: A measure of systematic bias PRECISION: A measure of random bias DEFINITION: Removing relational ambiguities by defining variables* | Expert Systems Sell, 1985; Mathematical Models Finlay *et al.*, 1988 |

| Validation METHODS | Definition of Validation METHODS | DOMAIN |
|---|---|---|
| **T-3. STATISTICAL TREATMENTS** | | |
| QUANTITATIVE Validation | *Using statistical techniques to measure consistency; either (1) confidence intervals, or (2) hypothesis testing* | Simulation Models<br>Balci & Sargent, 1983 & 1984 |
| STATISTICAL CONCLUSION Validity | *Assess the mathematical relationships between variables & the likelihood that this provides a correct picture of the true covariation between variables; determine statistical power, i.e., the probability that a null hypothesis has correctly been rejected* | Behavioral Science<br>Cohen, 1977 |
| | | MIS Research<br>Baroudi & Orlikowski, 1989 |
| | | Computer Simulation<br>Naylor *et al.*, 1966 |
| | | Digital Simulation<br>Fishman & Kiviat, 1967 |
| | | Simulation<br>Mihram, 1972<br>Fishman, 1973<br>Shannon, 1975 |
| | | Social Science<br>Norlen, 1976 |

| Validation METHODS | Definition of Validation METHODS | DOMAIN |
|---|---|---|
| **T-3.** (*cont'd*) | | |
| STATISTICAL Tests | *Determine model stability, sensitivity or predictability using tests, e.g.: Theil's Inequality Coefficient, Janus Quotient & Index of Validity* | |
| | *e.g.: Chi-squared tests; Cohen's kappa* | Psychology<br>Light, 1971<br>Cohen, 1968 |
| | *e.g.: Pearson correlation analysis* | Job Performance<br>Hollenbeck & Whitener, 1988 |
| | | Missile Systems (Simulation)<br>Kheir & Holmes, 1978 |
| **T-4. COMPLETE MODEL CHECKING** | | |
| ANALYTICAL | *Checking out each part of the model* | |
| EXHAUSTIVE Validation | *Test exhaustively all assertions; e.g., define& traverse all possible paths in a in a knowledge base* | |

| Validation METHODS | Definition of Validation METHODS | DOMAIN |
|---|---|---|
| **T-5. TESTING REPRESENTATIVE PARTS of the MODEL** | | |
| SYNTHETIC Validity | *Infer the validity of a whole by breaking up the whole into its logical elements, & determining the validity of the elements* | Personnel Selection<br>Primoff, 1959 |
| | | Personnel<br>Guion, 1965 |
| | | Job Analysis<br>McCormick *et al.*, 1979 |
| TRACING | *Following the behavior of specific entities through the model calculation* | |
| SELECTIVE Validation | *System is structured (e.g., in independent modules) so that selected tests are representative of the entire system* | Manufacturing<br>Enand *et al.*, 1990 |
| **T-6. MODEL SENSITIVITY** | | |
| INTERNAL Validation | *Stochastic analysis to determine the variability of the model* | Leadership & Management<br>McCall & Lombardo, 1982 |
| | | Gov't Model Evaluation<br>Gruhl, 1982 |
| INTERNAL Validity | *Repeated runs with constant input to show no change in output (if model is deterministic) or output variance is acceptably low (if model is stochastic)* | |
| INTERNAL Validation | *Determine whether observations/outputs could have been caused by or correlated with unhypothesized &/or unmeasured variables; i.e., rival explanations not incorporated into the model* | Social Science<br>Cook & Campbell, 1979 |

| Validation METHODS | Definition of Validation METHODS | DOMAIN |
|---|---|---|
| **T-6.** (*cont'd*) | | |
| SENSITIVITY Analysis | *Systematically change input parameters to see how output changes. Compare to other models re both structure and output* | Energy Modeling Baughman, 1980 |
| | | Decision Analysis Cazalet, 1980 |
| | | Regional Blood Supply Planning Cumming, 1976 |
| | | Management Eilon, 1974 |
| | | Decision Making (Management) Henderson & Nutt, 1980 |
| | | Global Modeling Richardson, 1978 |
| SENSITIVITY Analysis | *Test for stability of the model: compare the model "against itself" for sensitivity to slight changes in data, weights, knowledge, etc,* | Medical (MYCIN) Buchanan & Shortliffe, 1984 |
| | | World Iron & Steel Economy Hashimoto, 1983 |
| | | Wilderness Recreation Shechter & Lucas, 1980 |
| | | Water Ecosystems Rechnagel & Benndorf, 1982 |
| SENSITIVITY Testing | *Examine how the model reacts to violations of its key assumptions* | Police Patrols Green & Kolesar, 1989 |
| SENSITIVITY Analysis | *Alter input values to determine effect on output; compare test output to historical data* | |

| Validation METHODS | Definition of Validation METHODS | DOMAIN |
|---|---|---|
| **T-6.** (*cont'd*) | | |
| VARIABLE-PARAMETER Validity | *Compare model's variables, parameters & outputs with assumed counterparts in the real world* | |
| **T-7. PHILOSOPHICAL** | | |
| INDUCTION | *Proving definitively the truth of a theory; Popper (1965) has shown this to be impossible* | |
| REFUTATION | *Increasing confidence in a theory (model) after failures of increasingly severe attempts to disprove it* | |
| **T-8. COMPARISONS WITH OTHER MODELS or IDEAS** | | |
| EMPIRICAL Validation | *Comparing a model's results with the results of other models or measures of the same situation, but which are structured entirely differently from the model in question* | Risk Analysis (Equity Securities) Falk & Gordon, 1978 |
| | | Regional Blood Supply Planning Cumming, 1976 |
| | | Strategic Planning (Financial) Gale, 1978 |

| Validation METHODS | Definition of Validation METHODS | DOMAIN |
|---|---|---|
| **T-8.** (*cont'd*) | | |
| SIMULATION MODELS | *Compare results of simulation runs with results of simpler (e.g., analytic) models to determine if the simpler, more economical model may be safely used to describe system behavior* | Fire Fighting<br>Carter & Ignall, 1970 |
| | | Fire Fighting<br>Chaiken, 1971 |
| | | Police Patrols<br>Larson, 1972 |
| | | Fire Fighting<br>Ignall & Urbach, 1975 |
| | | Telephone Traffic Routing<br>Chlebus, 1991 |
| | | Fault Diagnosis<br>Stewart & Surgenor, 1987 |
| EXPERT SYSTEM | *Construct knowledge-based computer code to mimic structure & function of a real system, and compare outputs* | Diagnostic Fault Location<br>Scarl *et al.* 1987 |
| MODEL SIGNIFICANCE | *Determine if estimates produced by a model differ substantially from those produced by the model it is intended to replace* | Police Patrols<br>Cobham, 1954<br>Green & Kolesar 1984 |
| INDEPENDENT CORROBORA-TION | *Confirmation of the results of a model from an unrelated, & typically unanticipated source (e.g., another discipline, industry or country)* | Linguistics<br>Greenberg & Ruhlen, 1992 |

| Validation METHODS | Definition of Validation METHODS | DOMAIN |
|---|---|---|
| **T-9. REDUNDANT MODEL CREATION** | | |
| DUAL PROGRAMMING | *Coding two or more implementations of a set of system requirements, & then comparing their results (i.e., a test "Oracle")* | Defense: Aircraft Communications SW<br>Davis, 1989 |
| TRIPLE REDUNDANCY | *Employing different analysts, programmers, algorithms, computer languages for parallel design with the same input/output objectives* | Chemical Processes<br>Miller, 1989 |
| | | Fire Detection<br>Luck & Schlossarek, 1987 |
| **T-10. MODELING LANGUAGE** | | |
| LOGICAL Validation | *Evaluate the appropriateness of the modeling language to determine if it has captured the richness of the conceptual model* | |
| EXPERIMENTAL Validation | *Determine quality and efficiency of the solution method (e.g., algorithmic, heuristic), including parameter sensitivity* | |
| **T-11. CLIENT ACCEPTANCE** | | |
| FORMAL ACCEPTANCE TESTING | *Sponsor or End-User formally signs off, accepting the technicians'/programmers' assurances regarding the quality of all decisions made by the system, as a result of technical testing* | Expert Systems<br>Gaschnig *et al.*, 1983 |
| | | Software Engineering<br>Shooman, 1983 |

App. A.2: Semi-Technical Validation Methods

| Validation METHODS | Definition of Validation METHODS | DOMAIN |
|---|---|---|
| **S-1. MODEL STRUCTURE & DATA** | | |
| MODEL Validity | *Identify all stated & implied assumptions, variables & hypothesized relationships between variables* | |
| DATA Validation | *Determine sufficiency, accuracy, appropriateness & availability of data, within acceptable cost limits* | |
| RAW DATA Validity | *Measuring/determining if data are accurate, impartial and representative* | |
| STRUCTURED DATA Validity | *Review each step of model manipulation; i.e., verification with interpretation* | |
| **S-2. TOTAL MODEL ANALYSIS** | | |
| RIGOROUS HAND-ANALYSIS | *Analysing detailed printouts of model-generated scenarios to determine the accuracy & reasonableness of each result* | Defense: Aircraft Communications SW Davis, 1989 |
| PROCEDURAL Validation | *Review each path possible, to ensure that the system always reaches acceptable conclusions* | Manufacturing Enand *et al.*, 1990 |
| "BOTTOM UP" Validity | *Determine if all aspects of the real-life system being modeled are included in full detail* | Publishing Forman, 1978 |
| SUBSYSTEM Validation | *Users observe performance of each subsystem, & then estimate total system performance* | Decision Theory Langlotz *et al.*, 1986 |
| | | Missile Systems (Simulation) Kheir & Holmes, 1978 |
| SYNOPTIC Validation | *Checking that an acceptable output is achieved for each set of inputs and thus total model performance is acceptable* | |

## Appendix A.2 - *page 2*

| Validation METHODS | Definition of Validation METHODS | DOMAIN |
|---|---|---|
| **S-3. REPRESENTATIONS ANALYSIS** | | |
| CONSTRUCT Validity | *Determine if measures (variables, etc.) chosen actually describe the event/system of interest, or if they are merely artifacts of the methodology used Includes: CONVERGENT Validity DISCRIMINANT Validity CONCURRENT Validity PREDICTIVE Validity* | Psychology Campbell & Fiske, 1959 |
| | | Educational Testing Cronbach, 1971 |
| | | Military Miller *et al.*, 1967 |
| CONVERGENT Validity | *Correlation of the same trait & varying measurement methods is significantly different from zero* | Psychology Campbell & Fiske, 1959 |
| | | Engineering Pollack, 1964 |
| | | Decision Making (Electronics) Yntema & Klem, 1965 |
| | | Decision Making (Management) Eckenrode, 1965 |
| | | Utility Models Hoepfl & Huber, 1970 |
| | | Decision Making Pai *et al.*, 1971 |
| | | Banking (Loans) Kennedy, 1971 |
| | | Utility Models (Job Preference) von Winterfeldt, 1971 |
| | | Water Quality O'Connor, 1972 |
| | | Utility Theory Newman, 1975 |
| | | Job Preference Fischer, 1977 |

## **Appendix A.2** - *page 3*

| Validation METHODS | Definition of Validation METHODS | DOMAIN |
|---|---|---|
| **S-3.** *cont'd* | | |
| DISCRIMINANT Validity | *Evidence that traits are correlated with other traits by using both the same and different analysis methods* | Psychology<br>Campbell & Fiske, 1959 |
| CONCURRENT Validity | *(subsumed in CONSTRUCT Validity, S-3.)* | Psychology<br>Cronbach & Meehl, 1955 |
| PREDICTIVE Validity | *(subsumed in CONSTRUCT Validity, S-3.)* | Organizational Research<br>Mitchell, 1985 |
| INSTRUMENT Validation | *Determine extent to which instruments (models) accurately measure/represent the directly unobserveable events that they are supposed to measure or represent*<br>*Includes: CONTENT Validity*<br>*CONSTRUCT Validity*<br>*MEASUREMENT ACCURACY* | MIS Research<br>Straub, 1989 |

| Validation METHODS | Definition of Validation METHODS | DOMAIN |
|---|---|---|
| **S-4. REPLICATING the PAST** | | |
| RETROSPECTIVE SCENARIO ANALYSIS | *Using scientific and practical scenarios that test a model's replicative validity* | Water Ecosystems Recknagel & Benndorf, 1982 |
| HISTORICAL Validation | *Using selected historical data to determine if the model behaves as the system did* | Inventory Baglow, 1977 |
| | | Leadership & Management McCall & Lombardo, 1982 |
| | | Global Modeling Richardson, 1978 |
| | | World Iron & Steel Economy Hashimito, 1983 |
| | | Wilderness Recreation Shechter & Lucas, 1980 |
| | | Semiconductor Manufacturing Elleby *et al.*, 1987 |
| | | Vegetation: Climate Change Monserud *et al.*, 1993 |
| REPLICATIVE Validity | *Comparing model's data to data already acquired from the real system; e.g., data about past human performance* | Semiconductor Manufacturing Elleby *et al.*, 1987 |
| REPLICATIVE Validity | *Determine to what extent data produced by a model match data already produced by a real-world system* | |

| Validation METHODS | Definition of Validation METHODS | DOMAIN |
|---|---|---|
| **S-5. COMPARISONS to the "REAL WORLD"** | | |
| EVENTS Validation | *Comparing the events occurances in the model with the distribution of those in the system* | Regional Blood Suppliers **Cumming *et al.*, 1976** |
| SPECTRAL Analysis | *To evaluate if, in the frequency domain, the dynamic behaviour of the model differs from the behaviour of the system* | Energy **Baughman, 1980** |
| | | Management (simulation) **Fox, 1978** |
| | | Operations Research **Gruhl, 1982** |
| | | Leadership & Management **McCall & Lombardo, 1982** |
| | | Coal & Electric Utilities **Stauffer, 1980** |
| | | Management (simulation) **van Horn, 1971** |

| Validation METHODS | Definition of Validation METHODS | DOMAIN |
|---|---|---|
| **S-5.** *(cont'd)* | | |
| STRUCTURAL Validity | *If the model not only reproduces behaviour observed in a real system, but also truly reflects the way in which the real system operates to produce this behavior* | Simulation<br>Ziegler, 1976 |
| STRUCTURAL Validity | *Check whether the structure of the model is an adequate representation of the real structure; i.e., (1) empirical comparison of model equations with real system relationships, & (2) theoretical comparison of model equations with the available theory* | Systems Dynamics<br>Barlas, 1989<br>Strategic Planning (Financial)<br>Gale, 1978 |
| STRUCTURAL Validity | *Examination of key assumptions of the model; asking, in effect, "why it works"* | Police Patrols<br>Green & Kolesar, 1989 |
| PROGRAM PROVING | *Test to see if model's structure and logic correspond in a one-to-one way with the structure of the system being modelled* | Discrete Event Simulation<br>Cutler, 1980 |
| BEHAVIOR Validity | *Check if the model is capable of producing an acceptable output behavior* | |
| PATTERN PREDICTION Testing | *Determine whether behavior patterns generated by the model are close enough to the major patterns exhibited by the real system* | Systems Dynamics<br>Barlas, 1989 |

| Validation METHODS | Definition of Validation METHODS | DOMAIN |
|---|---|---|
| **S-6. PREDICTING the FUTURE** | | |
| PROSPECTIVE SCENARIO ANALYSIS | *Using scientific & practical scenarios scenarios that test a model's predictive validity* | Water Ecosystems Recknagel & Benndorf, 1975 |
| PREDICTIVE Validation | *Determining if the system's behaviour & the model predictions are the same* | Inventory Baglow, 1977 |
| | | Regional Blood Suppliers Cumming *et al.*, 1976 |
| | | Energy Models Mayer, 1980 |
| PREDICTIVE Validity | *Analysis of errors (& magnitudes of errors) between actual outcomes & predicted outcomes for a model's components & relationships* | |
| PREDICTIVE Validation | *Compare test cases with known results, or human expert performance on the same cases* | Medical (chest pain analysis) Hudson *et al.*, 1984 |
| PREDICTIVE Validity | *Determine to what extent data produced by a model match data subsequently produced by a real system; i.e., the reverse of replicative validity (a more stringent test than replicative validity)* | Strategic Planning (Financial) Gale, 1978 |
| | | Wilderness Recreation Shechter & Lucas, 1980 |
| BLACK BOX "OBJECTIVE" Validation | *Testing the relationship between a given input and the resultant output* | 22 Industrial/ Commercial DSS Finlay & Wilson, 1991 |
| OUTPUT Validity | *Viewing model as a "black box" to be judged only on whether given reasonable estimates of system parameters, it will produce reasonable estimates of system performance* | Police Patrols Green & Kolesar, 1989 |

| Validation METHODS | Definition of Validation METHODS | DOMAIN |
|---|---|---|
| **S-7. INPUT/OUTPUT COMPARISONS** | | |
| DOMAIN Validation | *Laboratory simulation of as much input as possible, in order to remove obvious discrepancies between an expert's view of a domain & reality; (i.e., reconciling input with expected output)* | Manufacturing<br>Enand *et al.*, 1990 |
| VISUAL INTERACTIVE INTERACTION | *Computer graphic animation of model parameters, rules, etc., allowing human experts to test or change the model's reasoning process* | Hazard Risk Assessment<br>Fedra *et al.* 1987, 1988 |
| | | Operational Research<br>Bell, 1985 |
| | | Expert Systems<br>Richer & Clancey, 1985 |
| EXPERIMEN-TATION | *Manipulating variables in both real-world & the model; then comparing the outputs* | |

| Validation METHODS | Definition of Validation METHODS | DOMAIN |
|---|---|---|
| **S-8. PERFORMANCE COMPARISONS** | | |
| PHYSICAL Validity | *Demonstrate that model's results work in the real-world environment* | Engineering Orden, 1979 |
| IMPLEMENTATION Validity | *Determine extent to which the real world system being modeled responds as indicated by the model's recommended solution* | |
| OPERATIONAL Validity | *Assess the importance of errors found in other phases of validation; conclude whether the model is appropriate, given the errors found; Also determine robustness (i.e., if the model protects users from unknowingly producing impossible or absurd results); Also determine if model produces reasonable results given costs/benefits involved* | |
| QUALITATIVE Validation | *Formal, subjective comparisons of performance*<br>*Includes: FACE Validation*<br>*PREDICTIVE Validation*<br>*TURING Tests*<br>*FIELD Tests*<br>*SUBSYSTEM Validation*<br>*SENSITIVITY Analysis*<br>*VISUAL INTERACTION* | |

| Validation METHODS | Definition of Validation METHODS | DOMAIN |
|---|---|---|
| **S-9. MODEL MAINTENANCE** | | |
| DYNAMIC Validity | *Establish model life cycle maintenance, i.e., procedures for reviewing and updating model's parameters, structures & assumptions* | |
| PROTOTYPING | *Incremental methodology; detailed specification or functional definition can't be written before coding.* | Expert Systems<br>**Sacerdoti, 1991** |
| LIFECYCLE Validation | *Spectrum of system development, from initial 'design for testing' phase to final decision whether the system should be abandoned* | Nuclear Power Plants<br>**Groundwater, 1987** |
| | | Business Risk Assessment<br>**Moore, 1983** |

## App. A.3: Human Judgment Validation Methods

| Validation METHODS | Definition of Validation METHODS | DOMAIN |
|---|---|---|
| **H-1. FIRST IMPRESSIONS** | | |
| FACE Validity | *Gathering of opinion about the reasonableness & accuracy of the model from people knowledgeable about the system* | Leadership & Management McCall & Lombardo, 1982 |
| | | Wilderness Recreation Shecter & Lucas, 1980 |
| POSITIVE INITIAL REACTION | *If initial reception of the model by decision makers who will use the model is that it is realistic and credible* | |
| BLACK BOX "SUBJECTIVE" Validation | *Looking at results & judging whether they appear reasonable or not* | Chemical Spills Oxman, 1991 |
| | | 25 Industrial/ Commercial DSS (Operations Mgmt.) Finlay & Wilson, 1987 |
| | | Life Insurance Long & Neale, 1990 |

| Validation METHODS | Definition of Validation METHODS | DOMAIN |
|---|---|---|
| **H-2. EXPERT OPINIONS** | | |
| DELPHI TECHNIQUE | *Multi-round anonymous feedback analysis exercise in which opinions of several experts are expected to eventually converge; use the experts' consensus to evaluate output from a model* | Electro-cardiograms Kors *et al.*, 1990 |
| CONTENT Validity | *Determining what a system knows, does not know, or knows incorrectly, by experts who either examine the rule base directly, or perform sample tests* | Behavioral Research Kerlinger, 1973 |
| | | Medicine Valluy *et al.* 1989 |
| CONTENT Validity | *Assure that a high degree of realism is built into the model via (1) postulated relationships among variables & parameters, & (2) reliability of the database, by involving users in model construction from conceptualization to implementation* | Wilderness Recreation Shechter & Lucas, 1980 |
| CONTENT Validity | *Experts familiar with the content universe evaluate versions of the instrument (model) again and again until a form of consensus is reached* | Educational Testing Cronbach, 1971 |

| Validation METHODS | Definition of Validation METHODS | DOMAIN |
|---|---|---|
| **H-3. COMPARISONS to EXPERT SOURCES** | | |
| TURING Tests | *To verify if knowledgeable people can distinguish between a real system & output of a model of the system* | Medicine Yu *et al.*, 1979 |
| TURING Tests | *"Blind" comparison of computer with human experts; then use statistical tests for variability between model & human experts, & for consistency among human experts* | Chemotherapy Hickam *et al.*, 1985 |
| | | Business (Scheduling) Schruben, 1980 |
| CRITERION Validity | *Ascertain the system's level of expertise by comparing system decisions to decisions made by human experts* | Manufacturing Production Scheduling Kerr & Ebsary, 1988 |
| SCENARIOS | *Presenting experts with textual descriptions of problems; analyse for congruence with model recommendations* | Strategic Decision Making (Finance) Derkinderen & Crum, 1988 |
| CONVERGENT Validation | *Comparing the model's predictions with those of experts* | Psychology Campbell & Fiske, 1959 |
| INTRA-SUBJECTIVE CONSISTENCY | *Degree to which two or more experts using the same information produce results that are similar to results produced by a model* | Life Insurance Long & Neale, 1990 |
| PERFORMANCE Validation | *Specify acceptable level or range of output, usually with regard to the performance of the human experts that the system models* | |
| LITERATURE Comparison | *Compare system output with selected cases from appropriate industry publications* | Medicine Buchanan & Shortliffe, 1984 |

| Validation METHODS | Definition of Validation METHODS | DOMAIN |
|---|---|---|
| **H-4a. USER PARTICIPATION - Initial** | | |
| FOCUS GROUPS | *Group discussion designed to result in a model or hypothesis that can then be tested quantitatively* | Marketing Strategy (Advertising) Percy, 1981 |
| USER INVOLVEMENT | *Involve users in model construction from conceptualization to implementation; a necessary condition for Content Validity* | Wilderness Recreation Shechter & Lucas, 1980 |
| STRUCTURED WALK-THROUGH | *Formal presentation of a model's assumptions to key people, eliciting discussion and allowing corrections and changes to the conceptual model before coding begins* | Manufacturing Law & McComas, 1990 |
| **H-4b. USER PARTICIPATION - - Model Development** | | |
| CONCEPTUAL Validation | *Determine the degree of relevance of assumptions & theories underlying the conceptual model of the problem situation, from the point of view of the intended users* | |
| RATIONALISM | *Does the model "make sense"? Is the model's theoretical structure clear & does it inspire confidence in the intended users?* | Strategic Business Planning Naylor, 1978 |
| MEASUREMENT ACCURACY (Reliability) | *Extent to which an instrument/model is unambiguous in eliciting response from users; i.e., if respondents do not misunderstand questions, instructions, etc.* | Psychometrics Cronbach, 1951 |

| Validation METHODS | Definition of Validation METHODS | DOMAIN |
|---|---|---|
| **H-4c. USER PARTICIPATION - Testing** | | |
| PERIODIC INFORMAL Validation | *On-going review & testing of systems during development; using rapid prototyping to elicit feedback from users* | |
| FIELD TESTS | *User testing of prototype systems in actual application environments, one purpose of which is to elicit and incorporate user critiques in the final model design* | Computer System Configuration Bachant & McDermott, 1984 |
| | | Strategic Planning (Financial) Gale, 1978 |
| | | Refrigeration System Maintenance Cochran & Hutchins, 1987 |
| OPERATIONAL Validation | *Determine quality and applicability of solutions & recommendations with respect to the intended users* | |
| "WHITE BOX" TESTING | *(1) Using a hard copy of the program code, read through the rules to determine if they are correct; (2) Trace a set of inputs manually through the knowledge base* | Chemical Spills Oxman, 1991 |

| Validation METHODS | Definition of Validation METHODS | DOMAIN |
|---|---|---|
| **H-5. THEORETICAL** | | |
| NONFORMAL TEST INTERROGATOR | *Extent to which the expert/subject (e.g., an engineer) finds the model useful for understanding & thereby improving upon past behavior* | Engineering Mitroff, 1969 |
| HYPOTHESIS Validity | *Determine if higher-level relationships in the model correspond with similiar relationships in the real world* | International Politics Herman, 1967 |
| | | Simulation Emshoff & Sisson, 1970 |
| HYPOTHESIS Testing | *Study model relationships to determine if they interact as intended by the modeler* | Ecology Mankin *et al.*, 1975 |
| | | Behavioral Science Kirk, 1968 |
| STRONG INFERENCE | *Iterative process (proposed by Platt, 1964) of testing theoretical models: develop alternative hypotheses, perform experiments to eliminate at least one of the hypotheses & refine the theory, usually by adding new hypotheses to test* | Ecology Caswell, 1976 |
| **H-6. CLIENT JUDGMENT** | | |
| FAITH-in-the MODELER | *Manager/client has enough faith in the ability of the modeler that the manager relies on the modeler to say that the model "works"; Modelers themselves are essentially validated by virtue of being chosen to do the work, i.e, on the basis of their reputations* | 13 Industrial/ Commercial DSS (Opns. Mgmt.) Finlay & Wilson, 1987 |

| Validation METHODS | Definition of Validation METHODS | DOMAIN |
|---|---|---|
| **H-7. ECONOMIC EFFECTS** | | |
| ECONOMIC (COST-BENEFIT) Analysis | *Estimate & compare costs of validation (e.g., time, extent, staff) with anticipated economic benefits from the ultimate use of the system* | |
| ECONOMIC FEASIBILITY | *Query users to determine if using the model will justify the expense of setting up and prototyping the model in real circumstances* | Regional Blood Supply Planning Cumming *et al.*, 1976 |
| PRACTICAL Validation | *Recognizing that models built for use in business are never fully validated in any truly technical sense; replacing technical validation terminology with terms such as "reliability", "stability" or "accuracy", which are more appropriate in the business environment* | |

