Proceedings of the 1988 Winter Simulation Conference
M. Abrams, P. Haigh, and J. Comfort (eds.)

Panel Session: Validation: Expanding the boundaries

SESSION ORGANIZER: Robert G. Sargent
Simulation Research Group
441 Link Hall
Syracuse University
Syracuse, NY 13244

SESSION MODERATOR: Richard E. Nance
System Research Center
Virginia Tech
Blacksburg, VA 24061

FOCUS PRESENTATION: Robert G. Sargent
Syracuse University

PANELISTS

Jerry Banks
School of Industrial and System Engineering
Georgia Institute of Technology
Atlanta, GA 30332

Jorge Haddock
Department of Decision Sciences & Engineering Systems
Rensselaer Polytechnic Insitute
Troy, NY 12180

Kenneth N. McKay
WATMIMS Research Group
University of Waterloo
Waterloo, Ontario, NTL 3G1
Canada

Nelson Pacheco
Mitre Corporation
Falcon Air Force Station
Colorado 80912

Jeff Rothenberg
The Rand Corporation
1700 Main Street
Santa Monica, CA 90406

INTRODUCTION

This panel session is to address the topic "Validation: Expanding the Boundaries." A focus presentation has been prepared consisting of thirteen questions to be addressed. Each of the panelists was asked to prepare a position statement for the Conference Proceedings addressing a subset of the questions. The range of responses consisted of long answers to a single question or topic, medium size answers to a few questions, short answers to all questions, and a narrative. This panel session should be "lively" as the boundaries of validation are expanded.

FOCUS PRESENTATION:

VALIDATION: EXPANDING THE BOUNDARIES

Prepared by
Richard E. Nance, Virginia Tech and
Robert G. Sargent, Syracuse University

Model validation is universally recognized as a crucial responsibility in the model development effort. Both model and data validation have been cited as keys in the credibility assessment process leading to the acceptance or rejection of model results in decision support. However, current research is expanding the boundaries of the validity determination responsibility, sometimes in subtle ways that appear not to