## Appendix B.1: Validation Methodologies – Historical

| Validation Group Description | Specific Examples | Years of Publication |
|---|---|---|
| **TECHNICAL VALIDATION** | | |
| Traditional Verification | 1 | 1989 |
| Mathematical Treatments | 0 | |
| Statistical Treatments | 17 | 1966-89 |
| Complete Model Checking | 0 | |
| Testing Representative Parts of the Model | 4 | 1959-90 |
| Model Sensitivity | 14 | 1974-89 |
| Philosophical | 0 | |
| Comparisons with Other Models or Ideas | 12 | 1970-92 |
| Redundant Model Creation | 1 | 1987 |
| Modeling Language | 0 | |
| Client Acceptance | 2 | 1983 |
| **SEMI-TECHNICAL VALIDATION** | | |
| Model Structure & Data | 0 | |
| Total Model Analysis | 5 | 1963-90 |
| Representations Analysis | 19 | 1955-89 |
| Replicating the Past | 8 | 1975-87 |
| Comparisons to the Real World | 13 | 1971-89 |
| Predicting the Future | 30 | 1975-89 |
| Input/Output Comparisons | 3 | 1985-90 |
| Performance Comparisons | 1 | 1979 |
| Model Maintenance | 1 | 1987 |
| **HUMAN JUDGMENT VALIDATION** | | |
| First Impressions | 28 | 1982-90 |
| Expert Opinions | 5 | 1971-90 |
| Comparisons to Expert Sources | 8 | 1959-90 |
| User Participation – Initial | 3 | 1980-91 |
| User Participation – Model Development | 1 | 1951 |
| User Participation – Testing | 3 | 1984-87 |
| Expert/Subject Effect & Theoretical | 6 | 1967-76 |
| Client Judgment | 13 | 1987 |
| Economic Effects | 1 | 1976 |

## Appendix B.2: Validation Methodologies

### App.B.2a: Problem Domains – Technical Validation

| Validation Group | Problem Domains |
|---|---|
| Traditional Verification | Medicine |
| Mathematical Treatments | – |
| Statistical Treatments | Behavioral Science<br>Computer Simulation<br>Digital Simulation<br>Job Performance<br>MIS Research<br>Missile Systems<br>Other Simulations (3)<br>Psychology<br>Social Science |
| Complete Model Checking | – |
| Testing Representative Parts of the Model | Job Analysis<br>Manufacturing<br>Personnel<br>Personnel Selection |
| Model Sensitivity | Decision Making<br>Global Modeling<br>Leadership & Management<br>Management<br>Police Patrols<br>Regional Blood Supply Planning<br>Social Science<br>Water Ecosystems<br>Wilderness Recreation<br>World Iron & Steel Economy |
| Philosophical | – |
| Comparisons with Other Models or Ideas | Fault Diagnosis & Location (2)<br>Fire Fighting (3)<br>Linguistics<br>Police Patrols (2)<br>Regional Blood Supply Planning<br>Risk Analysis (Equity Securities)<br>Strategic Planning (Financial)<br>Telephone Traffic Routing |
| Redundant Model Creation | Fire Detection |
| Modeling Language | – |
| Client Acceptance | Expert Systems<br>Software Engineering |

**App.B.2b: Problem Domains – Semi-Technical Validation**

| Validation Group | Problem Domains |
|---|---|
| Model Structure & Data | – |
| Total Model Analysis | Defense Aircraft Communications SW<br>Manufacturing<br>Missile Systems Simulation<br>Publishing |
| Representations Analysis | Banking (Loans)<br>Decision Making (Electronics)<br>Decision Making (Management)<br>Educational Testing<br>Engineering<br>Job Preference<br>Military<br>MIS Research<br>Organizational Research<br>Psychology (4)<br>Utility Models<br>Utility Theory (2)<br>Water Quality |
| Replicating the Past | Global Modeling<br>Inventory<br>Leadership & Management<br>Semiconductor Manufacturing (2)<br>Water Ecosystems<br>World Iron & Steel Economy<br>Wilderness Recreation |
| Comparisons to the Real World | Coal & Electric Utilities<br>Discrete Event Simulation<br>Leadership & Management<br>Police Patrols<br>Regional Blood Suppliers<br>Simulation<br>Strategic Planning (Financial)<br>Systems Dynamics (2) |
| Predicting the Future | Energy Models<br>Industrial/Commercial DSS (Opns. Mgmt.) (22)<br>Inventory<br>Medical (Chest Pain Analysis)<br>Police Patrols<br>Regional Blood Suppliers<br>Strategic Planning (Financial)<br>Water Ecosystems<br>Wilderness Recreation |

**App.B.2b: Problem Domains – Semi-Technical Validation** *(cont'd)*

| Validation Group | Problem Domains |
|---|---|
| Input/Output Comparisons | Manufacturing |
| Performance Comparisons | Engineering |
| Model Maintenance | Nuclear Power Plants |

**App.B.2c: Problem Domains – Human Judgment Validation**

| Validation Group | Problem Domains |
|---|---|
| First Impressions | Industrial/Commercial DSS (Opns. Mgmt.) (25)<br>Leadership & Management<br>Life Insurance<br>Wilderness Recreation |
| Expert Opinions | Behavioral Research<br>Educational Testing<br>Electrocardiograms<br>Medicine<br>Wilderness Recreation |
| Comparisons to Expert Sources | Business (Scheduling)<br>Life Insurance<br>Manufacturing Production Scheduling<br>Medical<br>Medical (Chemotherapy)<br>Medical (MYCIN)<br>Psychology<br>Strategic Decision Making (Finance) |
| User Participation – Initial | Marketing (Advertising Strategy)<br>Manufacturing<br>Wilderness Recreation |
| User Participation – Model Development | |
| User Participation – Testing | Computer Systems Configuration<br>Refrigeration System Maintenance<br>Strategic Planning (Financial) |
| Expert/Subject Effect & Theoretical | Engineering<br>Behavioral Science<br>Ecology (2)<br>Games & Simulations |
| Theoretical/Client Judgment | Industrial/Commercial DSS (Opns.Mgmt.) (13) |
| Economic Effects | Regional Blood Supply Planning |

# Appendix C: The Data Collection Questionnaire

**PROBLEM SOLVING SYSTEMS -**
**A QUESTIONNAIRE**

This questionnaire is part of a research project to study several aspects of computer systems that were designed to help people make decisions about complex problems. The questionnaire is in five parts:

1. Short Description of your system.
2. Technical Complexity of your system.
3. The Involvement of People in your system.
4. The Observability of your system.
5. Verification and Validation of your system.

## PART 1.
## A Short Description of Your System

Briefly describe your Problem Solving System. What does it do? What problems or questions is it designed to address?

What is the name of this system? ______________________

In what year was the design/development of this system started? ____________

In what year was this system first used by the intended users? ____________

Is this system in use at the present time? ____________

In how many countries is this system used? ____________
Please list the primary countries in which it is used.
______________________
______________________

## Appendix C: *page 2*

**PART 2.**
**The Technical Complexity of Your System**

In this section you are asked to rate your system using the following eleven measures of complexity. Make one entry along the line for each feature. The numbers under each line are there to give you an indication of how each feature typically is measured. PLEASE ENTER NUMBERS (e.g., 50 Mbytes, 25 pages) wherever you have enough knowledge about the system to do so; approximate numbers are OK. Otherwise, just put a check mark somewhere along each line indicating your impressions of your system in terms of that feature. Be sure to make one entry for each of the eleven features.

| | | | | | | |
|---|---|---|---|---|---|---|
| Program Code | 10 | $10^2$ | $10^3$ | $10^4$ | $10^5$ | $10^6$ Lines |
| Hard Disk Space Required for System | 100K | 1Mb | 10Mb | 100Mb | 1,000Mb | 10,000 Mbytes |
| Number of Modules | 0 | 20 | 40 | 60 | 80 | 100 Modules |
| Number of Items User is Required to Input *(per run)* | 0 | 20 | 40 | 60 | 80 | 100 Items |
| Length of User Documentation | 0 | 20 | 40 | 60 | 80 | 100 Pages |
| Training Period | 0 | 20 | 40 | 60 | 80 | 100 Days |
| Amount of Output - *Pages* | 0 | 20 | 40 | 60 | 80 | 100 Pages |
| - *Screens* | 0 | 2 | 4 | 6 | 8 | 10 Screens |
| Development Time *(person-years)* | 0 | 2 | 4 | 6 | 8 | 10 Years |
| Development Cost *(Total)* | \$10 | \$100 | \$1000 | \$10,000 | \$100,000 | \$1,000,000 |
| Maintenance Cost *(per year)* | \$10 | \$100 | \$1000 | \$10,000 | \$100,000 | \$1,000,000 |
| Number of Installations | 0 | 20 | 40 | 60 | 80 | 100 Locations |

**PART 3.**
**The Involvement of People in Your System**

In this section you are asked to indicate how many people have been involved in seven specific phases of the life of your system. Please account for all people who have worked in *each phase* of the system. If one person has worked in more than one phase or role, include that person in every appropriate box.

Enter a number in each box on the table below. Approximate numbers are OK. If you know that there have been *no* people involved in a particular phase, enter *"0"*. If you absolutely do not know, enter *"N/A"*. Please do not leave any boxes blank.

Enter the **total number** of individuals involved in **each PHASE and ROLE**:

| | **ROLES** | | |
|---|---|---|---|
| **PHASES** | EXPERTS | DEVELOPERS | USERS |
| Problem Definition | | | |
| Design | | | |
| Development/Programming | | | |
| Testing | | | |
| Implementation | | | |
| Routine Use | | | |
| Interpretation of Output | | | |

Enter the number of **distinct individuals** involved in the above process.

**TOTAL number of individuals** who have worked on this system: ____________

*This number should include ALL persons noted above, including users of the system. It will likely not be the total of the entries in the table, since one person may have been counted in more than one role or phase.*

**YOUR ROLE(s)** in this system:

EXPERT ___ DEVELOPER ___ USER ___ OTHER ___
*(Check all that apply)*

If "Other" is checked, please indicate role *(e.g., Sponsor, Client, Manager)* ________

## PART 4.
## The Observability of Your System

A Problem Solving System (PSS) is designed to mimic certain features of an event or problem that occurs in the "real world." Usually it is of interest to compare the output of the PSS with what happens when the real event occurs. This requires that both the real event or problem and the PSS can be repeatedly observed.

**Frequency:** How frequently does the real problem or event occur? State this in terms of a number of occurrences during a certain time period (e.g., twice a day, once a month). Or, if it is more appropriate for your specific situation, state the frequency of occurrence in general terms (e.g., many times a day, daily, monthly). But please be as precise as possible.

| | | | | | |
|---|---|---|---|---|---|
| ______ | ______ | ______ | ______ | ______ | ______ |
| per Minute | per Hour | per Day | per Week | per Month | per Year |

**Length of occurrence:** When the event of interest occurs, what length of time does its "solution" require? State time in hours, days, weeks, months from the beginning or recognition of the problem to when it concludes, or is resolved.

| | | | | | |
|---|---|---|---|---|---|
| ______ | ______ | ______ | ______ | ______ | ______ |
| Minutes | Hours | Days | Weeks | Months | Years |

**Repetition:** How long does it take to run your system, from initiation (usually by a human action such as data input) to output? Please give answer in minutes, hours, weeks, etc., or in a time range (e.g., from 2-4 hours, several weeks).

| | | | | | |
|---|---|---|---|---|---|
| ______ | ______ | ______ | ______ | ______ | ______ |
| Minutes | Hours | Days | Weeks | Months | Years |

**Control:** Can the situation in the real world *be made to happen?* (That is, is it a situation such as medical tests that a doctor can order anytime or is it a natural phenomenon, such as an earthquake or hazardous waste spill, that cannot reasonably be "made" to occur?)

YES ______ NO ______ Comment: ________________________________

**Limitations:** What are the limitation on how often your PSS can be run to produce its output regarding this problem situation?

| | | | | | |
|---|---|---|---|---|---|
| ______ | ______ | ______ | ______ | ______ | ______ |
| Cost | Data availability | Time required to collect data | Availability of trained staff | Time required to run system | Other |

## PART 5.
## Verification and Validation of Your System

**Verification** of computer software is usually described as the process of determining the *internal* correctness of the program's code. That is, the goal of performing verification procedures is to be able to say that the programmed mathematical formulas, data structures and other logical constructions that make up the written software always perform perfectly.

Using this definition of verification, to what extent are you confident or convinced that your system has been verified? Please give a number as a percentage from *"0%"* to *"100%"*.

```
I______I______I______I______I______I______I______I______I______I______I
0      10     20     30     40     50     60     70     80     90     100%
```

**Validation** of computerized systems generally refers to the process of determining if the system is reliable *externally*. That is, a valid system correctly and accurately represents specified, relevant aspects of a problem or event that occurs in the "real" world outside the computer. Because it correctly models the real problem, a valid system is capable of always giving useful information or advice that assists a user in making appropriate decisions.

Using this definition of validation, to what extent are you confident or convinced that your system is valid? Please give a number as a percentage from *"0%"* to *"100%"*.

```
I______I______I______I______I______I______I______I______I______I______I
0      10     20     30     40     50     60     70     80     90     100%
```

Have there been any instances in which this system gave *false*, or erroneous results? What happened? Please describe.

There are many ways to validate systems. On the next page is a list of general categories of validation methods. A system may be validated using multiple methods, either together in one phase of its development or during different phases.

**PART 5.** - cont'd
**Describe why you trust your system.**

Please put check marks in the boxes below to support the following statement:

**Our trust in our PSS is based on the results of the following validation methods:**

Check as many boxes as are appropriate, indicating the phase(s) of development during which the Validation Method occured.

| VALIDATION METHODS | PHASES | | |
|---|---|---|---|
| | Initial Design Phase | Programming & Debugging Phase | Initial User Phase |
| **Technical Methods:** Program compiles & runs without crashing | | | |
| Mathematical tests of programming logic | | | |
| Statistical tests of programming logic | | | |
| Traced & tested all possible inputs, their interactions & results | | | |
| Traced & tested representative inputs, their relationships & results | | | |
| Determination that repeated identical input always yields same result | | | |
| Repeated testing/use of PSS gave no bad results | | | |
| Compared PSS with other existing methods of solving the problem | | | |
| Created more than one PSS & tested against each other | | | |
| Determination that best programming language was used | | | |
| PSS passed client tests/procedures for acceptance (e.g., "benchmarks") | | | |
| **Semi-Technical Methods:** Qualitative assessment of results of mathematical/logic testing | | | |
| Using scenarios or other verbal/manual tests of the whole system | | | |
| Determine if variables are complete, unbiased, accurate representations | | | |
| Ran PSS with past data; compared model's output with actual occurrences | | | |
| Ran PSS in real-time situations & compared to on-going actual situation | | | |
| Ran PSS to predict future events; wait to compare with actual occurrence | | | |
| Controlled tests in which repeated inputs are compared to generated outputs | | | |
| PSS was improved based on comparison of its results to real events | | | |
| Determination that model can be continually improved, refined & expanded | | | |
| **Human Judgement Methods:** Favorable first impressions of PSS by potential users | | | |
| Experts' opinions of PSS are positive | | | |
| PSS compared well to expert sources (e.g., popular published methods) | | | |
| Users participated in PSS design, development &/or testing | | | |
| Modelling process clarified details, yielding better understanding of real problem | | | |
| Client liked/accepted PSS; client paid for PSS | | | |
| Users are satisfied & pay for PSS results; i.e., PSS generates positive cash flow | | | |

Thank you for your help on this questionnaire. All information contained herein will be kept confidential.

Date Questionnaire Completed: ____________________

# Appendix D: Variable Names & Descriptions

The following table lists the variables that contain all data from the questionnaires. This data was originally captured in a FoxPro database file, then downloaded to ASCII for importing and use in SAS.

| # | Variable Name | Variable Description | Values |
|---|---|---|---|
| 1 | NUMBER | Identification number of each observation | |
| 2 | YR_BEG | Year development of system began | |
| 3 | YR_USD | Year system was placed in use | |
| 4 | IN_USE | System in use now | Y=1, N=0 |
| 5 | CNTRYS | Number of countries system is curently in use | |
| 6 | T1_CAT | Category for number of lines of program code | 0-6 |
| 7 | T2_CAT | Category for hard disk space required | 0-6 |
| 8 | T3_CAT | Category for number of modules | 0-6 |
| 9 | T4_CAT | Category for number of items user must input | 0-6 |
| 10 | T5_CAT | Category for length of user documentation | 0-6 |
| 11 | T6_CAT | Category for number of training days | 0-6 |
| 12 | T7A_CAT | Category for number of output pages | 0-6 |
| 13 | T7B_CAT | Category for number of output screens | 0-6 |
| 14 | T8_CAT | Category for length of development time | 0-6 |
| 15 | T9_CAT | Category for cost of development | 0-6 |
| 16 | T10_CAT | Category for cost of maintenance | 0-6 |
| 17 | T11_CAT | Category for number of installations | 0-6 |
| 18 | H1E_DEFN | Number of experts in Problem Definition Phase | $k$[1] |
| 19 | H1D_DEFN | Number of developers in Problem Definition Phase | $k$ |
| 20 | H1U_DEFN | Number of users in Problem Definition Phase | $k$ |
| 21 | H2E_DES | Number of experts in Design Phase | $k$ |
| 22 | H2D_DES | Number of developers in Design Phase | $k$ |
| 23 | H2U_DES | Number of users in Design Phase | $k$ |
| 24 | H3E_PRG | Number of experts in Programming Phase | $k$ |
| 25 | H3D_PRG | Number of developers in Programming Phase | $k$ |
| 26 | H3U_PRG | Number of users in Programming Phase | $k$ |
| 27 | H4E_TEST | Number of experts in Testing Phase | $k$ |
| 28 | H4D_TEST | Number of developers in Testing Phase | $k$ |
| 29 | H4U_TEST | Number of users in Testing Phase | $k$ |
| 30 | H5E_IMPL | Number of experts in Implementation Phase | $k$ |

[1] '$k$' values of these variables are the exact numbers given by the respondents to the questionnaire.

| # | Variable Name | Variable Description | Values |
|---|---|---|---|
| 31 | H5D_IMPL | Number of developers in Implementation Phase | *k* |
| 32 | H5U_IMPL | Number of users in Implementation Phase | *k* |
| 33 | H6E_USR | Number of experts involved in Routine Use | *k* |
| 34 | H6D_USR | Number of developers involved in Routine Use | *k* |
| 35 | H6U_USR | Number of users involved in Routine Use | *k* |
| 36 | H7E_INTR | Number of experts involved in Interpretation of Output | *k* |
| 37 | H7D_INTR | Number of developers involved in Interpretation of Output | *k* |
| 38 | H7U_INTR | Number of users involved in Interpretation of Output | *k* |
| 39 | H8_TOTAL | Total number of people involved with the system | *k* |
| 40 | H9_DEV | Respondent is a System Developer | Y=1, N=0 |
| 41 | H9_USER | Respondent is a System User | Y=1, N=0 |
| 42 | H9_EXPRT | Respondent is a System Expert | Y=1, N=0 |
| 43 | O1_CAT | Category for frequency of occurrence of real event | 0-7 |
| 44 | O2_CAT | Category for length of occurrence of real event | 0-7 |
| 45 | O3_CAT | Category for human control over real event | 0-7 |
| 46 | O4_CAT | Category for length of tims system takes to run | 0-7 |
| 47 | O5_COST | Cost as a limitation on running system | Y=1, N=0 |
| 48 | O5_AVAIL | Data availability as a limitation on running system | Y=1, N=0 |
| 49 | O5_COLCT | Time to collect data as a limitation on running system | Y=1, N=0 |
| 50 | O5_STAFF | Staff availability as a limitation on running system | Y=1, N=0 |
| 51 | O5_RUNS | Time to run system as a limitation on running system | Y=1, N=0 |
| 52 | O5_OTHER | Other limitations on running system | Y=1, N=0 |
| 53 | O6_TOTAL | Calculated score for system Observability | -6 to 22 |
| 54 | V1VERPCT | Reported level of system Verification | 0-100 |
| 55 | V2VALPCT | Reported level of system Validation | 0-100 |
| 56 | VT1_DES[2] | Program compiles & runs without crashing | Y=1, N=0 |
| 57 | VT1_PRG[3] | " " | Y=1, N=0 |
| 58 | VT1_USR[4] | " " | Y=1, N=0 |

[2] Suffix '*_DES*' indicates during Design Phase

[3] Suffix '*_PRG*' indicates during Programming Phase

[4] Suffix '*_USR*' indicates during User Phase

| # | Variable Name | Variable Description | | Values |
|---|---|---|---|---|
| 59 | VT2_DES | Mathematical tests of programming logic | | Y=1, N=0 |
| 60 | VT2_PRG | " | " | Y=1, N=0 |
| 61 | VT2_USR | " | " | Y=1, N=0 |
| 62 | VT3_DES | Statistical tests of programming logic | | Y=1, N=0 |
| 63 | VT3_PRG | " | " | Y=1, N=0 |
| 64 | VT3_USR | " | " | Y=1, N=0 |
| 65 | VT4_DES | Traced & tested all possible inputs, their interactions & results | | Y=1, N=0 |
| 66 | VT4_PRG | " | " | Y=1, N=0 |
| 67 | VT4_USR | " | " | Y=1, N=0 |
| 68 | VT5_DES | Traced & tested representative inputs, their interactions & results | | Y=1, N=0 |
| 69 | VT5_PRG | " | " | Y=1, N=0 |
| 70 | VT5_USR | " | " | Y=1, N=0 |
| 71 | VT6_DES | Determination that repeated identical input always yields same result | | Y=1, N=0 |
| 72 | VT6_PRG | " | " | Y=1, N=0 |
| 73 | VT6_USR | " | " | Y=1, N=0 |
| 74 | VT7_DES | Repeated testing/use of PSS gave no bad results | | Y=1, N=0 |
| 75 | VT7_PRG | " | " | Y=1, N=0 |
| 76 | VT7_USR | " | " | Y=1, N=0 |
| 77 | VT8_DES | Compared PSS with other existing methods of solving the problem | | Y=1, N=0 |
| 78 | VT8_PRG | " | " | Y=1, N=0 |
| 79 | VT8_USR | " | " | Y=1, N=0 |
| 80 | VT9_DES | Created more than one PSS & tested against each other | | Y=1, N=0 |
| 81 | VT9_PRG | " | " | Y=1, N=0 |
| 82 | VT9_USR | " | " | Y=1, N=0 |
| 83 | VT10_DES | Determination that best programming language was used | | Y=1, N=0 |
| 84 | VT10_PRG | " | " | Y=1, N=0 |
| 85 | VT10_USR | " | " | Y=1, N=0 |
| 86 | VT11_DES | PSS passed client tests/procedures for acceptance (e.g., "benchmarks") | | Y=1, N=0 |
| 87 | VT11_PRG | " | " | Y=1, N=0 |
| 88 | VT11_USR | " | " | Y=1, N=0 |

| # | Variable Name | Variable Description | Values |
|---|---|---|---|
| 89 | VS1_DES | Qualitative assessment of results of mathematical/logic testing | Y=1, N=0 |
| 90 | VS1_PRG | " " | Y=1, N=0 |
| 91 | VS1_USR | " " | Y=1, N=0 |
| 92 | VS2_DES | Using scenarios or other verbal/manual tests of the whole system | Y=1, N=0 |
| 93 | VS2_PRG | " " | Y=1, N=0 |
| 94 | VS2_USR | " " | Y=1, N=0 |
| 95 | VS3_DES | Determine if variables are complete, unbiased, accurate representations | Y=1, N=0 |
| 96 | VS3_PRG | " " | Y=1, N=0 |
| 97 | VS3_USR | " " | Y=1, N=0 |
| 98 | VS4_DES | Ran PSS with past data; compared model's output with actual occurrences | Y=1, N=0 |
| 99 | VS4_PRG | " " | Y=1, N=0 |
| 100 | VS4_USR | " " | Y=1, N=0 |
| 101 | VS5_DES | Ran PSS in real-time situations & compared to on-going actual situation | Y=1, N=0 |
| 102 | VS5_PRG | " " | Y=1, N=0 |
| 103 | VS5_USR | " " | Y=1, N=0 |
| 104 | VS6_DES | Ran PSS to predict future events; wait to compare with actual occurrence | Y=1, N=0 |
| 105 | VS6_PRG | " " | Y=1, N=0 |
| 106 | VS6_USR | " " | Y=1, N=0 |
| 107 | VS7_DES | Controlled tests in which repeated inputs are compared to generated outputs | Y=1, N=0 |
| 108 | VS7_PRG | " " | Y=1, N=0 |
| 109 | VS7_USR | " " | Y=1, N=0 |
| 110 | VS8_DES | PSS was improved based on comparison of its results to real events | Y=1, N=0 |
| 111 | VS8_PRG | " " | Y=1, N=0 |
| 112 | VS8_USR | " " | Y=1, N=0 |
| 113 | VS9_DES | Determination that model can be continually improved, refined & expanded | Y=1, N=0 |
| 114 | VS9_PRG | " " | Y=1, N=0 |
| 115 | VS9_USR | " " | Y=1, N=0 |
| 116 | VH1_DES | Favorable first impressions of PSS by potential users | Y=1, N=0 |
| 117 | VH1_PRG | " " | Y=1, N=0 |
| 118 | VH1_USR | " " | Y=1, N=0 |

| # | Variable Name | Variable Description | Values |
|---|---|---|---|
| 119 | VH2_DES | Experts' opinions of PSS are positive | Y=1, N=0 |
| 120 | VH2_PRG | " " | Y=1, N=0 |
| 121 | VH2_USR | " " | Y=1, N=0 |
| 122 | VH3_DES | PSS compared well to expert sources (e.g., popular published methods) | Y=1, N=0 |
| 123 | VH3_PRG | " " | Y=1, N=0 |
| 124 | VH3_USR | " " | Y=1, N=0 |
| 125 | VH4_DES | Users participated in PSS design, development &/or testing | Y=1, N=0 |
| 126 | VH4_PRG | " " | Y=1, N=0 |
| 127 | VH4_USR | " " | Y=1, N=0 |
| 128 | VH5_DES | Modelling process clarified details, yielding better understanding of real problem | Y=1, N=0 |
| 129 | VH5_PRG | " " | Y=1, N=0 |
| 130 | VH5_USR | " " | Y=1, N=0 |
| 131 | VH6_DES | Client liked/accepted PSS; client paid for PSS | Y=1, N=0 |
| 132 | VH6_PRG | " " | Y=1, N=0 |
| 133 | VH6_USR | " " | Y=1, N=0 |
| 134 | VH7_DES | Users are satisfied & pay for PSS results; i.e., PSS generates positive cash flow | Y=1, N=0 |
| 135 | VH7_PRG | " " | Y=1, N=0 |
| 136 | VH7_USR | " " | Y=1, N=0 |
| 137 | TEC_NUM | Calculated Technical Complexity score | 0-72 |
| 138 | VAL_NUM | Calculated total Test Methods score | 0-81 |
| 139 | VT1_GRP | Program compiles & runs without crashing | 0-3 |
| 140 | VT2_GRP | Mathematical tests of programming logic | 0-3 |
| 141 | VT3_GRP | Statistical tests of programming logic | 0-3 |
| 142 | VT4_GRP | Traced & tested all possible inputs, their interactions & results | 0-3 |
| 143 | VT5_GRP | Traced & tested representative inputs, their interactions & results | 0-3 |
| 144 | VT6_GRP | Determination that repeated identical input always yields same result | 0-3 |
| 145 | VT7_GRP | Repeated testing/use of PSS gave no bad results | 0-3 |
| 146 | VT8_GRP | Compared PSS with other existing methods of solving the problem | 0-3 |

Appendix D: *page 6*

| # | Variable Name | Variable Description | Values |
|---|---|---|---|
| 147 | VT9_GRP | Created more than one PSS & tested against each other | 0-3 |
| 148 | VT10_GRP | Determination that best programming language was used | 0-3 |
| 149 | VT11_GRP | PSS passed client tests/procedures for acceptance (e.g., "benchmarks") | 0-3 |
| 150 | VS1_GRP | Qualitative assessment of results of mathematical/logic testing | 0-3 |
| 151 | VS2_GRP | Using scenarios or other verbal/manual tests of the whole system | 0-3 |
| 152 | VS3_GRP | Determine if variables are complete, unbiased, accurate representations | 0-3 |
| 153 | VS4_GRP | Ran PSS with past data; compared model's output with actual occurrences | 0-3 |
| 154 | VS5_GRP | Ran PSS in real-time situations & compared to on-going actual situation | 0-3 |
| 155 | VS6_GRP | Ran PSS to predict future events; wait to compare with actual occurrence | 0-3 |
| 156 | VS7_GRP | Controlled tests in which repeated inputs are compared to generated outputs | 0-3 |
| 157 | VS8_GRP | PSS was improved based on comparison of its results to real events | 0-3 |
| 158 | VS9_GRP | Determination that model can be continually improved, refined & expanded | 0-3 |
| 159 | VH1_GRP | Favorable first impressions of PSS by potential users | 0-3 |
| 160 | VH2_GRP | Experts' opinions of PSS are positive | 0-3 |
| 161 | VH3_GRP | PSS compared well to expert sources (e.g., popular published methods) | 0-3 |
| 162 | VH4_GRP | Users participated in PSS design, development &/or testing | 0-3 |
| 163 | VH5_GRP | Modelling process clarified details, yielding better understanding of real problem | 0-3 |
| 164 | VH6_GRP | Client liked/accepted PSS; client paid for PSS | 0-3 |
| 165 | VH7_GRP | Users are satisfied & pay for PSS results; i.e., PSS generates positive cash flow | 0-3 |

## Appendix E: Listing of the Data

| OBS. # | Year Dev. Began | Year 1st in Use | In Use Now | In # Coun-tries | T1-CAT | T2-CAT | T3-CAT | T4-CAT | T5-CAT | T6-CAT | T7A-CAT | T7B-CAT | T8-CAT |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 1 | 1989 | 1990 | T | 1 | 2 | 1 | 1 | 1 | 1 | 1 | 1 | 2 | 1 |
| 2 | 1989 | 1990 | F | 1 | 5 | 3 | 1 | 6 | 5 | 2 | 0 | 6 | 0 |
| 3 | 1982 | 1984 | T | 4 | 4 | 3 | 1 | 6 | 5 | 1 | 6 | 0 | 0 |
| 4 | 1980 | 1989 | T | 3 | 3 | 2 | 2 | 1 | 2 | 1 | 1 | . | 3 |
| 5 | 1989 | 1990 | F | 0 | 4 | 2 | 1 | 6 | 6 | 1 | 1 | . | 2 |
| 6 | 1979 | 1985 | T | 4 | 5 | 4 | 6 | 6 | 6 | 2 | 6 | . | . |
| 7 | 1987 | 1989 | T | 2 | 3 | 1 | 1 | 3 | 3 | 2 | 1 | . | 2 |
| 8 | 1994 | 1994 | T | 1 | 3 | 2 | 1 | 2 | 1 | 1 | 1 | . | 2 |
| 9 | 1991 | 1993 | T | 1 | 2 | 2 | 1 | 1 | 1 | 1 | 1 | 0 | 1 |
| 10 | 1990 | 1990 | T | 2 | 4 | 3 | 6 | 1 | 2 | 1 | 1 | 6 | 1 |
| 11 | 1994 | 1995 | T | 1 | 3 | 1 | 1 | 2 | 2 | 1 | 1 | 2 | 2 |
| 12 | 1992 | 1994 | T | 1 | 4 | 2 | 1 | 6 | 1 | 1 | 1 | 0 | 2 |
| 13 | 1993 | 1994 | T | 2 | 6 | 4 | 6 | 6 | 6 | 1 | 1 | . | 5 |
| 14 | 1985 | 1986 | T | 1 | 5 | 4 | 1 | 6 | 3 | . | . | . | 6 |
| 15 | 1990 | 1994 | T | 1 | 4 | 4 | 1 | 6 | 1 | 5 | 0 | . | 2 |
| 16 | 1993 | 1993 | F | 2 | 4 | 4 | 1 | 6 | 6 | 1 | . | . | 3 |
| 17 | 1984 | 1984 | T | 1 | 3 | 3 | 1 | 2 | 3 | 1 | 1 | 0 | 1 |
| 18 | 1990 | 1990 | F | 1 | 3 | 3 | 1 | 3 | 5 | 4 | 2 | . | 2 |
| 19 | 1993 | 1993 | T | 1 | 3 | 3 | 1 | 2 | 1 | 1 | 1 | 3 | 2 |
| 20 | 1992 | 1995 | T | 1 | 3 | 4 | 1 | 2 | 2 | 1 | 1 | 1 | 1 |
| 21 | 1993 | 1993 | T | 1 | 3 | 1 | 1 | 2 | 1 | 1 | 2 | 3 | 1 |
| 22 | 1993 | 1993 | T | 1 | 4 | 1 | 2 | 2 | 0 | 1 | 3 | 5 | 1 |
| 23 | 1995 | 1995 | T | 1 | 3 | 1 | 1 | 2 | 1 | 1 | 1 | 3 | 1 |
| 24 | 1970 | 1977 | T | 2 | 5 | 2 | 6 | 1 | 6 | 1 | 1 | 1 | 6 |
| 25 | 1990 | 1990 | T | 1 | 2 | 0 | 1 | 6 | 1 | 1 | 1 | 1 | 1 |
| 26 | 1991 | 1993 | T | 1 | 5 | 5 | 6 | 6 | 6 | 3 | 6 | 6 | 6 |
| 27 | 1987 | 1987 | T | 1 | 3 | 1 | 1 | 1 | 0 | 1 | 3 | 3 | 1 |
| 28 | 1994 | 1995 | T | 4 | 4 | 0 | 6 | 6 | 6 | 1 | 1 | . | 5 |
| 29 | 1986 | 1987 | T | 4 | 4 | 2 | 6 | 2 | 2 | 1 | 1 | 3 | 3 |
| 30 | 1989 | 1991 | T | 6 | 4 | 4 | 5 | 0 | 6 | 1 | 5 | 0 | 6 |
| 31 | 1981 | 1981 | T | 1 | 3 | 4 | 2 | 3 | 1 | 2 | 6 | 6 | 5 |
| 32 | 1983 | 1985 | T | 6 | 3 | 2 | 1 | 1 | 1 | 1 | 6 | 5 | 3 |
| 33 | 1987 | 1987 | T | 1 | 6 | 6 | 1 | 1 | 6 | 2 | 6 | 6 | 6 |
| 34 | 1984 | 1993 | T | 1 | 6 | 6 | 1 | 2 | 6 | 5 | 6 | 6 | 2 |
| 35 | 1993 | 1995 | T | 1 | 3 | 2 | . | 2 | 1 | . | . | 1 | 1 |
| 36 | 1985 | 1985 | T | 1 | 2 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| 37 | 1992 | 1993 | T | 3 | 5 | 5 | 1 | 3 | 6 | 1 | 6 | 6 | 6 |
| 38 | 1991 | 1991 | T | 1 | 3 | 2 | 1 | 6 | 3 | 1 | 1 | 0 | 1 |
| 39 | 1986 | 1988 | T | 12 | 6 | 3 | 5 | 6 | 6 | 4 | 1 | 6 | 6 |
| 40 | 1964 | 1974 | T | 1 | . | . | . | . | . | . | . | . | . |

## Appendix E. - *page 2*

| OBS. # | T9_ CAT | T10_ CAT | T11_ CAT | H1E_ DEFN | H1D_ DEFN | H1U_ DEFN | H2E_ DES | H2D_ DES | H2U_ DES | H3E_ PGMG | H3D_ PGMG | H3U_ PGMG |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 1 | 2 | 1 | 1 | 2 | 0 | 0 | 2 | 0 | 0 | 1 | 0 | 0 |
| 2 | 5 | 3 | 1 | 3 | 0 | 0 | 3 | 1 | 0 | 2 | 5 | 3 |
| 3 | 5 | 3 | 1 | 5 | 4 | 2 | 5 | 4 | 2 | 4 | 5 | 0 |
| 4 | 5 | 2 | 5 | 3 | 4 | 0 | 2 | 3 | 0 | 1 | 3 | 0 |
| 5 | 5 | 1 | 1 | 1 | 1 | 1 | 0 | 3 | 0 | 1 | 10 | 0 |
| 6 | 6 | 4 | 1 | . | . | . | . | . | . | . | . | . |
| 7 | 4 | 4 | 1 | 5 | 1 | 1 | 2 | 1 | 1 | 1 | 1 | 1 |
| 8 | 5 | 3 | 1 | 1 | 2 | 0 | 3 | 3 | 0 | 1 | 2 | 0 |
| 9 | 1 | 1 | 1 | 0 | 1 | 0 | 0 | 1 | 0 | 0 | 1 | 0 |
| 10 | 2 | 2 | 1 | . | . | . | . | . | . | . | . | . |
| 11 | 4 | 1 | 1 | 2 | 2 | 0 | 3 | 2 | 0 | 1 | 1 | 0 |
| 12 | 5 | . | 1 | 2 | 1 | 0 | 0 | 2 | 0 | 0 | 2 | 0 |
| 13 | 5 | 4 | 1 | 1 | 1 | 1 | 1 | 3 | 0 | 1 | 10 | 0 |
| 14 | 6 | . | 1 | 2 | 3 | 1 | 2 | 2 | 1 | 1 | 4 | 1 |
| 15 | 6 | . | 1 | 2 | 0 | 0 | 0 | 2 | 0 | 0 | 5 | 0 |
| 16 | 6 | . | . | 4 | 0 | 1 | 0 | 2 | 0 | 0 | 7 | 0 |
| 17 | 3 | 3 | 1 | 0 | 1 | 4 | 0 | 1 | 2 | 0 | 1 | 1 |
| 18 | . | . | 1 | 2 | 1 | 1 | 2 | 1 | 1 | 2 | 1 | 1 |
| 19 | 4 | 2 | 1 | 2 | 2 | 0 | 2 | 2 | 0 | 0 | 2 | 0 |
| 20 | 5 | 4 | 0 | 3 | 1 | 4 | 1 | 1 | 3 | 1 | 2 | 1 |
| 21 | 3 | 2 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 0 |
| 22 | 3 | 3 | 1 | 2 | 2 | 2 | 2 | 2 | 0 | 0 | 2 | 0 |
| 23 | 3 | 2 | 1 | 1 | 1 | 0 | 1 | 1 | 0 | 0 | 1 | 0 |
| 24 | 5 | 3 | 6 | 1 | 5 | 1 | 3 | 4 | 2 | 2 | 9 | 0 |
| 25 | 2 | 1 | 1 | 3 | 1 | 0 | 1 | 1 | 0 | 0 | 1 | 0 |
| 26 | 6 | 4 | 1 | 3 | 5 | 10 | 5 | 5 | 3 | 5 | 10 | 0 |
| 27 | 4 | 4 | 1 | 3 | 2 | 2 | 1 | 1 | 0 | 1 | 1 | 0 |
| 28 | 5 | 5 | 4 | 3 | 1 | 2 | 1 | 5 | 0 | 0 | 5 | 0 |
| 29 | 4 | 2 | 1 | 2 | 0 | 1 | 2 | 0 | 0 | 2 | 0 | 0 |
| 30 | 6 | 6 | 6 | 1 | 1 | 0 | 1 | 3 | 0 | 1 | 3 | 0 |
| 31 | 4 | 4 | 1 | 1 | 0 | 0 | 1 | 1 | 2 | 1 | 1 | 0 |
| 32 | 5 | 2 | 5 | 3 | 2 | 0 | 3 | 2 | 0 | 0 | 2 | 0 |
| 33 | 4 | 4 | 1 | 4 | 4 | 40 | 4 | 4 | 15 | 4 | 4 | 15 |
| 34 | 5 | 4 | 2 | 2 | 3 | 20 | 4 | 5 | 3 | 3 | 5 | 3 |
| 35 | 4 | 1 | 1 | 5 | 1 | 15 | 1 | 2 | 0 | 1 | 2 | 0 |
| 36 | 4 | 1 | 2 | 1 | 1 | 0 | 1 | 1 | 0 | 0 | 1 | 0 |
| 37 | 6 | 5 | 6 | 2 | . | 10 | . | . | . | . | . | . |
| 38 | 1 | 1 | 1 | 1 | 1 | 0 | 1 | 1 | 0 | 1 | 1 | 0 |
| 39 | 6 | 5 | 6 | 1 | 3 | 0 | 1 | 3 | 0 | 0 | 6 | 0 |
| 40 | . | . | . | . | . | . | . | . | . | . | . | . |

## Appendix E. - *page 3*

| OBS. # | H4E-TST | H4D-TST | H4U-TST | H5E-IMPL | H5D-IMPL | H5U-IMPL | H6E-USE | H6D-USE | H6U-USE | H7E-INTR | H7D-INTR | H7U-INTR | H8-TOTL |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 1 | 2 | 0 | 3 | 2 | 2 | 0 | 0 | 0 | 2 | 0 | 0 | 0 | 6 |
| 2 | 1 | 2 | 3 | 1 | 1 | 0 | 1 | 0 | 1 | 1 | 0 | 3 | 12 |
| 3 | 4 | 3 | 1 | 3 | 3 | 3 | 4 | 0 | 2 | 4 | 0 | 3 | 20 |
| 4 | 2 | 2 | 0 | 0 | 2 | 2 | 0 | 0 | 15 | 2 | 0 | 15 | 22 |
| 5 | 1 | 2 | 1 | 0 | 2 | 2 | 0 | 2 | 0 | 0 | 0 | 0 | 15 |
| 6 | . | . | . | . | . | . | . | . | . | . | . | . | 77 |
| 7 | 1 | 1 | 1 | 1 | 1 | 2 | 1 | 1 | 4 | 2 | 1 | 2 | 6 |
| 8 | 1 | 1 | 0 | 2 | 3 | 0 | 5 | 5 | 0 | 2 | 2 | 0 | 7 |
| 9 | 0 | 1 | 0 | 0 | 1 | 0 | 0 | 0 | 2 | 0 | 1 | 0 | 2 |
| 10 | . | . | . | . | . | . | . | . | . | . | . | . | 10 |
| 11 | 1 | 1 | 0 | 0 | 1 | 1 | 0 | 0 | 5 | 0 | 0 | 5 | 6 |
| 12 | 1 | 1 | 0 | 0 | 2 | 0 | 3 | 0 | 4 | 2 | 1 | 0 | 4 |
| 13 | 1 | 10 | 0 | 0 | 10 | 0 | 1 | 1 | 5 | 1 | 1 | 5 | 15 |
| 14 | 2 | 2 | 2 | 1 | 5 | 1 | 1 | 0 | 2 | 1 | 0 | 2 | 10 |
| 15 | 1 | 2 | 2 | 0 | 6 | 0 | 0 | 0 | 1 | 0 | 0 | . | 10 |
| 16 | 1 | 2 | 1 | 0 | 6 | 0 | . | . | . | . | . | . | 12 |
| 17 | 0 | 1 | 0 | 0 | 1 | 2 | 0 | 0 | 10 | 0 | 0 | 0 | 15 |
| 18 | 2 | 1 | 1 | 1 | 1 | 1 | 0 | 0 | 1 | 1 | 1 | 1 | 3 |
| 19 | 1 | 1 | 3 | 1 | 0 | 3 | 1 | 0 | 6 | 2 | 0 | 0 | 9 |
| 20 | 2 | 2 | 3 | 1 | . | 1 | 1 | 0 | 3 | 5 | 0 | 3 | 5 |
| 21 | 0 | 1 | 2 | 0 | 1 | 2 | 0 | 1 | 3 | 1 | 0 | 2 | 4 |
| 22 | 0 | 2 | 1 | 0 | 1 | 1 | 0 | 1 | 0 | 1 | 0 | 2 | 4 |
| 23 | 1 | 1 | 4 | 1 | 0 | 6 | 1 | 0 | 2 | 1 | 0 | 0 | 8 |
| 24 | 3 | 9 | 5 | 2 | 5 | 0 | . | . | . | . | . | . | 20 |
| 25 | 0 | 1 | 3 | 0 | 1 | 0 | 0 | 0 | 30 | 0 | 0 | 50 | 54 |
| 26 | 5 | 5 | 10 | 4 | 2 | 10 | 2 | 0 | 100 | 2 | 0 | 20 | 120 |
| 27 | 2 | 1 | 3 | 1 | 1 | 0 | 0 | 0 | 20 | 1 | 0 | 3 | 27 |
| 28 | 1 | 7 | 0 | 1 | 1 | 100 | 1 | 1 | 100 | 1 | 1 | 100 | 112 |
| 29 | 2 | 0 | 1 | 2 | 0 | 1 | 2 | 0 | 6 | 2 | 0 | 6 | 8 |
| 30 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 200 | 0 | 0 | 200 | 200 |
| 31 | 1 | 1 | 20 | 1 | 1 | 20 | 0 | 0 | 20 | 1 | 1 | 20 | 25 |
| 32 | 3 | 1 | 5 | 3 | 1 | 0 | 1 | 0 | 5 | 1 | 0 | 5 | 10 |
| 33 | 4 | 4 | 5 | 4 | 4 | 15 | 4 | 4 | 40 | 4 | 4 | 20 | 50 |
| 34 | 3 | 7 | 7 | 10 | 0 | 14 | 10 | 0 | 20 | 3 | 3 | 20 | 20 |
| 35 | 1 | 1 | 20 | 1 | 1 | 0 | 0 | 1 | 20 | 1 | 1 | 20 | 20 |
| 36 | 1 | 1 | 1 | 1 | 1 | 2 | 0 | 0 | 10 | 0 | 0 | 100 | 103 |
| 37 | 12 | . | 35 | 12 | . | 25 | 15 | . | 300 | 5 | . | 30 | 300 |
| 38 | 1 | 1 | 2 | 1 | 0 | 1 | 0 | 1 | 2 | 0 | 0 | 0 | 2 |
| 39 | 1 | 5 | 0 | 0 | 5 | 0 | 0 | 5 | 500 | 1 | 5 | 0 | 510 |
| 40 | . | . | . | . | . | . | . | . | . | . | . | . | . |

## Appendix E. - *page 4*

| OBS. # | H9_ DEV | H9_ USER | H9_ EXPT | O1_ CAT | O2_ CAT | O3_ CAT | O4_ CAT | O5_ COST | O5_ AVAIL | O5_ COLCT | O5_ STAFF | O5_ RUNS |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 1 | T | F | T | 3 | 1 | F | 6 | F | F | F | F | F |
| 2 | T | F | T | 1 | 0 | T | 4 | F | F | F | T | F |
| 3 | F | T | T | 0 | 0 | F | 3 | F | F | F | F | F |
| 4 | T | F | T | 1 | 3 | T | 5 | F | F | F | F | F |
| 5 | T | F | F | 1 | 2 | F | 3 | F | T | T | F | T |
| 6 | T | T | F | 1 | 2 | F | 4 | F | T | F | F | F |
| 7 | T | T | T | 1 | 1 | F | 3 | F | T | F | F | T |
| 8 | T | T | T | 1 | 6 | T | 5 | F | F | F | F | T |
| 9 | T | T | T | 4 | 6 | T | 6 | F | F | F | F | F |
| 10 | T | F | F | 2 | 4 | T | 5 | F | F | F | F | F |
| 11 | T | F | T | 1 | 3 | T | 3 | F | T | T | F | F |
| 12 | T | F | F | 6 | 5 | F | 6 | F | F | F | F | F |
| 13 | F | F | T | 1 | 2 | T | 3 | F | F | F | F | F |
| 14 | T | F | F | 1 | 4 | F | 4 | T | T | F | F | F |
| 15 | F | T | F | 1 | 0 | F | 5 | F | F | F | F | T |
| 16 | T | F | F | 1 | 3 | T | 3 | F | F | T | F | F |
| 17 | F | T | F | 4 | 6 | T | 6 | F | F | F | F | F |
| 18 | T | T | T | 3 | T | T | 3 | F | F | F | F | F |
| 19 | T | T | T | 3 | 4 | T | 6 | F | T | F | F | F |
| 20 | T | T | T | 2 | 2 | F | 3 | F | T | T | F | F |
| 21 | F | F | T | 4 | 5 | T | 5 | F | F | F | T | F |
| 22 | F | T | T | 3 | 5 | T | 5 | F | F | F | T | F |
| 23 | F | T | T | 2 | 2 | T | 4 | F | F | F | F | F |
| 24 | T | T | T | 0 | 0 | T | 6 | F | T | T | F | F |
| 25 | T | T | F | 3 | 6 | F | 6 | F | F | T | F | F |
| 26 | F | F | F | 5 | 4 | T | 5 | F | F | T | F | T |
| 27 | T | F | T | 4 | . | T | 6 | T | T | F | F | T |
| 28 | T | F | T | 7 | 7 | T | 0 | F | F | F | F | F |
| 29 | T | T | T | 0 | 0 | T | 6 | F | T | F | T | F |
| 30 | T | F | F | 1 | 1 | T | 1 | T | F | T | T | T |
| 31 | F | F | F | 5 | 6 | T | 6 | F | F | F | F | F |
| 32 | F | T | F | 2 | 2 | T | 4 | F | T | T | F | F |
| 33 | T | F | T | 4 | 6 | F | 6 | F | F | F | F | F |
| 34 | T | T | T | 6 | 6 | T | 6 | T | F | F | T | F |
| 35 | T | F | T | 5 | 6 | T | 6 | F | F | F | T | F |
| 36 | T | F | F | 2 | 6 | T | 2 | F | T | T | F | F |
| 37 | F | F | T | 7 | 5 | T | 6 | F | F | F | F | F |
| 38 | T | T | T | 2 | 4 | T | 5 | F | T | T | F | F |
| 39 | T | F | F | 1 | 5 | T | 5 | F | T | T | F | F |
| 40 | . | . | . | 1 | 1 | T | 5 | F | T | T | F | F |

| OBS. # | O5_ OTHR | O6_ TOTL | VER % | VAL % | VT1_ DES | VT1_ PRG | VT1_ USR | VT2_ DES | VT2_ PRG | VT2_ USR | VT3_ DES | VT3_ PRG | VT3_ USR |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 1 | F | 10 | 100 | 60 | F | T | T | F | F | F | F | F | F |
| 2 | F | 3 | 80 | 20 | . | . | . | . | . | . | . | . | . |
| 3 | T | 2 | 98 | 95 | . | . | . | . | . | . | . | . | . |
| 4 | F | 8 | 95 | 75 | F | T | T | T | F | F | T | F | F |
| 5 | F | 3 | 70 | 80 | F | T | F | F | F | F | F | F | F |
| 6 | T | 5 | 99 | 80 | T | T | F | T | T | F | F | F | F |
| 7 | F | 3 | 90 | 80 | T | T | T | T | T | F | F | F | F |
| 8 | F | 10 | 90 | 80 | F | F | T | T | F | F | T | F | F |
| 9 | F | 15 | 100 | 99 | . | . | . | . | . | . | . | . | . |
| 10 | F | 10 | 75 | 70 | . | . | . | . | . | . | . | . | . |
| 11 | F | 4 | 85 | 65 | F | T | T | F | T | F | T | T | F |
| 12 | F | 17 | 99 | 80 | . | . | . | . | . | . | . | . | . |
| 13 | F | 5 | 50 | 0 | F | T | F | F | F | F | F | F | F |
| 14 | F | 7 | 95 | 90 | F | T | F | F | F | F | F | F | F |
| 15 | F | 5 | 100 | 100 | F | T | F | F | F | F | F | F | F |
| 16 | F | 5 | 70 | 75 | F | T | F | F | F | F | F | F | F |
| 17 | F | 15 | 100 | 100 | F | T | T | F | T | T | F | T | T |
| 18 | T | 4 | 80 | 70 | . | . | . | . | . | . | . | . | . |
| 19 | F | 11 | 96 | 85 | F | F | T | F | F | F | T | T | T |
| 20 | F | 5 | 70 | 80 | F | T | F | T | F | F | T | F | F |
| 21 | F | 12 | 90 | 80 | F | T | T | F | F | F | T | T | T |
| 22 | F | 11 | 70 | 70 | F | T | T | F | F | F | F | F | T |
| 23 | F | 7 | 90 | 90 | T | T | T | F | F | F | F | T | T |
| 24 | F | 3 | 99 | 70 | T | T | T | F | T | F | F | F | F |
| 25 | F | 14 | 95 | 95 | T | T | F | T | T | F | T | T | F |
| 26 | F | 11 | 90 | 80 | F | T | F | F | F | F | F | F | F |
| 27 | T | 5 | 90 | 80 | F | T | T | T | T | T | F | F | F |
| 28 | F | 13 | 90 | 95 | T | T | F | F | F | F | F | F | F |
| 29 | F | 3 | 98 | 20 | T | T | T | T | T | T | F | F | F |
| 30 | F | -2 | 75 | 95 | F | T | T | F | F | F | F | F | F |
| 31 | T | 15 | 92 | 63 | T | T | T | F | F | F | F | F | F |
| 32 | F | 5 | 95 | 98 | T | F | F | T | F | F | T | F | F |
| 33 | F | 16 | 98 | 98 | T | T | T | T | T | T | F | F | F |
| 34 | F | 15 | 100 | 100 | F | F | T | F | F | T | F | F | T |
| 35 | F | 15 | 60 | 5 | T | T | T | F | F | F | F | F | F |
| 36 | F | 7 | 95 | 95 | T | T | T | T | T | F | F | F | F |
| 37 | F | 17 | 95 | 95 | T | T | F | F | T | F | F | T | F |
| 38 | F | 8 | 99 | 99 | T | T | F | F | T | T | F | F | F |
| 39 | F | 8 | 0 | 50 | T | T | T | T | T | T | F | F | F |
| 40 | F | 4 | 90 | 60 | F | T | T | T | F | T | F | F | F |

## Appendix E. - *page 6*

| OBS. # | VT4-DES | VT4-PRG | VT4-USR | VT5-DES | VT5-PRG | VT5-USR | VT6-DES | VT6-PRG | VT6-USR | VT7-DES | VT7-PRG | VT7-USR | VT8-DES |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 1 | F | T | T | F | T | T | F | T | T | F | T | T | F |
| 2 | . | . | . | . | . | . | . | . | . | . | . | . | . |
| 3 | . | . | . | . | . | . | . | . | . | . | . | . | . |
| 4 | F | F | T | F | F | F | F | F | T | F | F | T | F |
| 5 | F | F | F | F | T | T | F | T | T | F | F | T | T |
| 6 | F | F | F | F | T | F | F | T | F | F | T | T | F |
| 7 | F | F | F | T | T | T | T | T | T | F | T | T | T |
| 8 | F | F | F | F | T | T | F | T | F | F | F | F | F |
| 9 | . | . | . | . | . | . | . | . | . | . | . | . | . |
| 10 | . | . | . | . | . | . | . | . | . | . | . | . | . |
| 11 | F | F | F | F | F | T | F | F | T | F | F | F | F |
| 12 | . | . | . | . | . | . | . | . | . | . | . | . | . |
| 13 | F | F | F | T | T | T | F | T | F | T | T | T | T |
| 14 | F | F | F | F | T | T | F | T | T | F | F | T | F |
| 15 | F | F | F | F | T | F | F | T | T | F | F | T | F |
| 16 | F | F | F | F | T | T | F | T | F | F | F | T | F |
| 17 | F | T | T | T | T | T | F | T | T | F | F | T | F |
| 18 | . | . | . | . | . | . | . | . | . | . | . | . | . |
| 19 | F | T | F | F | T | F | F | T | T | T | T | T | T |
| 20 | F | T | T | F | T | F | F | F | T | F | F | T | T |
| 21 | F | F | F | F | T | T | F | T | F | F | T | T | T |
| 22 | F | F | F | F | F | F | F | F | F | F | F | F | F |
| 23 | F | T | T | F | F | F | F | F | T | F | T | T | T |
| 24 | F | T | F | F | T | F | F | T | T | F | F | F | F |
| 25 | F | T | F | F | T | F | F | T | F | F | T | T | F |
| 26 | F | F | F | F | T | F | F | T | T | F | F | T | F |
| 27 | T | T | F | T | T | T | F | T | F | F | T | T | F |
| 28 | F | F | F | T | T | F | F | F | F | T | T | T | F |
| 29 | F | F | F | T | T | T | T | T | T | T | T | T | T |
| 30 | F | T | T | F | T | T | F | T | T | F | T | T | F |
| 31 | T | T | T | T | T | F | T | T | F | F | F | F | F |
| 32 | T | T | F | T | T | F | T | T | T | F | F | F | F |
| 33 | T | T | T | F | F | F | T | T | T | T | T | T | T |
| 34 | F | F | T | F | F | T | F | F | T | F | F | T | F |
| 35 | F | F | F | T | T | T | T | T | T | F | F | F | F |
| 36 | F | F | F | F | T | F | F | T | F | F | T | T | F |
| 37 | F | T | F | F | T | F | F | T | F | F | F | T | F |
| 38 | F | T | F | F | T | T | F | T | T | F | F | F | F |
| 39 | F | F | F | T | T | T | F | T | T | F | F | F | F |
| 40 | T | T | T | F | T | T | F | T | T | F | T | T | T |

| OBS. # | VT8_ PRG | VT8_ USR | VT9_ DES | VT9_ PRG | VT9_ USR | VT10_ DES | VT10_ PRG | VT10_ USR | VT11_ DES | VT11_ PRG | VT11_ USR | VS1_ DES |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 1 | T | F | F | F | T | F | F | F | T | T | T | F |
| 2 | . | . | . | . | . | . | . | . | . | . | . | . |
| 3 | . | . | . | . | . | . | . | . | . | . | . | . |
| 4 | F | F | F | F | F | T | F | F | F | F | T | F |
| 5 | F | F | F | F | F | F | F | F | F | F | F | F |
| 6 | F | F | F | F | F | F | F | F | F | F | F | F |
| 7 | T | F | T | T | F | T | T | T | F | F | F | T |
| 8 | F | T | F | F | F | T | F | F | F | F | F | F |
| 9 | . | . | . | . | . | . | . | . | . | . | . | . |
| 10 | . | . | . | . | . | . | . | . | . | . | . | . |
| 11 | F | T | F | F | F | T | F | F | F | F | F | F |
| 12 | . | . | . | . | . | . | . | . | . | . | . | . |
| 13 | F | F | T | F | F | F | F | F | F | F | F | F |
| 14 | F | F | F | F | F | F | F | F | F | F | T | F |
| 15 | F | F | F | F | F | F | F | F | F | F | T | F |
| 16 | F | T | F | F | F | F | F | F | F | F | F | F |
| 17 | F | T | F | F | F | T | T | T | F | F | T | F |
| 18 | . | . | . | . | . | . | . | . | . | . | . | . |
| 19 | F | F | F | F | F | T | F | F | F | T | F | F |
| 20 | T | F | F | T | F | T | F | F | F | T | F | T |
| 21 | T | T | F | F | F | F | F | F | F | F | T | F |
| 22 | F | F | F | F | F | F | F | F | F | F | F | F |
| 23 | F | F | F | F | F | F | F | F | F | T | T | F |
| 24 | T | T | F | F | F | F | F | F | F | T | T | F |
| 25 | T | T | F | F | T | F | F | T | F | F | T | T |
| 26 | T | T | F | F | F | F | F | F | T | F | T | F |
| 27 | F | F | F | F | F | T | F | F | F | F | F | F |
| 28 | F | F | F | F | F | F | F | F | F | F | F | F |
| 29 | F | F | F | F | F | F | F | F | T | T | T | T |
| 30 | F | F | F | F | F | F | F | F | F | F | T | F |
| 31 | F | F | T | T | T | F | F | T | F | F | T | F |
| 32 | F | F | F | F | F | F | F | F | F | F | T | F |
| 33 | T | T | F | F | F | F | F | F | T | T | T | F |
| 34 | F | F | F | F | F | F | F | F | F | F | T | F |
| 35 | F | T | F | F | F | T | T | T | F | F | F | F |
| 36 | F | F | F | F | F | F | F | F | F | T | T | F |
| 37 | F | T | F | F | F | F | F | F | F | T | F | F |
| 38 | F | F | F | F | F | F | F | F | F | F | F | F |
| 39 | F | T | F | F | F | F | F | F | F | F | F | F |
| 40 | T | T | F | F | F | T | F | F | F | F | T | F |

| OBS. # | VS1-PRG | VS1-USR | VS2-DES | VS2-PRG | VS2-USR | VS3-DES | VS3-PRG | VS3-USR | VS4-DES | VS4-PRG | VS4-USR | VS5-DES | VS5-PRG |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 1 | F | F | F | T | T | T | T | T | F | T | T | F | T |
| 2 | . | . | . | . | . | . | . | . | . | . | . | . | . |
| 3 | . | . | . | . | . | . | . | . | . | . | . | . | . |
| 4 | F | F | F | F | T | F | F | T | F | F | T | F | F |
| 5 | F | F | T | T | F | T | T | F | F | T | F | F | T |
| 6 | T | T | F | T | F | F | F | F | F | T | F | F | F |
| 7 | T | T | T | T | T | T | T | F | T | T | T | T | T |
| 8 | F | F | F | F | F | T | F | F | F | T | T | F | F |
| 9 | . | . | . | . | . | . | . | . | . | . | . | . | . |
| 10 | . | . | . | . | . | . | . | . | . | . | . | . | . |
| 11 | F | F | F | F | F | F | F | F | F | F | T | F | F |
| 12 | . | . | . | . | . | . | . | . | . | . | . | . | . |
| 13 | T | F | T | T | T | T | T | T | F | F | F | T | F |
| 14 | F | F | F | T | T | F | F | T | F | T | T | F | F |
| 15 | F | F | F | T | T | F | F | T | F | T | F | F | F |
| 16 | F | F | F | F | T | F | F | F | F | T | T | F | F |
| 17 | T | T | T | T | T | F | F | T | F | T | T | F | T |
| 18 | . | . | . | . | . | . | . | . | . | . | . | . | . |
| 19 | F | T | T | T | T | T | T | T | F | T | T | F | T |
| 20 | T | F | F | T | T | F | T | T | F | T | F | F | F |
| 21 | F | F | F | F | T | F | F | F | F | T | T | F | F |
| 22 | F | F | F | F | F | F | F | F | F | F | F | F | F |
| 23 | F | F | T | T | T | F | F | F | F | T | T | F | T |
| 24 | F | F | F | F | F | F | F | F | F | T | T | F | F |
| 25 | T | T | T | T | T | T | T | T | F | F | F | F | F |
| 26 | T | T | F | F | F | F | F | F | F | T | F | F | F |
| 27 | F | F | T | T | T | F | F | F | F | T | F | F | F |
| 28 | F | F | F | F | F | F | F | F | F | F | F | F | F |
| 29 | T | T | F | T | T | F | F | F | F | F | F | F | F |
| 30 | F | F | F | F | F | F | F | F | F | T | T | F | F |
| 31 | F | T | F | F | F | F | F | T | T | F | F | F | T |
| 32 | F | F | F | F | F | T | F | F | T | F | F | T | T |
| 33 | F | F | T | T | T | T | T | T | T | T | T | F | T |
| 34 | F | F | F | F | F | F | F | T | F | F | T | F | F |
| 35 | F | F | F | T | T | F | T | F | F | F | T | F | F |
| 36 | F | F | F | T | F | T | F | F | F | T | F | F | F |
| 37 | T | F | F | T | F | F | T | F | F | T | F | F | T |
| 38 | F | F | F | F | F | F | F | F | F | T | T | F | F |
| 39 | T | T | F | F | F | F | F | F | F | F | T | F | F |
| 40 | F | F | F | F | F | F | F | F | F | T | T | F | F |

| OBS. | VS5_ | VS6_ | VS6_ | VS6_ | VS7_ | VS7_ | VS7_ | VS8_ | VS8_ | VS8_ | VS9_ | VS9_ | VS9_ |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| # | USR | DES | PRG | USR | DES | PRG | USR | DES | PRG | USR | DES | PRG | USR |
| 1 | T | F | F | F | F | F | F | F | F | F | F | F | T |
| 2 | . | . | . | . | . | . | . | . | . | . | . | . | . |
| 3 | . | . | . | . | . | . | . | . | . | . | . | . | . |
| 4 | F | F | F | F | F | F | F | F | F | F | F | F | T |
| 5 | F | F | F | F | F | T | F | F | T | F | T | F | F |
| 6 | F | F | F | F | F | T | F | F | F | F | F | F | F |
| 7 | T | T | T | T | T | T | F | F | T | T | F | T | T |
| 8 | F | F | F | F | F | F | F | F | F | F | F | T | F |
| 9 | . | . | . | . | . | . | . | . | . | . | . | . | . |
| 10 | . | . | . | . | . | . | . | . | . | . | . | . | . |
| 11 | F | F | F | F | F | F | F | F | F | F | F | T | F |
| 12 | . | . | . | . | . | . | . | . | . | . | . | . | . |
| 13 | T | F | F | F | F | T | F | T | T | T | T | T | T |
| 14 | T | F | F | F | F | F | T | F | F | T | T | F | F |
| 15 | F | F | F | F | F | T | F | F | F | F | T | F | F |
| 16 | F | F | F | F | F | T | F | F | F | F | T | F | F |
| 17 | T | F | F | F | F | F | F | F | F | F | F | F | T |
| 18 | . | . | . | . | . | . | . | . | . | . | . | . | . |
| 19 | T | F | F | F | F | T | F | F | F | T | F | F | T |
| 20 | T | F | F | T | F | F | T | F | T | T | F | T | F |
| 21 | T | F | F | F | F | F | F | F | F | T | F | F | F |
| 22 | F | F | F | F | F | F | F | F | F | T | F | F | F |
| 23 | T | F | F | F | F | T | T | F | T | F | F | F | F |
| 24 | F | F | F | F | F | F | F | F | T | T | T | T | T |
| 25 | F | T | F | T | F | F | T | T | T | T | T | T | T |
| 26 | T | F | F | F | F | T | F | F | T | T | F | F | T |
| 27 | T | F | F | F | F | F | F | F | F | F | F | F | T |
| 28 | F | F | F | F | F | T | F | F | F | F | F | F | F |
| 29 | F | T | T | T | T | T | T | F | F | F | F | T | F |
| 30 | F | F | F | F | F | F | F | F | T | T | F | T | T |
| 31 | T | F | F | F | F | F | T | F | F | T | T | T | T |
| 32 | T | F | F | F | T | T | F | F | T | T | T | T | T |
| 33 | T | F | F | F | T | T | F | T | T | F | T | T | T |
| 34 | T | F | F | F | F | F | T | F | F | T | F | F | T |
| 35 | T | F | F | F | F | F | F | F | F | T | F | F | T |
| 36 | F | F | F | F | F | T | T | F | T | T | F | F | F |
| 37 | F | F | F | T | F | T | F | F | F | F | F | F | F |
| 38 | T | F | T | T | F | F | F | F | T | T | F | F | F |
| 39 | F | F | F | T | F | T | T | F | T | T | T | T | T |
| 40 | T | F | F | T | F | F | F | F | T | T | F | F | T |

| OBS. # | VH1_DES | VH1_PRG | VH1_USR | VH2_DES | VH2_PRG | VH2_USR | VH3_DES | VH3_PRG | VH3_USR | VH4_DES | VH4_PRG | VH4_USR | VH5_DES |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 1 | F | F | T | T | T | T | F | T | T | F | F | T | T |
| 2 | . | . | . | . | . | . | . | . | . | . | . | . | . |
| 3 | . | . | . | . | . | . | . | . | . | . | . | . | . |
| 4 | F | F | T | F | F | T | F | F | T | F | F | T | F |
| 5 | T | F | F | T | T | F | F | T | T | T | F | F | F |
| 6 | F | F | F | F | F | T | F | F | F | F | F | F | F |
| 7 | F | F | T | F | F | T | F | F | T | F | T | T | F |
| 8 | F | F | T | F | F | T | F | F | T | F | F | F | F |
| 9 | . | . | . | . | . | . | . | . | . | . | . | . | . |
| 10 | . | . | . | . | . | . | . | . | . | . | . | . | . |
| 11 | F | F | F | F | F | T | F | F | T | F | F | F | F |
| 12 | . | . | . | . | . | . | . | . | . | . | . | . | . |
| 13 | T | F | F | F | F | T | T | F | F | F | T | T | T |
| 14 | T | F | F | T | F | T | F | F | T | T | T | T | F |
| 15 | T | F | F | T | F | F | F | F | F | T | T | T | F |
| 16 | T | F | F | T | F | F | F | T | F | T | T | T | F |
| 17 | T | T | T | F | T | T | F | T | T | T | T | T | F |
| 18 | . | . | . | . | . | . | . | . | . | . | . | . | . |
| 19 | F | T | F | T | T | T | F | F | T | T | T | T | F |
| 20 | F | T | F | T | F | F | T | T | F | T | T | T | F |
| 21 | F | T | T | T | F | F | F | F | F | F | T | F | F |
| 22 | F | F | T | F | F | T | F | F | F | F | F | T | F |
| 23 | F | T | T | F | T | F | F | F | F | F | T | T | T |
| 24 | F | F | F | F | F | T | F | F | T | F | F | T | F |
| 25 | T | F | T | T | F | T | T | T | T | T | T | T | T |
| 26 | T | T | T | T | F | F | T | F | F | F | F | T | F |
| 27 | F | F | T | T | T | F | F | F | F | F | F | F | F |
| 28 | F | T | T | F | F | T | F | F | F | F | F | F | F |
| 29 | F | F | T | F | F | T | F | T | T | F | F | F | T |
| 30 | F | F | T | F | F | T | F | F | T | F | F | T | F |
| 31 | T | F | T | F | F | F | F | F | F | F | F | T | F |
| 32 | F | F | T | F | T | F | F | F | F | F | T | T | F |
| 33 | F | F | T | F | F | T | F | F | F | T | T | T | T |
| 34 | F | F | T | F | F | T | F | F | T | F | F | T | F |
| 35 | F | F | T | F | F | F | F | F | T | F | F | F | F |
| 36 | F | F | T | F | F | T | F | F | T | F | F | T | F |
| 37 | F | T | T | F | T | T | F | T | T | F | T | F | F |
| 38 | F | F | F | F | F | T | F | F | F | T | T | F | F |
| 39 | F | F | T | T | T | T | T | T | T | F | F | F | F |
| 40 | F | F | T | T | F | T | F | F | F | F | T | T | F |

| OBS. # | VH5-PRG | VH5-USR | VH6-DES | VH6-PRG | VH6-USR | VH7-DES | VH7-PRG | VH7-USR | TEC-NUM | VAL-NUM | VT1-GRP | VT2-GRP | VT3-GRP |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 1 | F | F | F | F | T | F | F | T | 15 | 35 | 2 | 0 | 0 |
| 2 | . | . | . | . | . | . | . | . | 37 | . | . | . | . |
| 3 | . | . | . | . | . | . | . | . | 35 | . | . | . | . |
| 4 | F | T | F | F | T | F | F | T | 27 | 20 | 2 | 1 | 1 |
| 5 | F | T | F | F | F | F | F | F | 30 | 23 | 1 | 0 | 0 |
| 6 | F | T | F | F | F | F | F | F | 46 | 15 | 2 | 2 | 0 |
| 7 | T | T | F | F | T | F | F | F | 25 | 51 | 3 | 2 | 0 |
| 8 | F | F | F | F | T | F | F | F | 22 | 16 | 1 | 1 | 1 |
| 9 | . | . | . | . | . | . | . | . | 13 | . | . | . | . |
| 10 | . | . | . | . | . | . | . | . | 30 | . | . | . | . |
| 11 | F | F | F | F | T | F | F | F | 21 | 14 | 2 | 1 | 2 |
| 12 | . | . | . | . | . | . | . | . | 24 | . | . | . | . |
| 13 | T | T | F | F | F | F | F | F | 45 | 34 | 1 | 0 | 0 |
| 14 | F | T | F | F | F | F | F | F | 32 | 24 | 1 | 0 | 0 |
| 15 | F | T | F | F | F | F | F | F | 30 | 18 | 1 | 0 | 0 |
| 16 | F | T | T | T | F | F | F | F | 31 | 20 | 1 | 0 | 0 |
| 17 | T | T | T | T | T | F | F | T | 22 | 46 | 2 | 2 | 2 |
| 18 | . | . | . | . | . | . | . | . | 24 | . | . | . | . |
| 19 | F | F | T | T | T | F | F | F | 24 | 39 | 1 | 0 | 3 |
| 20 | F | T | T | T | T | F | F | T | 25 | 38 | 1 | 1 | 1 |
| 21 | F | F | F | T | T | F | F | T | 21 | 26 | 2 | 0 | 3 |
| 22 | F | F | F | F | T | F | F | T | 26 | 9 | 2 | 0 | 1 |
| 23 | F | T | F | T | T | F | F | T | 20 | 33 | 3 | 0 | 2 |
| 24 | F | F | F | F | F | F | F | F | 43 | 22 | 3 | 1 | 0 |
| 25 | T | T | F | F | T | F | F | F | 18 | 48 | 2 | 2 | 2 |
| 26 | F | F | F | F | F | F | F | F | 60 | 23 | 1 | 0 | 0 |
| 27 | F | T | F | F | T | F | F | F | 23 | 25 | 2 | 3 | 0 |
| 28 | F | F | F | F | T | F | F | T | 43 | 13 | 2 | 0 | 0 |
| 29 | T | T | F | F | F | F | F | F | 31 | 38 | 3 | 3 | 0 |
| 30 | F | F | F | F | T | F | F | T | 49 | 23 | 2 | 0 | 0 |
| 31 | F | T | F | F | T | F | F | T | 41 | 31 | 3 | 0 | 0 |
| 32 | F | F | F | F | T | F | F | T | 35 | 29 | 1 | 1 | 1 |
| 33 | T | T | T | T | T | F | F | F | 49 | 50 | 3 | 3 | 0 |
| 34 | F | F | F | F | T | F | F | T | 51 | 20 | 1 | 1 | 1 |
| 35 | F | F | F | F | F | F | F | F | 16 | 22 | 3 | 0 | 0 |
| 36 | F | T | F | F | T | F | F | F | 17 | 24 | 3 | 2 | 0 |
| 37 | F | T | F | T | T | F | F | T | 56 | 28 | 2 | 1 | 1 |
| 38 | F | F | F | F | F | F | F | F | 21 | 19 | 2 | 2 | 0 |
| 39 | F | T | F | F | T | F | F | T | 60 | 33 | 3 | 3 | 0 |
| 40 | F | F | F | F | F | F | F | F | 0 | 30 | 2 | 2 | 0 |

| OBS. # | VT4_GRP | VT5_GRP | VT6_GRP | VT7_GRP | VT8_GRP | VT9_GRP | VT10_GRP | VT11_GRP | VS1_GRP | VS2_GRP | VS3_GRP | VS4_GRP | VS5_GRP |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 1 | 2 | 2 | 2 | 2 | 1 | 1 | 0 | 3 | 0 | 2 | 3 | 2 | 2 |
| 2 | . | . | . | . | . | . | . | . | . | . | . | . | . |
| 3 | . | . | . | . | . | . | . | . | . | . | . | . | . |
| 4 | 1 | 0 | 1 | 1 | 0 | 0 | 1 | 1 | 0 | 1 | 1 | 1 | 0 |
| 5 | 0 | 2 | 2 | 1 | 1 | 0 | 0 | 0 | 0 | 2 | 2 | 1 | 1 |
| 6 | 0 | 1 | 1 | 2 | 0 | 0 | 0 | 0 | 2 | 1 | 0 | 1 | 0 |
| 7 | 0 | 3 | 3 | 2 | 2 | 2 | 3 | 0 | 3 | 3 | 2 | 3 | 3 |
| 8 | 0 | 2 | 1 | 0 | 1 | 0 | 1 | 0 | 0 | 0 | 1 | 2 | 0 |
| 9 | . | . | . | . | . | . | . | . | . | . | . | . | . |
| 10 | . | . | . | . | . | . | . | . | . | . | . | . | . |
| 11 | 0 | 1 | 1 | 0 | 1 | 0 | 1 | 0 | 0 | 0 | 0 | 1 | 0 |
| 12 | . | . | . | . | . | . | . | . | . | . | . | . | . |
| 13 | 0 | 3 | 1 | 3 | 1 | 1 | 0 | 0 | 1 | 3 | 3 | 0 | 2 |
| 14 | 0 | 2 | 2 | 1 | 0 | 0 | 0 | 1 | 0 | 2 | 1 | 2 | 1 |
| 15 | 0 | 1 | 2 | 1 | 0 | 0 | 0 | 1 | 0 | 2 | 1 | 1 | 0 |
| 16 | 0 | 2 | 1 | 1 | 1 | 0 | 0 | 0 | 0 | 1 | 0 | 2 | 0 |
| 17 | 2 | 3 | 2 | 1 | 1 | 0 | 3 | 1 | 2 | 3 | 1 | 2 | 2 |
| 18 | . | . | . | . | . | . | . | . | . | . | . | . | . |
| 19 | 1 | 1 | 2 | 3 | 1 | 0 | 1 | 1 | 1 | 3 | 3 | 2 | 2 |
| 20 | 2 | 1 | 1 | 1 | 2 | 1 | 1 | 1 | 2 | 2 | 2 | 1 | 1 |
| 21 | 0 | 2 | 1 | 2 | 3 | 0 | 0 | 1 | 0 | 1 | 0 | 2 | 1 |
| 22 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 23 | 2 | 0 | 1 | 2 | 1 | 0 | 0 | 2 | 0 | 3 | 0 | 2 | 2 |
| 24 | 1 | 1 | 2 | 0 | 2 | 0 | 0 | 2 | 0 | 0 | 0 | 2 | 0 |
| 25 | 1 | 1 | 1 | 2 | 2 | 1 | 1 | 1 | 3 | 3 | 3 | 0 | 0 |
| 26 | 0 | 1 | 2 | 1 | 2 | 0 | 0 | 2 | 2 | 0 | 0 | 1 | 1 |
| 27 | 2 | 3 | 1 | 2 | 0 | 0 | 1 | 0 | 0 | 3 | 0 | 1 | 1 |
| 28 | 0 | 2 | 0 | 3 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 29 | 0 | 3 | 3 | 3 | 1 | 0 | 0 | 3 | 3 | 2 | 0 | 0 | 0 |
| 30 | 2 | 2 | 2 | 2 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 2 | 0 |
| 31 | 3 | 2 | 2 | 0 | 0 | 3 | 1 | 1 | 1 | 0 | 1 | 1 | 2 |
| 32 | 2 | 2 | 3 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 1 | 1 | 3 |
| 33 | 3 | 0 | 3 | 3 | 3 | 0 | 0 | 3 | 0 | 3 | 3 | 3 | 2 |
| 34 | 1 | 1 | 1 | 1 | 0 | 0 | 0 | 1 | 0 | 0 | 1 | 1 | 1 |
| 35 | 0 | 3 | 3 | 0 | 1 | 0 | 3 | 0 | 0 | 2 | 1 | 1 | 1 |
| 36 | 0 | 1 | 1 | 2 | 0 | 0 | 0 | 2 | 0 | 1 | 1 | 1 | 0 |
| 37 | 1 | 1 | 1 | 1 | 1 | 0 | 0 | 1 | 1 | 1 | 1 | 1 | 1 |
| 38 | 1 | 2 | 2 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 2 | 1 |
| 39 | 0 | 3 | 2 | 0 | 1 | 0 | 0 | 0 | 2 | 0 | 0 | 1 | 0 |
| 40 | 3 | 2 | 2 | 2 | 3 | 0 | 1 | 1 | 0 | 0 | 0 | 2 | 1 |

| OBS. # | VS6_GRP | VS7_GRP | VS8_GRP | VS9_GRP | VH1_GRP | VH2_GRP | VH3_GRP | VH4_GRP | VH5_GRP | VH6_GRP | VH7_GRP |
|---|---|---|---|---|---|---|---|---|---|---|---|
| 1 | 0 | 0 | 0 | 1 | 1 | 3 | 2 | 1 | 1 | 1 | 1 |
| 2 | . | . | . | . | . | . | . | . | . | . | . |
| 3 | . | . | . | . | . | . | . | . | . | . | . |
| 4 | 0 | 0 | 0 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| 5 | 0 | 1 | 1 | 1 | 1 | 2 | 2 | 1 | 1 | 0 | 0 |
| 6 | 0 | 1 | 0 | 0 | 0 | 1 | 0 | 0 | 1 | 0 | 0 |
| 7 | 3 | 2 | 2 | 2 | 1 | 1 | 1 | 2 | 2 | 1 | 0 |
| 8 | 0 | 0 | 0 | 1 | 1 | 1 | 1 | 0 | 0 | 1 | 0 |
| 9 | . | . | . | . | . | . | . | . | . | . | . |
| 10 | . | . | . | . | . | . | . | . | . | . | . |
| 11 | 0 | 0 | 0 | 1 | 0 | 1 | 1 | 0 | 0 | 1 | 0 |
| 12 | . | . | . | . | . | . | . | . | . | . | . |
| 13 | 0 | 1 | 3 | 3 | 1 | 1 | 1 | 2 | 3 | 0 | 0 |
| 14 | 0 | 1 | 1 | 1 | 1 | 2 | 1 | 3 | 1 | 0 | 0 |
| 15 | 0 | 1 | 0 | 1 | 1 | 1 | 0 | 3 | 1 | 0 | 0 |
| 16 | 0 | 1 | 0 | 1 | 1 | 1 | 1 | 3 | 1 | 2 | 0 |
| 17 | 0 | 0 | 0 | 1 | 3 | 2 | 2 | 3 | 2 | 3 | 1 |
| 18 | . | . | . | . | . | . | . | . | . | . | . |
| 19 | 0 | 1 | 1 | 1 | 1 | 3 | 1 | 3 | 0 | 3 | 0 |
| 20 | 1 | 1 | 2 | 1 | 1 | 1 | 2 | 3 | 1 | 3 | 1 |
| 21 | 0 | 0 | 1 | 0 | 2 | 1 | 0 | 1 | 0 | 2 | 1 |
| 22 | 0 | 0 | 1 | 0 | 1 | 1 | 0 | 1 | 0 | 1 | 1 |
| 23 | 0 | 2 | 1 | 0 | 2 | 1 | 0 | 2 | 2 | 2 | 1 |
| 24 | 0 | 0 | 2 | 3 | 0 | 1 | 1 | 1 | 0 | 0 | 0 |
| 25 | 2 | 1 | 3 | 3 | 2 | 2 | 3 | 3 | 3 | 1 | 0 |
| 26 | 0 | 1 | 2 | 1 | 3 | 1 | 1 | 1 | 0 | 0 | 0 |
| 27 | 0 | 0 | 0 | 1 | 1 | 2 | 0 | 0 | 1 | 1 | 0 |
| 28 | 0 | 1 | 0 | 0 | 2 | 1 | 0 | 0 | 0 | 1 | 1 |
| 29 | 3 | 3 | 0 | 1 | 1 | 1 | 2 | 0 | 3 | 0 | 0 |
| 30 | 0 | 0 | 2 | 2 | 1 | 1 | 1 | 1 | 0 | 1 | 1 |
| 31 | 0 | 1 | 1 | 3 | 2 | 0 | 0 | 1 | 1 | 1 | 1 |
| 32 | 0 | 2 | 2 | 3 | 1 | 1 | 0 | 2 | 0 | 1 | 1 |
| 33 | 0 | 2 | 2 | 3 | 1 | 1 | 0 | 3 | 3 | 3 | 0 |
| 34 | 0 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 0 | 1 | 1 |
| 35 | 0 | 0 | 1 | 1 | 1 | 0 | 1 | 0 | 0 | 0 | 0 |
| 36 | 0 | 2 | 2 | 0 | 1 | 1 | 1 | 1 | 1 | 1 | 0 |
| 37 | 1 | 1 | 0 | 0 | 2 | 2 | 2 | 1 | 1 | 2 | 1 |
| 38 | 2 | 0 | 2 | 0 | 0 | 1 | 0 | 2 | 0 | 0 | 0 |
| 39 | 1 | 2 | 2 | 3 | 1 | 3 | 3 | 0 | 1 | 1 | 1 |
| 40 | 1 | 0 | 2 | 1 | 1 | 2 | 0 | 2 | 0 | 0 | 0 |

# Appendix F: Equations Resulting from Testing the Hypotheses

The following table lists the accepted hypotheses, and their models:

**Systems Features Hypotheses:**

$H_5$: Verification depends on the **technical complexity** of a system.

$$VER\% = 102.81 + 5.3 * T7A\_CAT - 10.5 * T10\_CAT \tag{1}$$

$$VER\% = 105.66 + 4.3 * T7A\_CAT - 11.6 * T10\_CAT + 6.2 * T2\_CAT - 9.2 * T6\_CAT \tag{2}$$

$H_5$: Validation depends on the **technical complexity** of a system.

$$VAL\% = 79.7 + 6.0 * T7A\_CAT - 5.4 * T7B\_CAT \tag{3}$$

$H_6$: Verification depends on the level of **human involvement** in a system.

$$VER\% = 89.5 + 6.2 * H2D\_DES - 3.2 * H3D\_PRG - 0.15 * H8\_TOTAL \tag{4}$$

$$VER\% = 96.9 + 7.6 * H2D\_DES - 3.9 * H3D\_PRG - 0.17 * H8\_TOTAL - 11.1 * H9\_EXPRT \tag{5}$$

$H_6$: Validation depends on the level of **human involvement** in a system.

$$VAL\% = 70.3 + 9.7 * H1D\_DEFN - 6.8 * H5D\_IMPL \tag{6}$$

$$VAL\% = 77.2 + 9.9 * H1D\_DEFN - 7.5 * H5D\_IMPL - 1.9 * H4U\_TEST \tag{7}$$

$$VAL\% = 90.4 + 9.7 * H1D\_DEFN - 7.9 * H5D\_IMPL - 2.1 * H4U\_TEST - 17.9 * H9\_EXPRT \tag{8}$$

$$VAL\% = 102.5 + 14.2 * H2U\_DES - 7.8 * H5D\_IMPL - 3.1 * H4U\_TEST - 16.0 * H9\_EXPRT - 10.2 * H3U\_PRG \tag{9}$$

$H_9$: Verification depends on the degree to which a system is **observable**.

REJECTED – No model containing all significant variables.

$H_9$: Validation depends on the degree to which a system is **observable**.

REJECTED - $R^2 = 13\%$.

$H_{10}$: Verification depends on the combination of **complexity, human involvement & observability** of a system.

$$\begin{aligned}
VER\% &= 92.9 - 0.15 * H8\_TOTAL && (10)\\
VER\% &= 90.6 - 0.15 * H8\_TOTAL + 6.6 * H2D\_DES - 3.4 * H3D\_PRG && (11)\\
VER\% &= 97.1 - 0.17 * H8\_TOTAL + 7.7 * H2D\_DES - 4.1 * H3D\_PRG - 9.9 * H9\_EXPRT && (12)
\end{aligned}$$

$H_{10}$: Validation depends on the combination of **complexity, human involvement & observability** of a system.

$$\begin{aligned}
VAL\% &= 110.8 + 11.9 H2D\_DES - 15.7 * T1\_CAT && (13)\\
VAL\% &= 107.0 + 13.7 * H2D\_DES - 12.4 * T1\_CAT - 6.2 * T3\_CAT && (14)\\
VAL\% &= 115.3 + 13.7 * H2D\_DES - 12.5 * T1\_CAT - 6.2 * T3\_CAT - 12.2 * H9\_EXPRT && (15)\\
VAL\% &= 108.5 + 16.8 * H2D\_DES - 8.9 * T1\_CAT - 5.5 * T3\_CAT - 16.2 * H9\_EXPRT\\
&\quad -3.7 * H3D\_PRG && (16)
\end{aligned}$$

**Validation Methods Hypotheses:**

$H_7$: Verification depends on the **methods used to test** a system.

$$\begin{aligned}
VER\% &= 81.5 + 15.7 * VT11\_USR - 26.8 * VH3\_DES && (17)\\
VER\% &= 89.1 + 18.1 * VT11\_USR - 32.8 * VH3\_DES - 12.9 * VT1\_USR && (18)\\
VER\% &= 95.3 + 21.8 * VT11\_USR - 29.9 * VH3\_DES - 18.5 * VT1\_USR - 15.7 * VS9\_DES && (19)\\
VER\% &= 97.0 + 21.1 * VT11\_USR - 36.8 * VH3\_DES - 19.5 * VT1\_USR - 20.6 * VS9\_DES\\
&\quad +21.6 * VS8\_DES && (20)
\end{aligned}$$

$H_7$: Validation depends on the **methods used to test** a system.

$$\begin{aligned}
VAL\% &= 43.2 + 26.7 * VS4\_PRG + 24.1 * VH6\_USR && (21)\\
VAL\% &= 50.7 + 26.6 * VS4\_PRG + 33.8 * VH6\_USR - 22.1 * VT1\_USR && (22)\\
VAL\% &= 56.2 + 28.2 * VS4\_PRG + 37.5 * VH6\_USR - 26.6 * VT1\_USR - 21.2 * VS3\_PRG && (23)\\
VAL\% &= 51.7 + 23.3 * VS4\_PRG + 37.7 * VH6\_USR - 20.7 * VT1\_USR - 24.9 * VS3\_PRG\\
&\quad +17.4 * VH4\_DES && (24)
\end{aligned}$$

$H_{8a}$: Verification depends on the **technical methods** used to test a system.

REJECTED - $R^2 = 21\%$.

$H_{8a}$: Validation depends on the **technical methods** used to test a system.

REJECTED - $R^2 = 19\%$.

$H_{8b}$: Verification depends on the **semi-technical methods** used to test a system.

REJECTED – No model containing all significant variables.

$H_{8b}$: Validation depends on the **semi-technical methods** used to test a system.

REJECTED - $R^2 = 19\%$.

$H_{8c}$: Verification depends on the **human judgment methods** used to test a system.

$$VER\% = 90.2 - 29.2 * VH3\_DES \quad (25)$$

$$VER\% = 83.4 - 34.9 * VH3\_DES + 15.9 * VH1\_PRG + 13.6 * VH4\_USR - 12.0 * VH7\_USR \quad (26)$$

$H_{8c}$: Validation depends on the **human judgment methods** used to test a system.

$$VAL\% = 52.8 + 25.1 * VH4\_DES + 24.6 * VH6\_USR \quad (27)$$

$$VAL\% = 55.9 + 25.9 * VH4\_DES + 25.1 * VH6\_USR - 20.8 * VH5\_DES \quad (28)$$

$H_{11}$: Verification depends on groupings of **test methods** used in systems.

$$VER\% = 85.4 + 10.0 * VT11\_GRP - 9.1 * VH3\_GRP \quad (29)$$

$$\begin{aligned} VER\% = {} & 104.6 + 12.2 * VT11\_GRP - 11.2 * VH3\_GRP - 7.8 * VT1\_GRP + 7.7 * VS6\_GRP \\ & -6.7 * VS8\_GRP \quad (30) \end{aligned}$$

$H_{11}$: Validation depends on groupings of **test methods** used in systems.

$$VAL\% = 82.4 + 10.3 * VS4\_GRP - 12.1 * VT5\_GRP \quad (31)$$

$$VAL\% = 85.7 + 14.1 * VS4\_GRP - 12.3 * VT5\_GRP - 8.4 * VT8\_GRP \quad (32)$$

$$VAL\% = 72.0 + 16.2 * VS4\_GRP - 12.8 * VT5\_GRP - 10.9 * VT8\_GRP + 11.8 * VH1\_GRP \quad (33)$$

**Combined Hypotheses:**

$H_{12}$: Verification depends on summary scores (totals) for **technical complexity, human involvement, observability** & the **test methods** used in a system.

$$VER\% = 92.0 + -0.11 * H8\_TOTAL \tag{34}$$

$H_{12}$: Validation depends on summary scores (totals) for **technical complexity, human involvement, observability** & the **test methods** used in a system.

REJECTED - No model containing all significant variables.

$H_{14}$: Verification depends on the **complexity, human involvement & observability** factors of a system combined with the groups of **test methods** used.

$$VER\% = 85.2 - 0.14 * H8\_TOTAL + 7.7 * VT11\_GRP \tag{35}$$

$$VER\% = 79.3 - 0.14 * H8\_TOTAL + 7.4 * VT11\_GRP + 2.5 * O1\_CAT \tag{36}$$

$$VER\% = 82.7 - 0.15 * H8\_TOTAL + 7.2 * VT11\_GRP + 6.5 * H2D\_DES - 3.2 * H3D\_PGMG \tag{37}$$

$$VER\% = 89.6 - 0.17 * H8\_TOTAL + 7.1 * VT11\_GRP + 8.3 * H2D\_DES - 4.1 * H3D\_PGMG - 11.3 * H9\_EXPRT \tag{38}$$

$H_{14}$: Validation depends on the **complexity, human involvement & observability** factors of a system combined with the groups of **test methods** used.

$$VAL\% = 80.4 + 7.6 * H2D\_DES - 8.4 * T3\_CAT \tag{39}$$

$$VAL\% = 101.0 + 11.9 * H2D\_DES - 7.1 * T3\_CAT - 8.8 * T1\_CAT \tag{40}$$

$$VAL\% = 89.4 + 13.7 * H2D\_DES - 6.8 * T3\_CAT - 4.5 * H3D\_PGMG - 17.7 * H9\_EXPRT \tag{41}$$

$H_4$: Verification depends on the **complexity, human involvement & observability** factors of a system combined with the individual **test methods** used.

$$VER\% = 82.6 + 17.6 * VT11\_USR - 0.14 * H8\_TOTAL \tag{42}$$

$$VER\% = 74.1 + 20.2 * VT11\_USR - 0.14 * H8\_TOTAL + 10.6 * H9\_DEV \tag{43}$$

$$VER\% = 66.2 + 19.6 * VT11\_USR - 0.15 * H8\_TOTAL + 12.8 * H9\_DEV + 2.8 * O1\_CAT \tag{44}$$

$$VER\% = 67.8 + 19.6 * VT11\_USR - 0.15 * H8\_TOTAL + 14.1 * H9\_DEV + 2.5 * O1\_CAT - 8.0 * VH3\_PRG \tag{45}$$

$H_4$: Validation depends on the **complexity, human involvement & observability** factors of a system combined with the individual **test methods** used.

$$VAL\% = 80.4 + 7.6 * H2D\_DES - 8.4 * T3\_CAT \tag{46}$$

$$VAL\% = 101.0 + 11.9 * H2D\_DES - 7.1 * T3\_CAT - 8.8 * T1\_CAT \tag{47}$$

$$VAL\% = 93.7 + 12.1 * H2D\_DES - 7.3 * T3\_CAT - 8.7 * T1\_CAT + 12.1 * VT11\_USR \tag{48}$$

$$\begin{aligned} VAL\% = {} & 101.8 + 12.0 * H2D\_DES - 5.4 * T3\_CAT - 10.6 * T1\_CAT + 18.2 * VH4\_DES \\ & -16.5 * VS3\_PRG\,. \end{aligned} \tag{49}$$

# Appendix G: Variables, Signs and Hypotheses

The following table lists the variables that occur in models resulting from accepted hypotheses, sign

| VARIABLE | | Var. of Int. | Coeff. | $H_5$: Tech. Compl. Ver | Val | $H_6$: Human Involv. Ver | Val | $H_{10}$: Observ'blty Ver | Val | $H_7$: Human Inv. & Obs. Ver | Val |
|---|---|---|---|---|---|---|---|---|---|---|---|
| T1_CAT | Lines of Code | * | – | | | | | | X | | |
| T2_CAT | Hard Disk Space | | + | X | | | | | | | |
| T3_CAT | # of Modules | * | – | | | | | | X | | |
| T6_CAT | Training Days | | – | X | | | | | | | |
| T7A_CAT | Output Pages | * | + | X | X | | | | | | |
| T7B_CAT | Output Screens | | – | | X | | | | | | |
| T10_CAT | Maintenance Cost | | – | X | | | | | | | |
| H1D_DEFN | Devs. in Prob. Defn. | | + | | | | X | | | | |
| H2D_DES | Devs. in Design | * | + | | | X | | X | X | | |
| H3D_PRG | Devs. in Progrmmng. | * | – | | | X | | X | X | | |
| H5D_IMPL | Devs. in Implmntn. | | – | | | | X | | | | |
| H4U_TEST | Users in Testing | | – | | | | X | | | | |
| H2U_DES | Users in Design | | + | | | | X | | | | |
| H3U_PRG | Users in Progrmmng. | | – | | | | X | | | | |
| H8_TOTAL | # People Involved | * | – | | | X | | X | | | |
| H9_EXPRT | # Expert Respondents | * | – | | | X | X | X | X | | |
| H9_DEV | # Dev. Respondents | | + | | | | | | | | |
| O1_CAT | Freq. of Real Event | * | + | | | | | | | | |
| VT1_USR | Prg. Compiles/runs | * | – | | | | | | | X | X |
| VT11_USR | Passes Benchmarks | * | + | | | | | | | X | |
| VT5_GRP | Representative Inputs | | – | | | | | | | | |
| VT8_GRP | Comparison to Other Methods | | – | | | | | | | | |
| VS3_PRG | Accurate variables | * | – | | | | | | | | X |
| VS4_PRG | Comp. to Past Data | * | + | | | | | | | | X |
| VS8_DES | Comp. to Real Events | | + | | | | | | | X | |
| VS9_DES | Model Improvement | | – | | | | | | | X | |
| VS6_GRP | Predict Future Events | | + | | | | | | | | |
| VS8_GRP | Comp. to Real Events | | – | | | | | | | | |
| VH3_PRG | Comparison to Expert Sources | * | – | | | | | | | | |
| VH3_GRP | Comparison to Expert Sources | * | – | | | | | | | | |
| VH1_PRG | First Impressions | * | + | | | | | | | | |
| VH3_DES | Comparison to Expert Sources | * | – | | | | | | | X | |
| VH6_USR | Client Acceptance | * | + | | | | | | | | X |
| VH4_DES | User Participation | * | + | | | | | | | | X |
| VH4_USR | User Participation | | + | | | | | | | | |
| VH7_USR | User Satisfaction | | – | | | | | | | | |
| VH5_DES | Increased Problem Understanding | | – | | | | | | | | |

KEY: '*'-variable found in results for more than one hypothesis. KEY: 'G'-variabl

## otheses

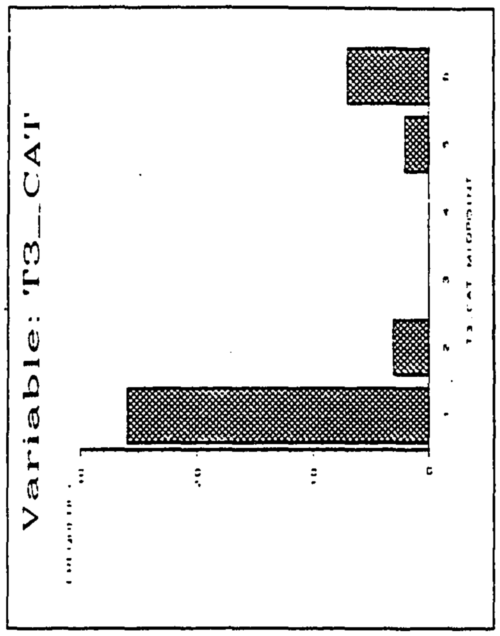
ng from accepted hypotheses, signs of their coefficients and the hypothesis sets in which they occur:

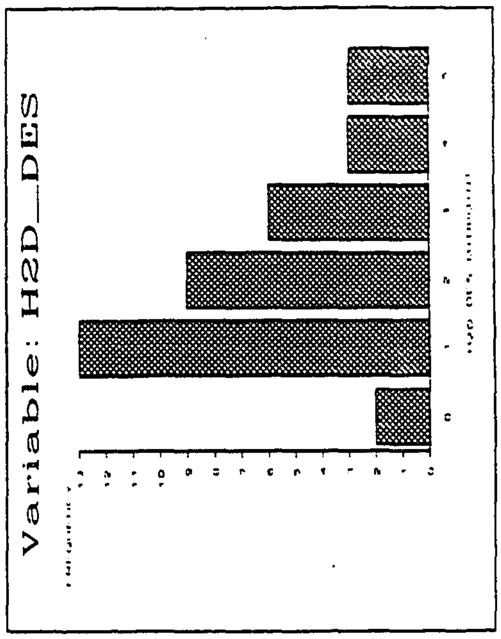
| $H_6$: man Involv. | | $H_{10}$: Observ'blty | | $H_7$: Human Inv. & Obs. | | $H_{8c}$: Human Judg. V&V | | $H_{11}$: Groups of V&V | | $H_{12}$: Sum. Scores (4) | $H_{14}$: C, HI, O + V&V Grps | | $H_4$: C, HI, O + V&V Meths | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| er | Val | Ver | Val | Ver | Val | Ver | Val | Ver | Val | Ver | Ver | Val | Ver | Val |
| | | | X | | | | | | | | | X | | X |
| | | | | | | | | | | | | | | |
| | | | X | | | | | | | | | X | | X |
| | | | | | | | | | | | | | | |
| | | | | | | | | | | | | | | |
| | | | | | | | | | | | | | | |
| | | | | | | | | | | | | | | |
| | X | | | | | | | | | | | | | |
| X | | X | X | | | | | | | | X | X | | X |
| X | | X | X | | | | | | | | X | X | | |
| | X | | | | | | | | | | | | | |
| | X | | | | | | | | | | | | | |
| | X | | | | | | | | | | | | | |
| | X | | | | | | | | | | | | | |
| X | | X | | | | | | | | X | X | | X | |
| X | X | X | X | | | | | | | | | X | | |
| | | | | | | | | | | | | | X | |
| | | | | | | | | | | | X | | X | |
| | | | | X | X | | | G | | | | | | |
| | | | | X | | | | G | | | G | | X | X |
| | | | | | | | | | G | | | | | |
| | | | | | | | | | G | | | | | |
| | | | | | X | | | | | | | | | X |
| | | | | | X | | | | G | | | | | |
| | | | | X | | | | | | | | | | |
| | | | | X | | | | | | | | | | |
| | | | | | | | | G | | | | | | |
| | | | | | | | | G | | | | | | |
| | | | | | | | | | | | | | X | |
| | | | | | | | | G | | | | | | |
| | | | | | | X | | | G | | | | | |
| | | | | X | | X | | G | | | | | | |
| | | | | | X | | X | | | | | | | |
| | | | | | X | | X | | | | | | | X |
| | | | | | | X | | | | | | | | |
| | | | | | | X | | | | | | | | |
| | | | | | | | X | | | | | | | |

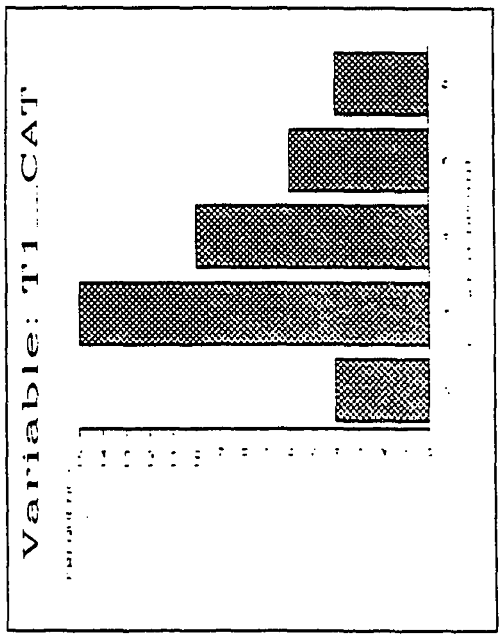
KEY: 'G'–variable is represented in V&V Group, not as individual.

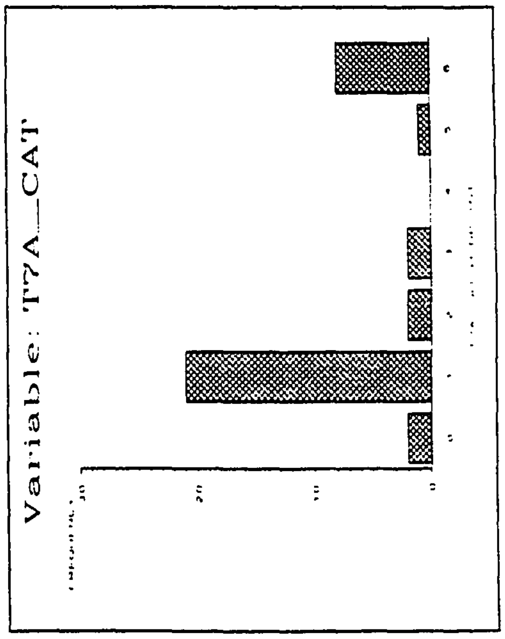
## Appendix H: Histograms of Selected Input Data

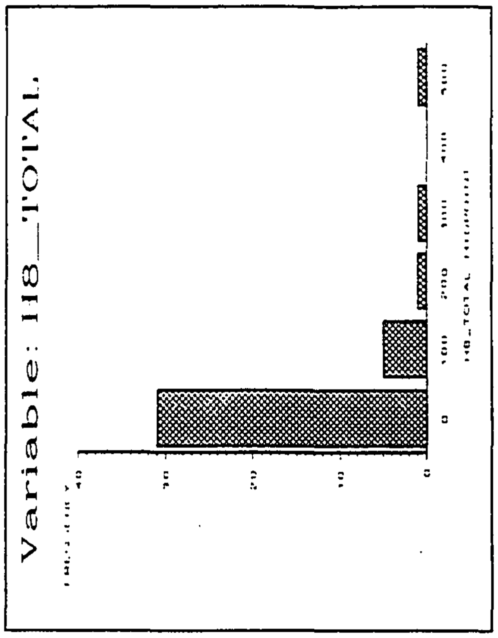
The following are histograms of the distributions of variables representing input data that were found to be the most important variables in equations resulting from accepted hypotheses. Refer to Tables 9a and 9b, page 111-112.

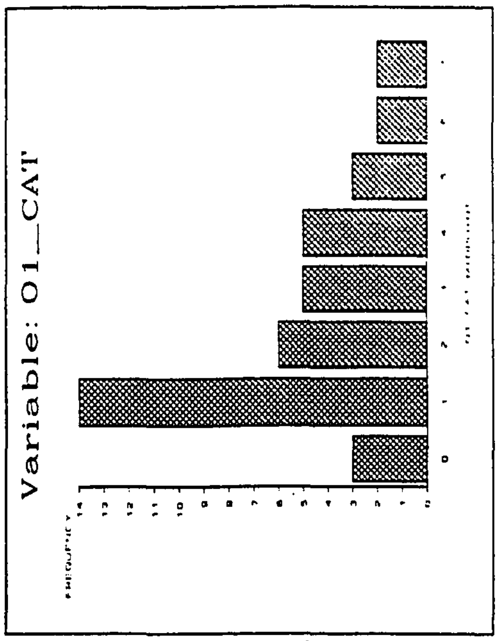


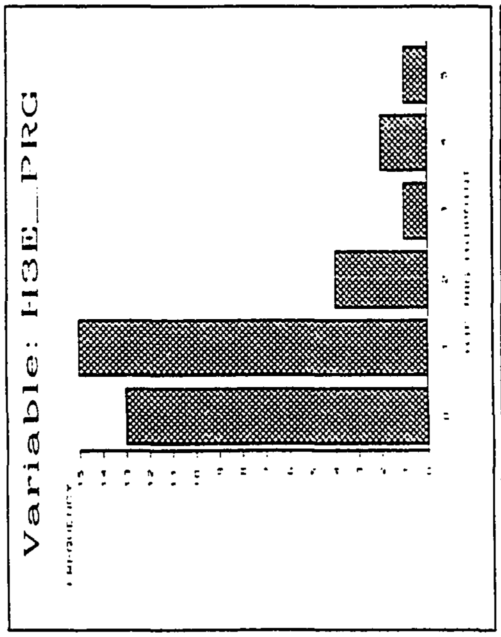


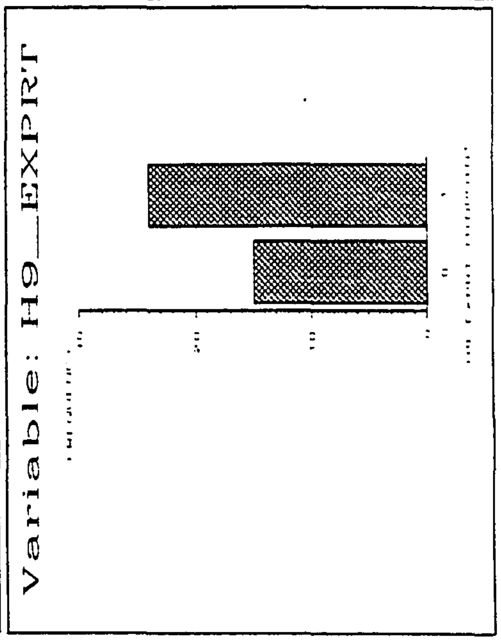


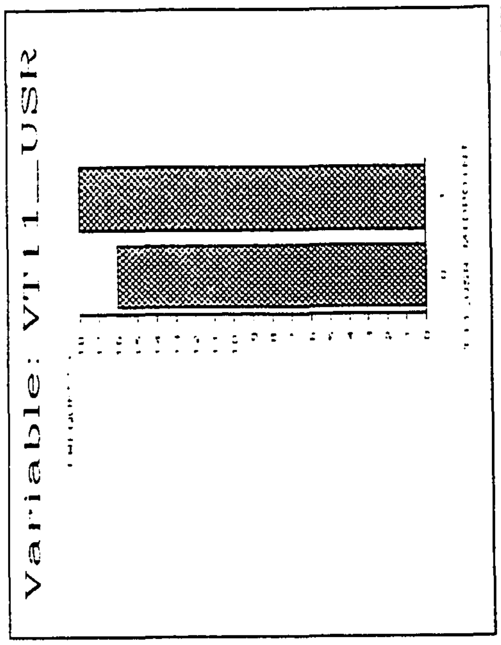


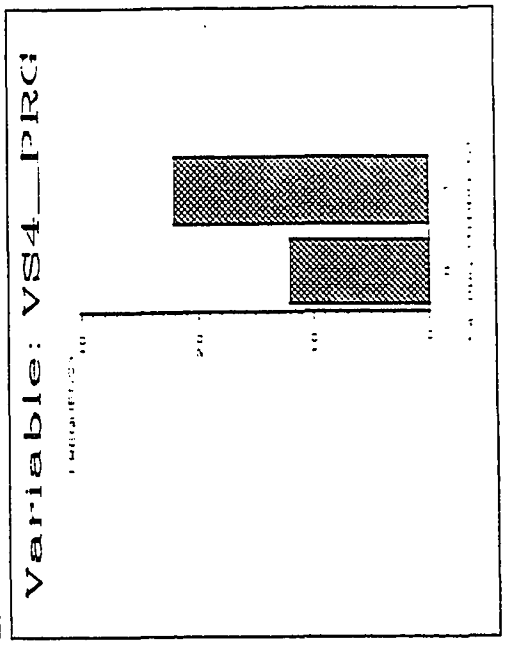

Variable: H8__TOTAL
FREQUENCY
H8_TOTAL MIDPOINT

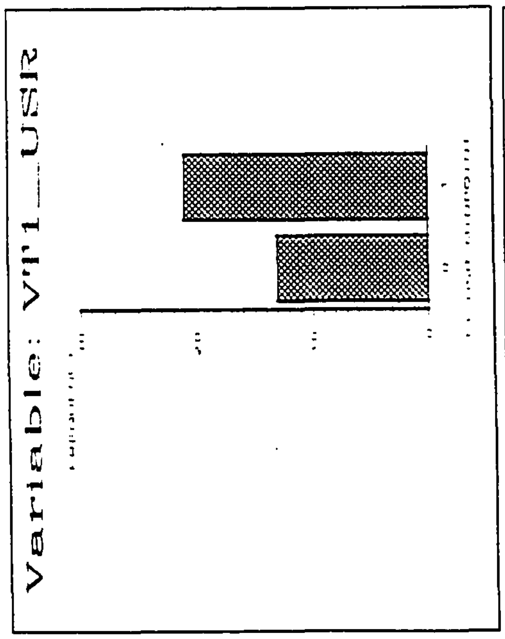

Variable: O1__CAT
FREQUENCY
O1_CAT MIDPOINT

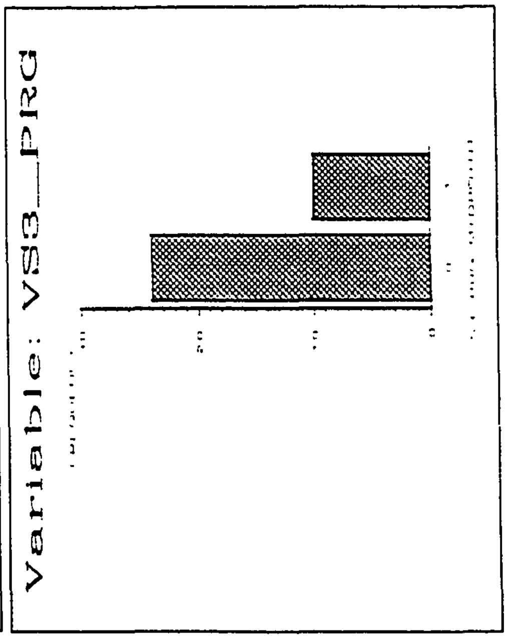

Variable: H3E__PRG
FREQUENCY
H3E_PRG MIDPOINT

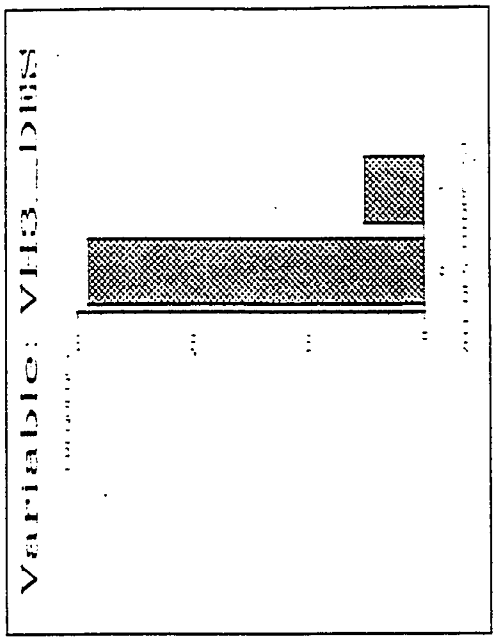

Variable: H9__EXPRT
FREQUENCY
H9_EXPRT MIDPOINT

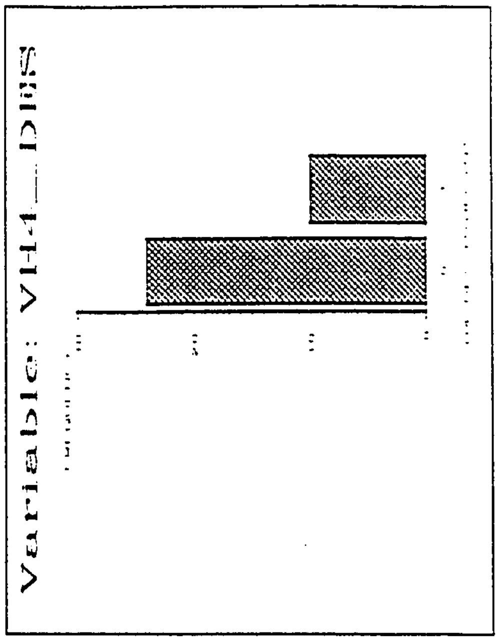

Variable: VT11__USR

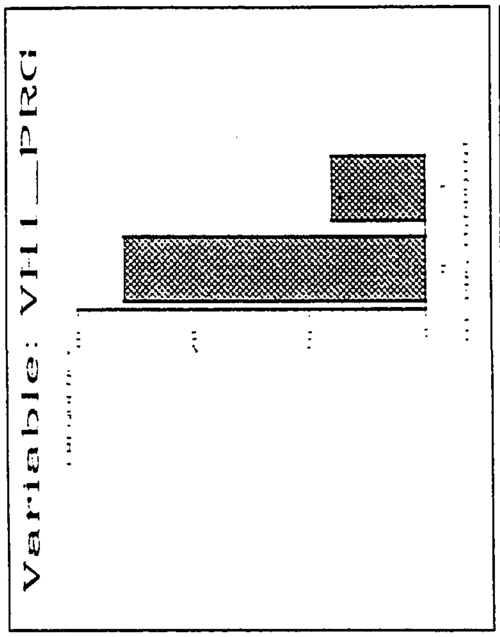

Variable: VS4__PRG

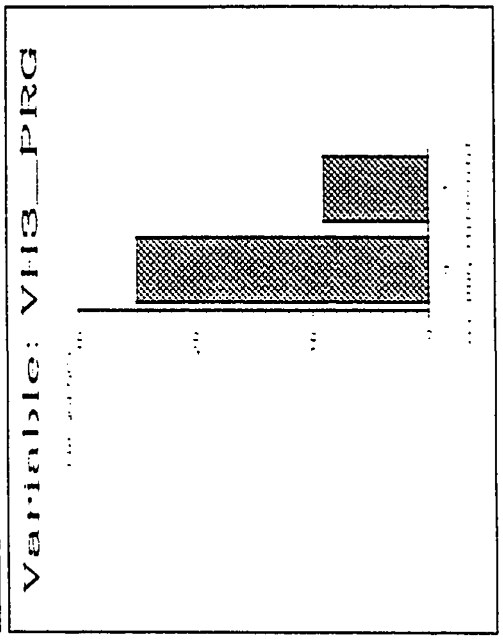

Variable: VT1__USR

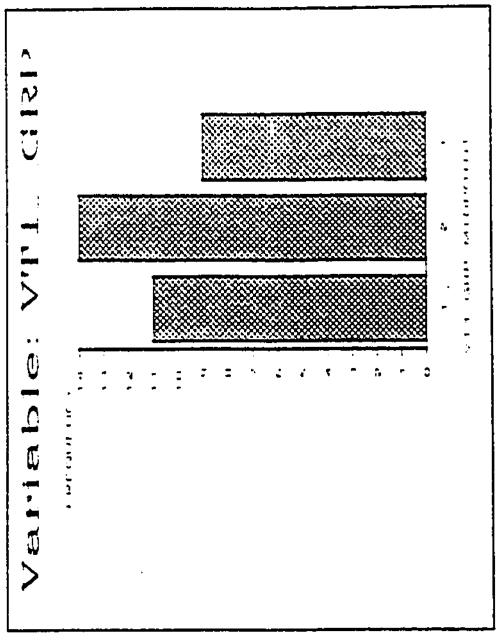

Variable: VS3__PRG

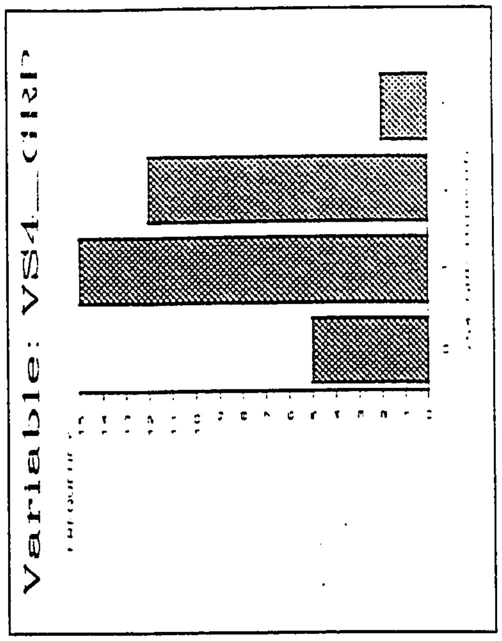

Variable: VH3__DES

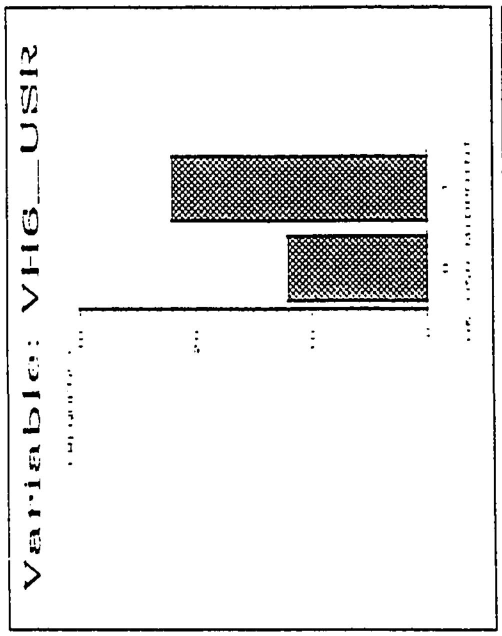

Variable: VH4__DES

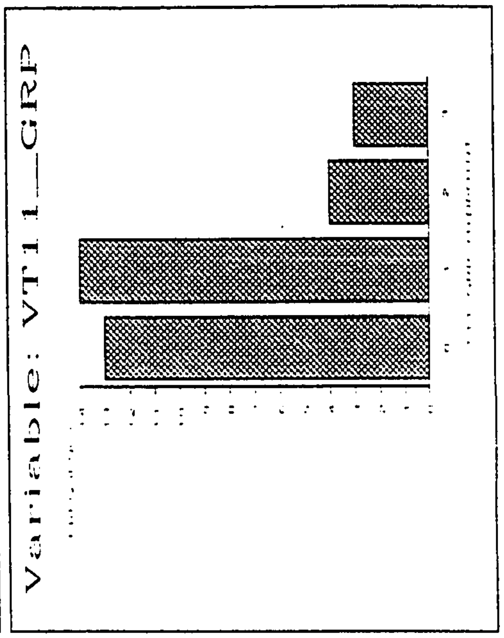

Variable: VH1__PRG

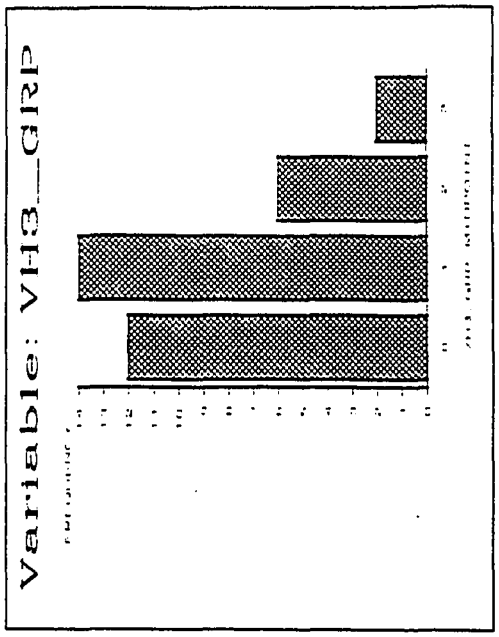

Variable: VH3__PRG

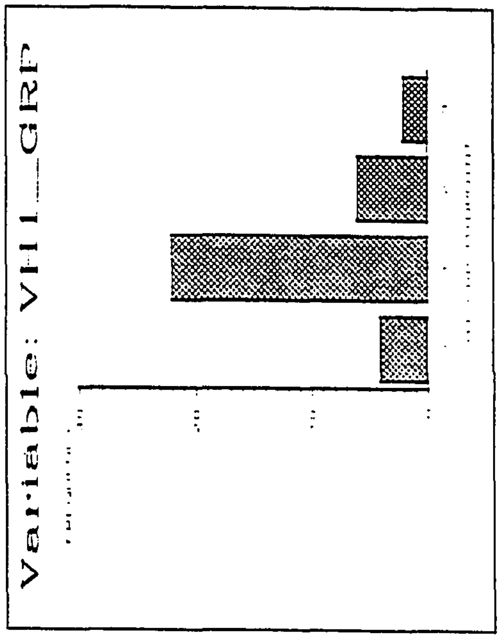

Variable: VT1__GRP


Variable: VS4__GRP


Variable: VH6__USR

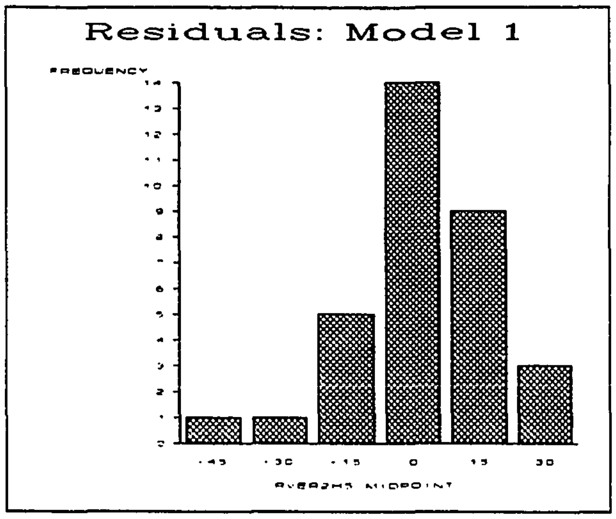

Variable: VT11__GRP

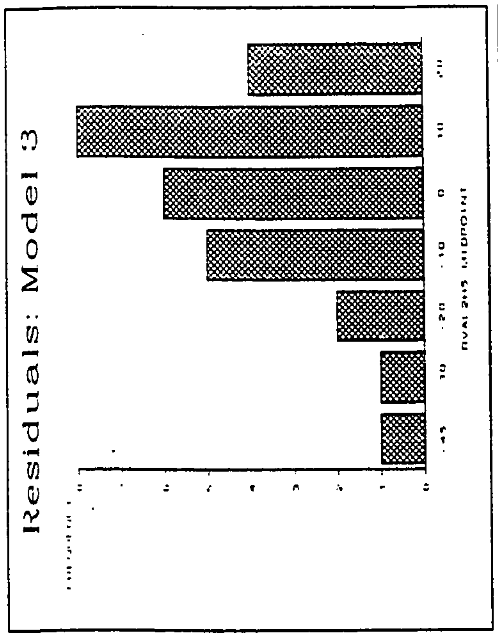

Variable: VH3_GRP

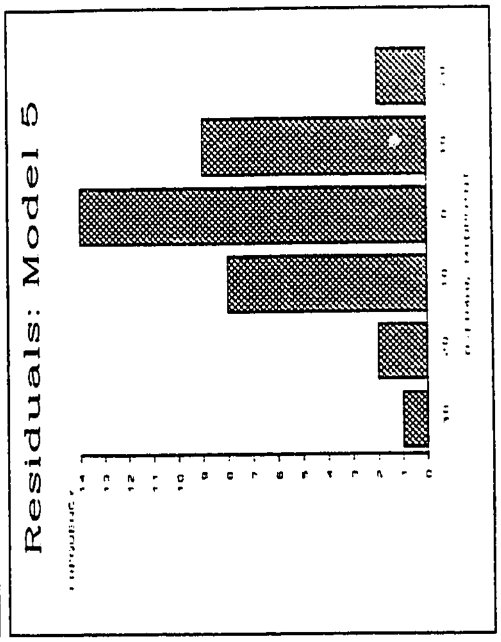

Variable: VH1_GRP

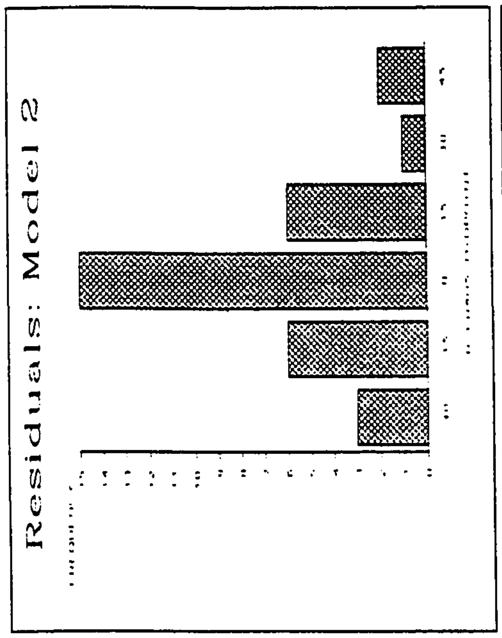
# Appendix I: Histograms of Residuals

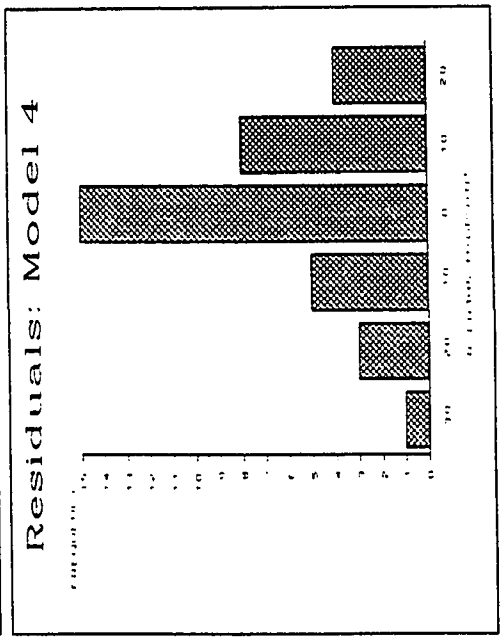
The following are histograms of the residuals of accepted hypotheses. Model numbers refer to numbers of the equations in Chapter 5 and Appendix F.

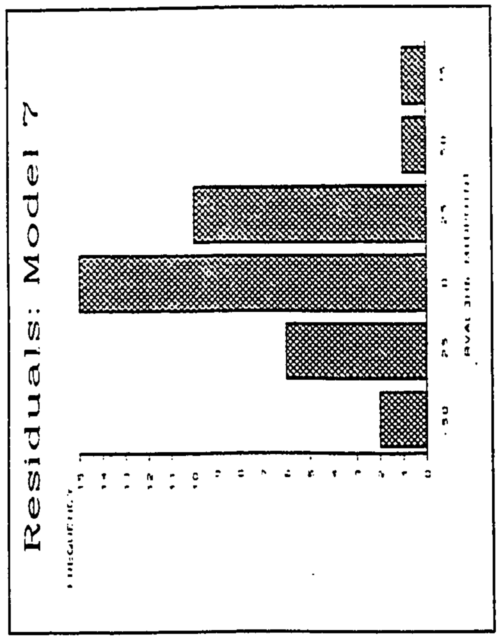


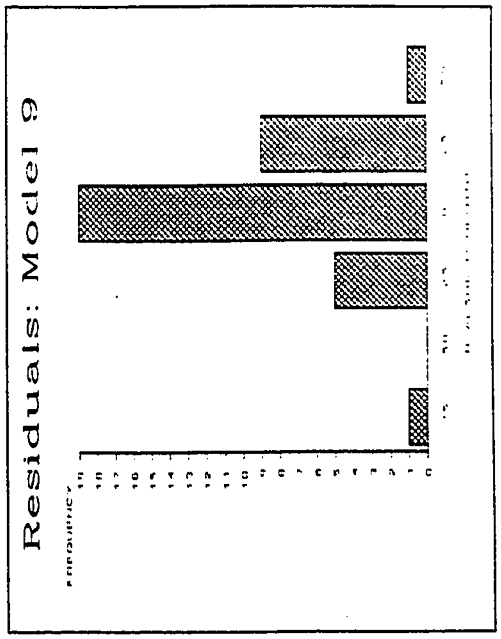

Residuals: Model 3

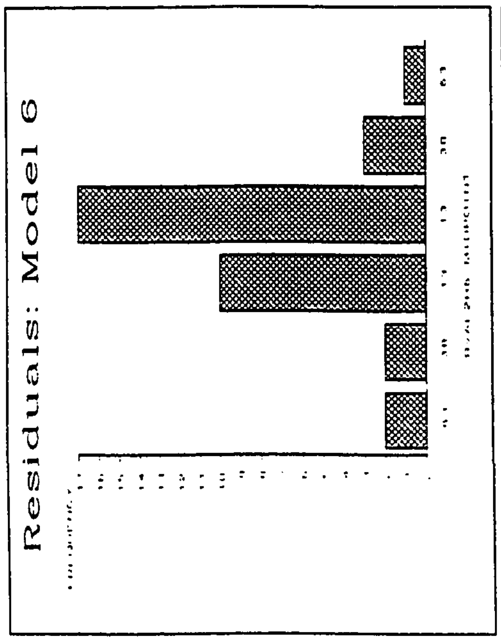

Residuals: Model 5

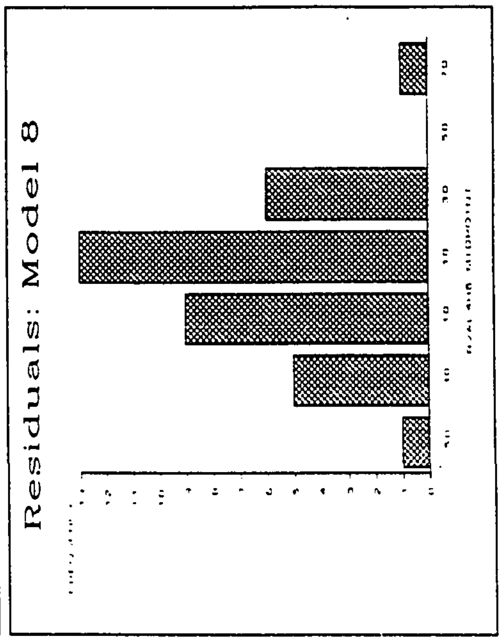

Residuals: Model 2

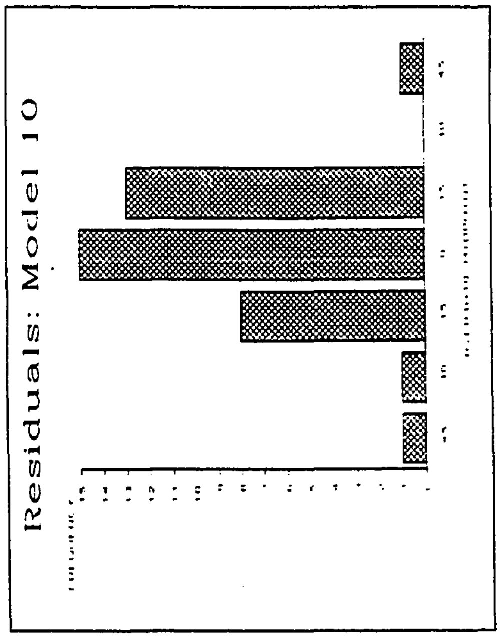

Residuals: Model 4

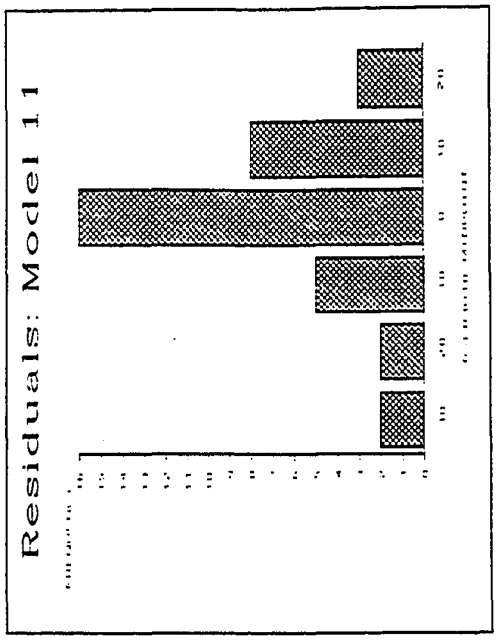

Residuals: Model 7

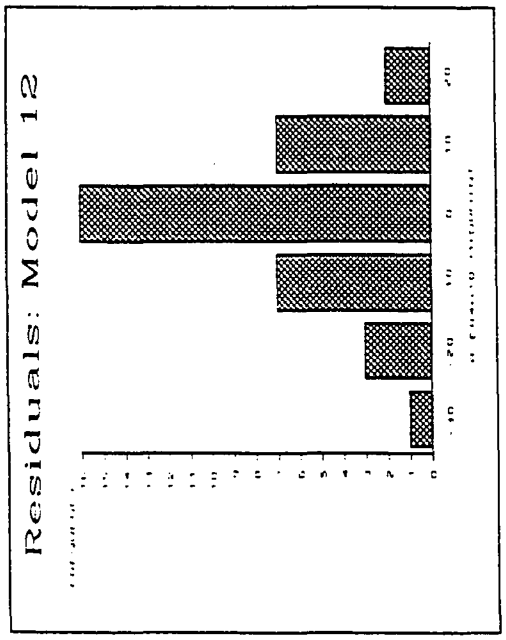

Residuals: Model 9

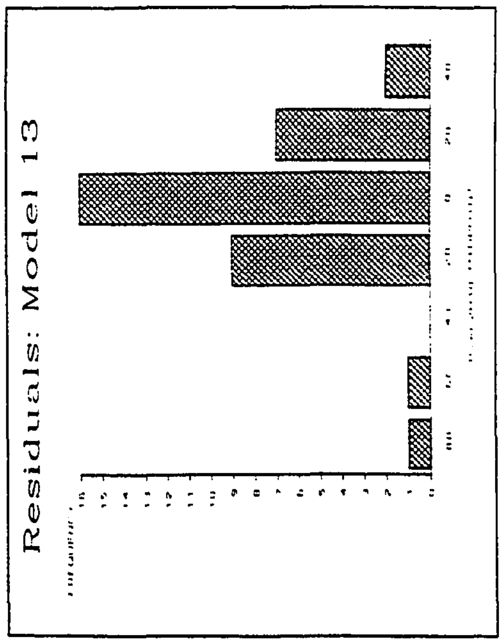

Residuals: Model 6

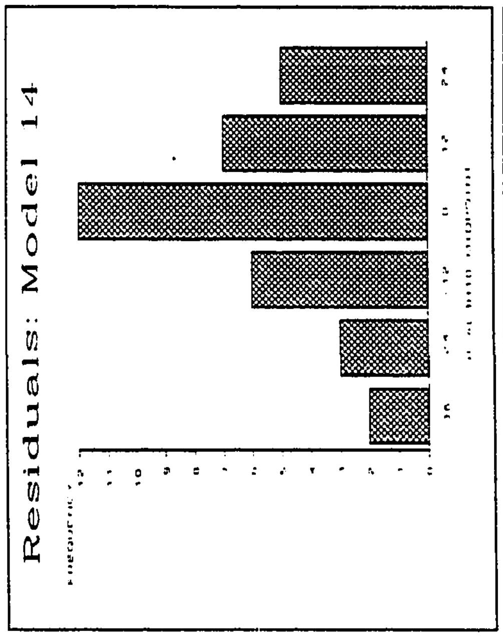

Residuals: Model 8

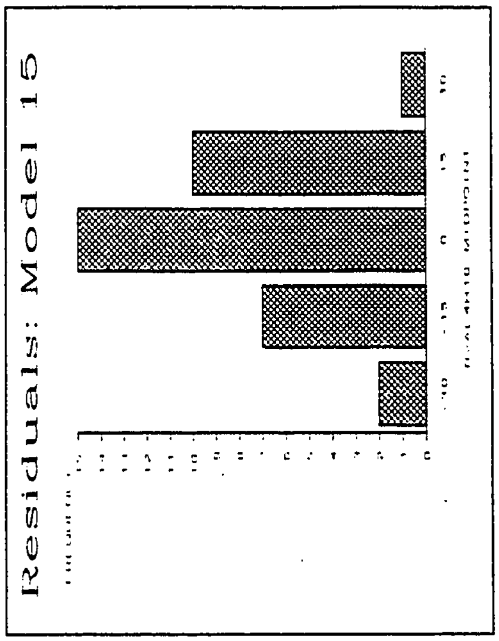

Residuals: Model 10

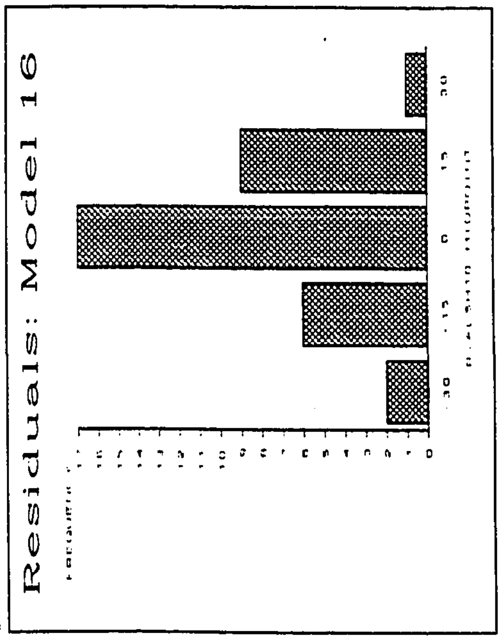

Residuals: Model 11

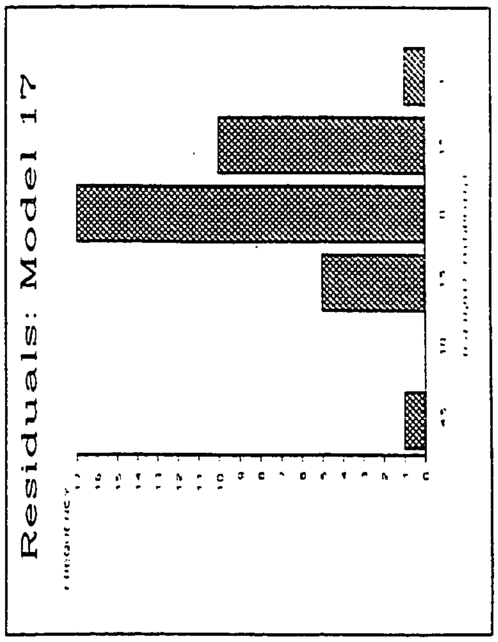

Residuals: Model 12


Residuals: Model 13

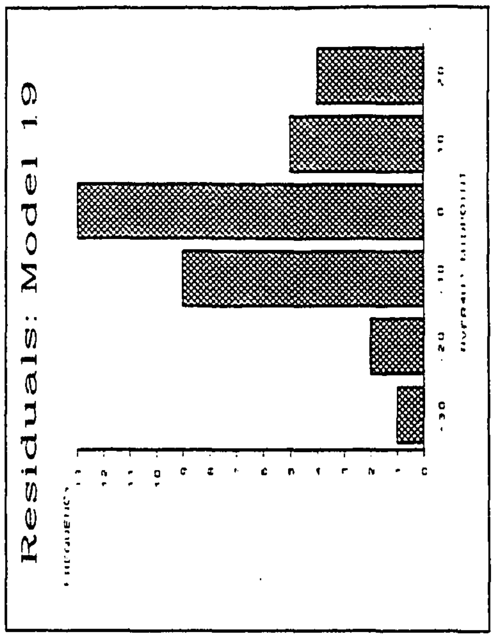

Residuals: Model 14

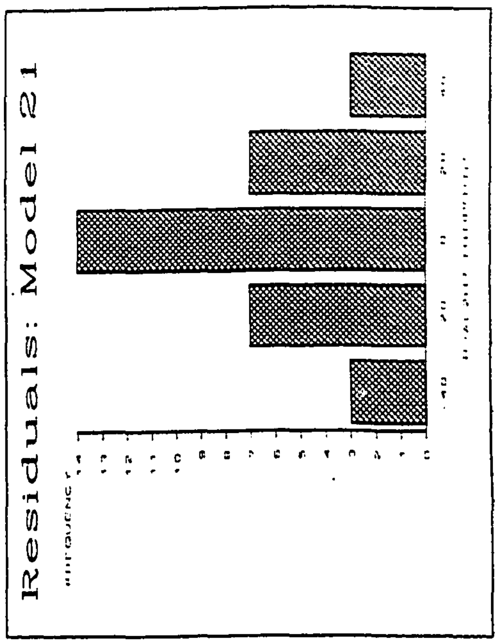

Residuals: Model 15

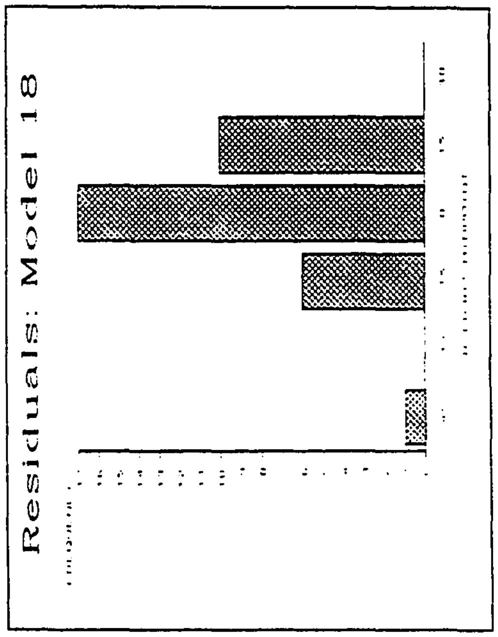

Residuals: Model 16

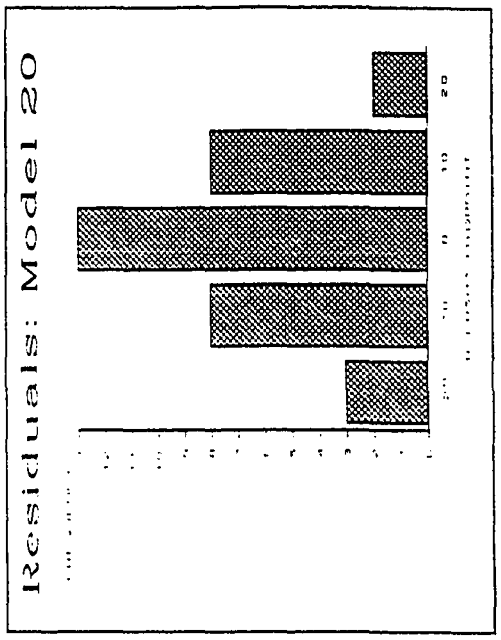

Residuals: Model 17

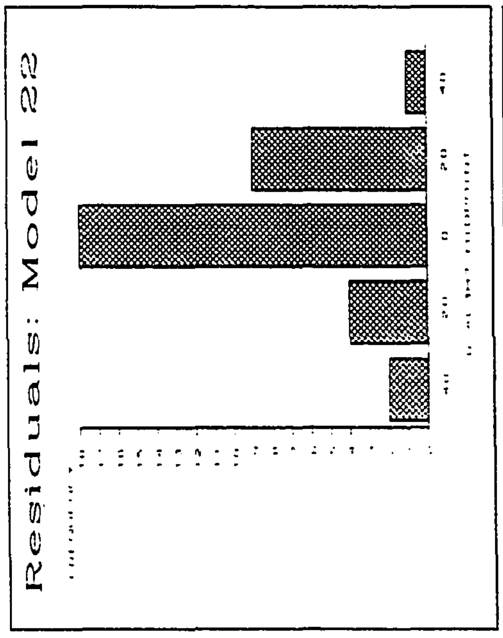
**Appendix I:** *page 6*

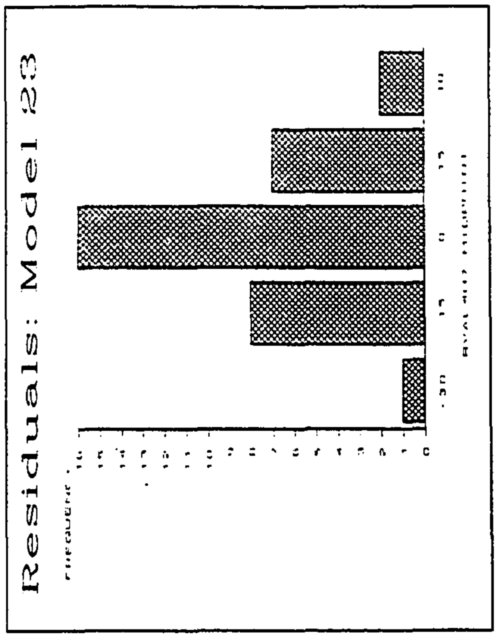


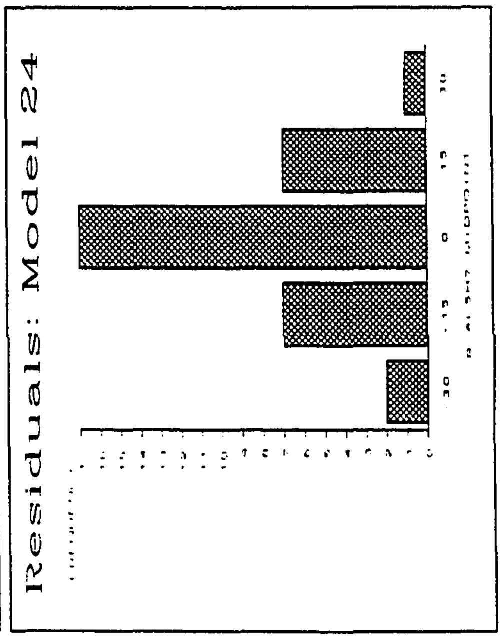


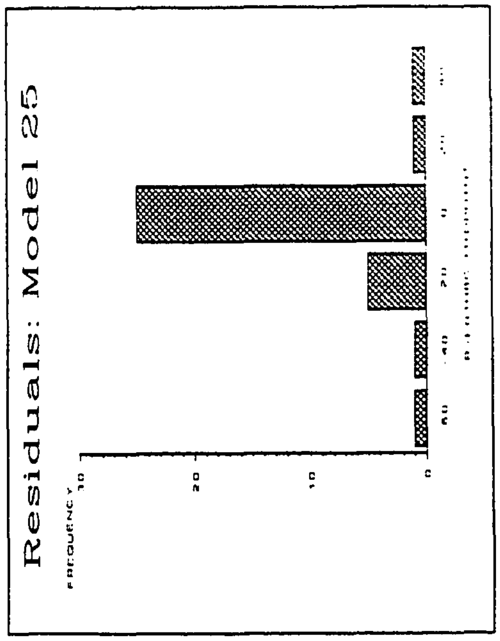


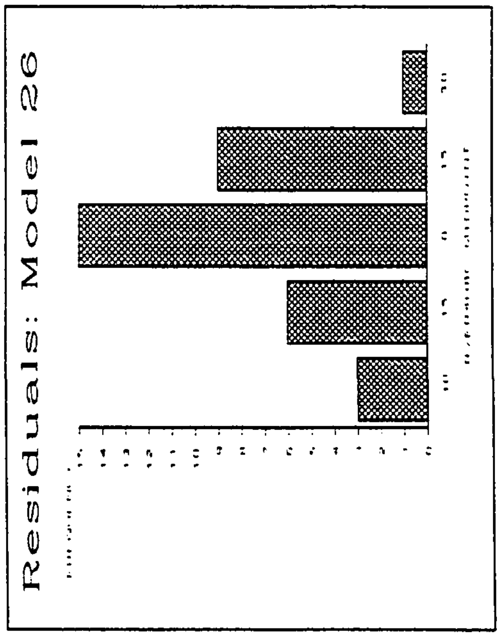


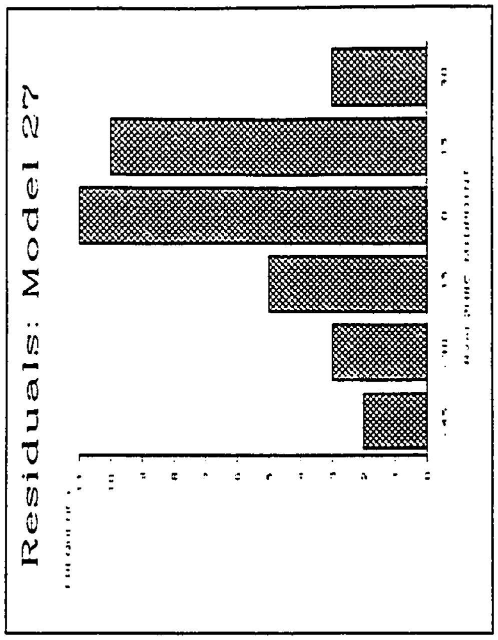

Residuals: Model 22
-40
-20
0
20
40

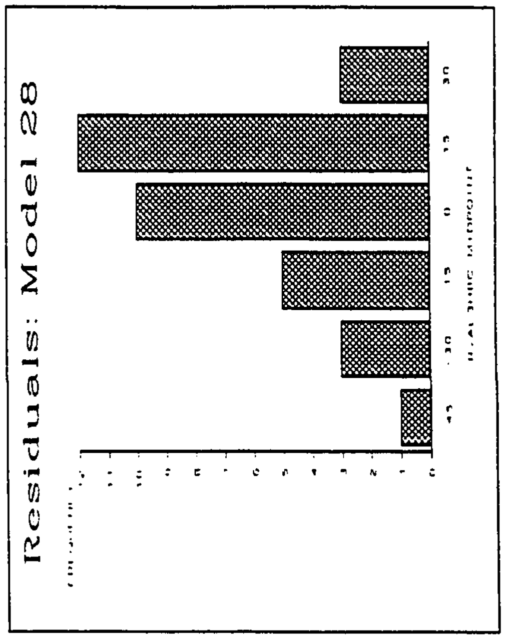

Residuals: Model 23
FREQUENCY
-30
-15
0
15
30

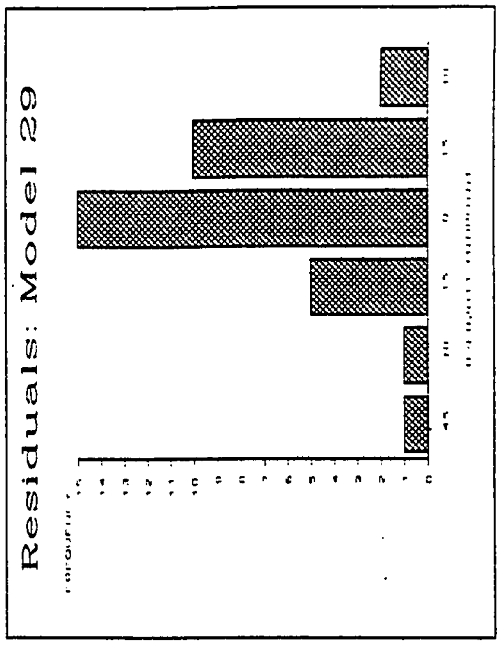

Residuals: Model 24
-30
-15
0
15
30

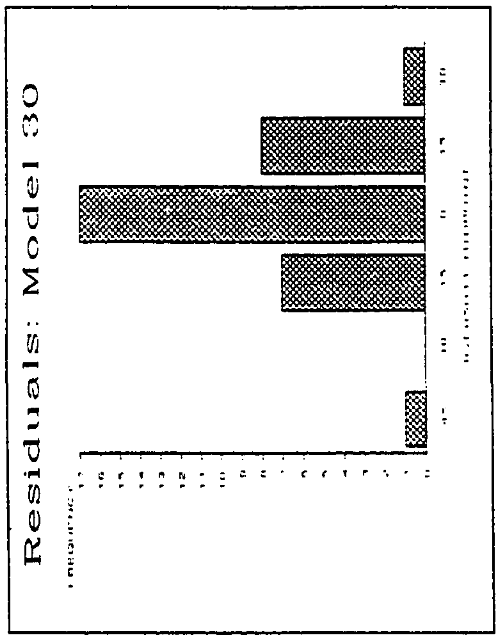

Residuals: Model 25
FREQUENCY
30
20
10
0
-60
-40
-20
0

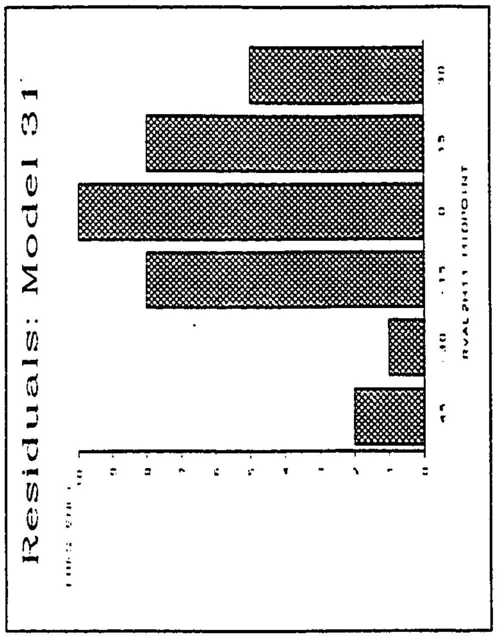

Residuals: Model 26

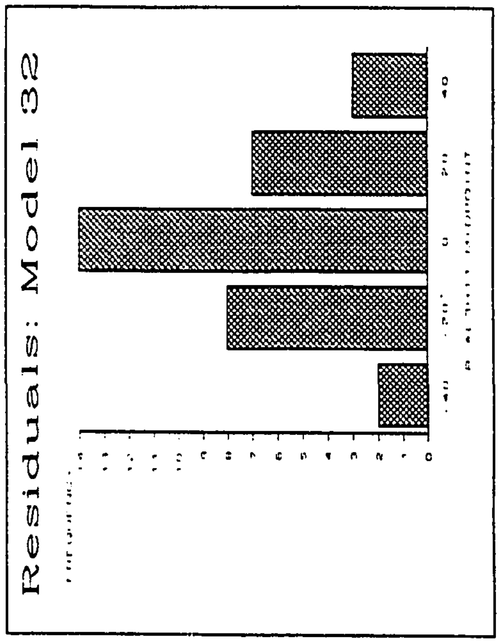

Residuals: Model 27

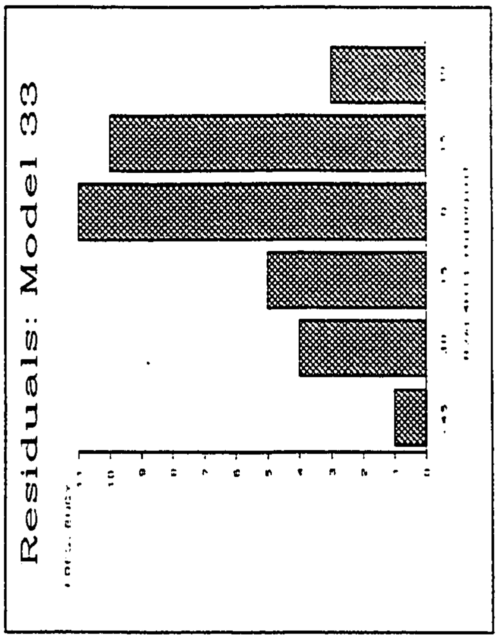

Residuals: Model 28

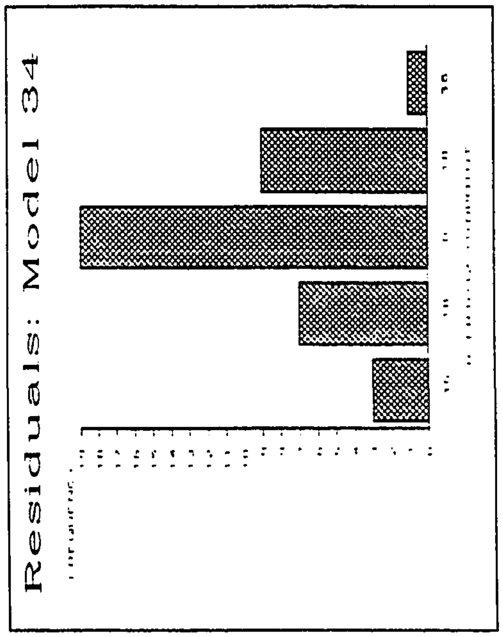

Residuals: Model 29

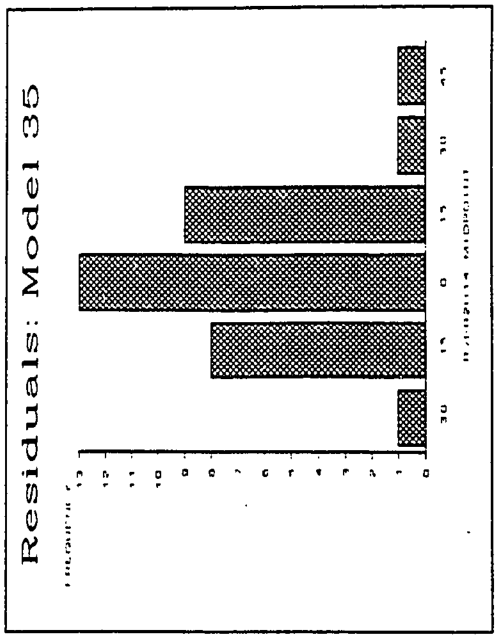

Residuals: Model 30

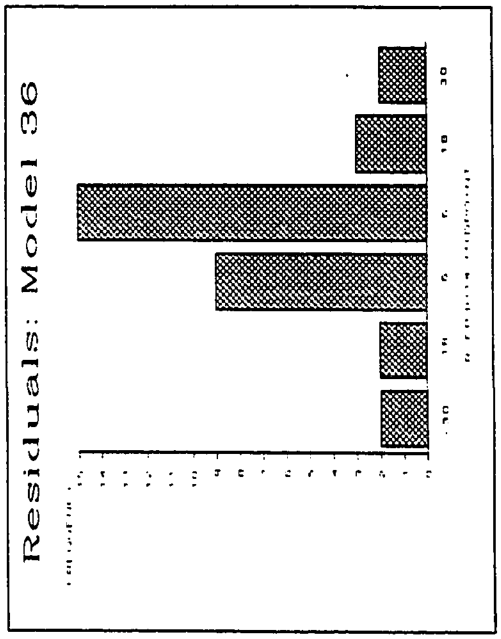

Residuals: Model 31

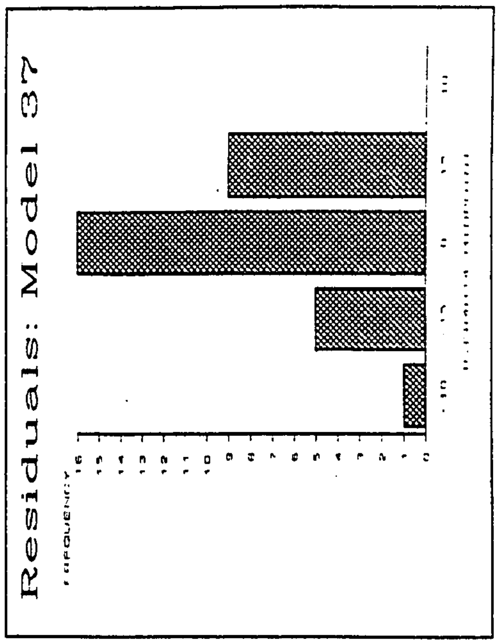

Residuals: Model 32

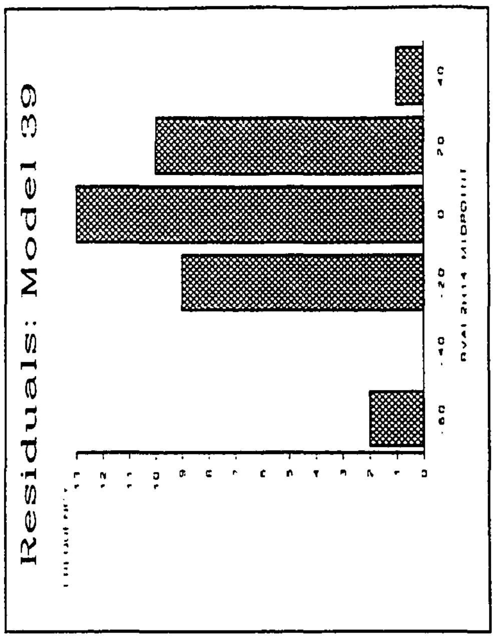

Residuals: Model 33

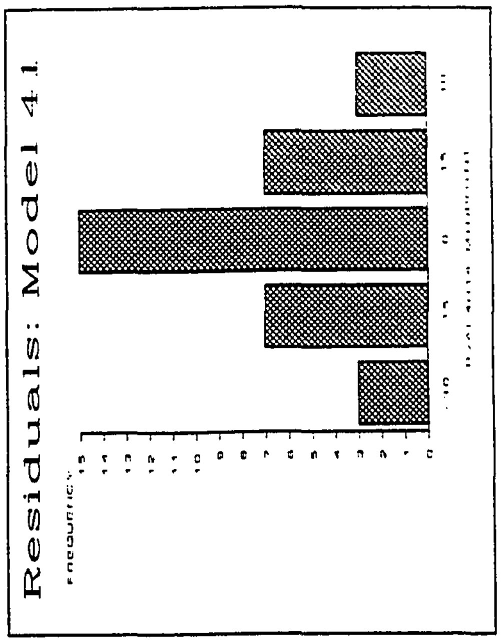

Residuals: Model 34

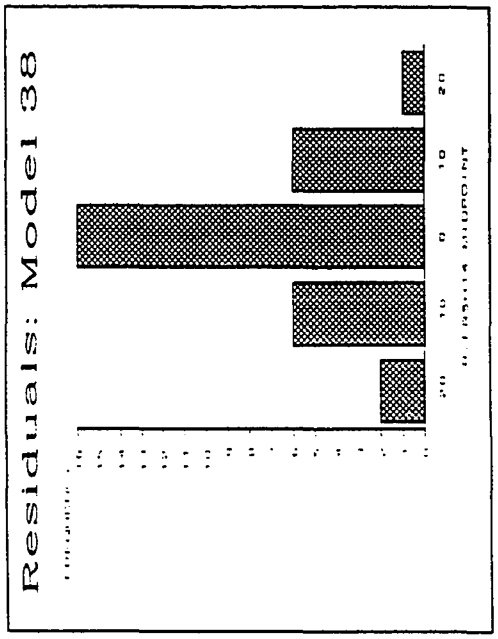

Residuals: Model 35

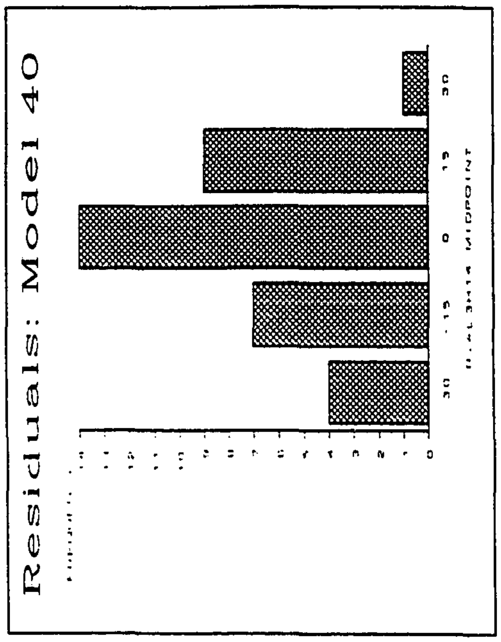

Residuals: Model 36

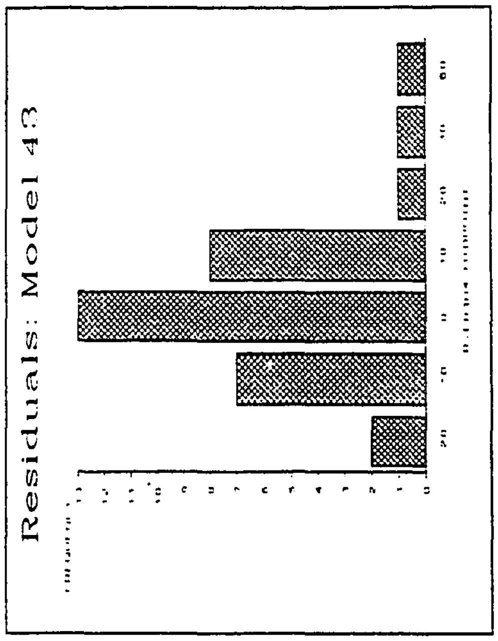

Residuals: Model 37
FREQUENCY

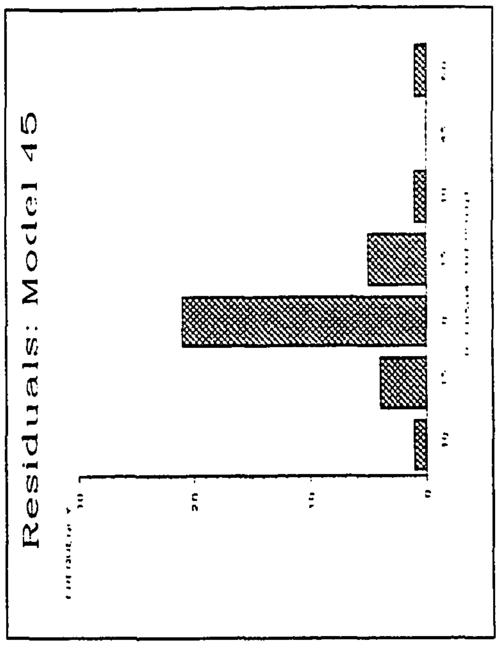

Residuals: Model 39
FREQUENCY
RESIDUAL MIDPOINT

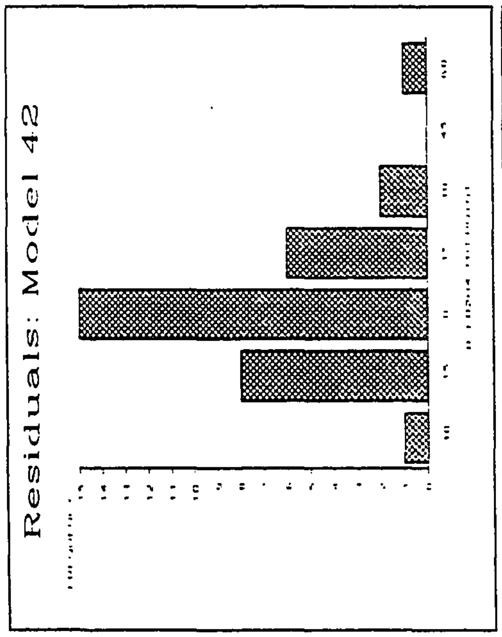

Residuals: Model 41
FREQUENCY
RESIDUAL MIDPOINT

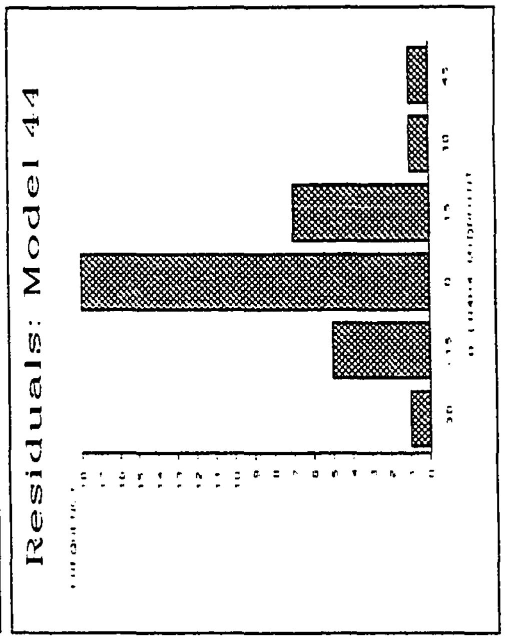

Residuals: Model 38
FREQUENCY
RESIDUAL MIDPOINT

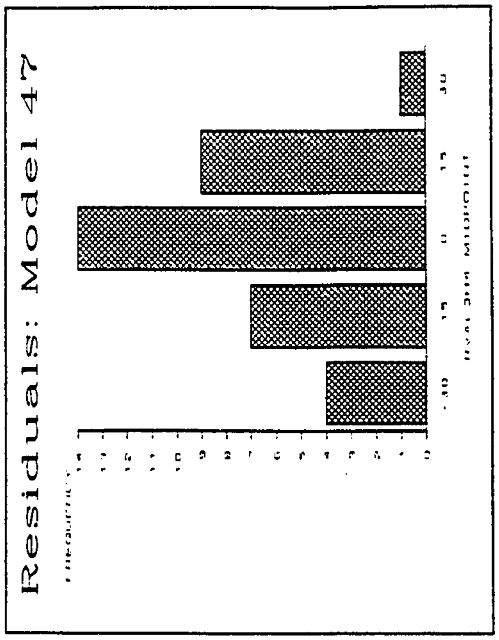

Residuals: Model 40
FREQUENCY
RESIDUAL MIDPOINT

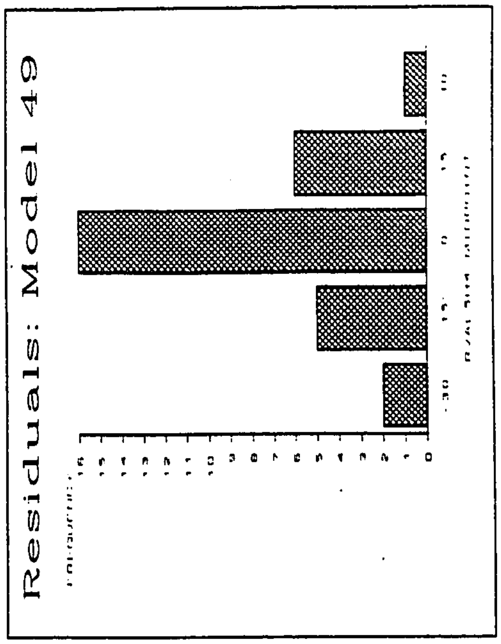

Residuals: Model 43

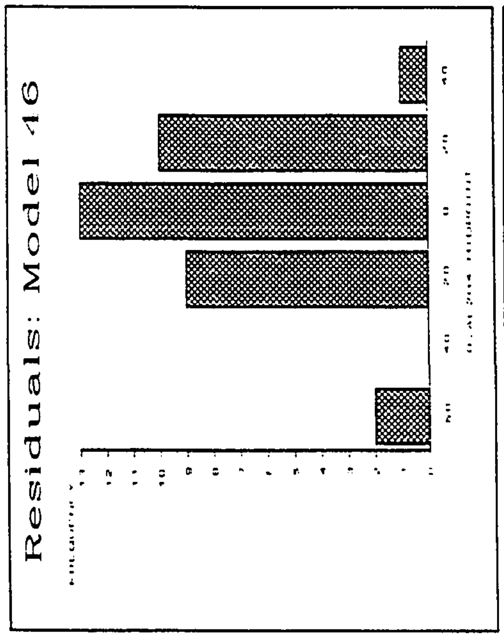

Residuals: Model 45

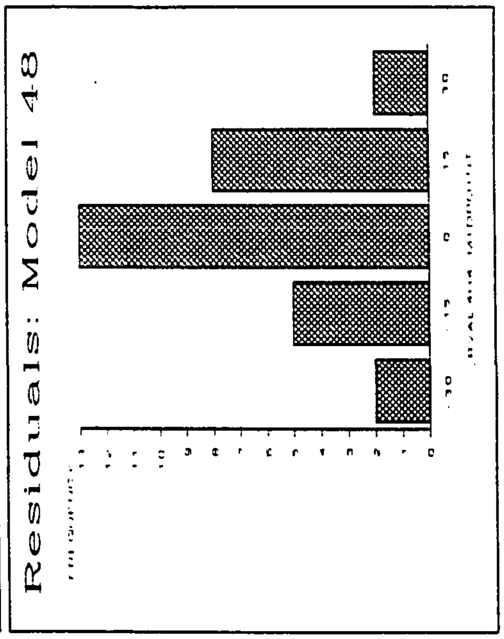

Residuals: Model 42


Residuals: Model 44


Residuals: Model 47


Residuals: Model 49


Residuals: Model 46


Residuals: Model 48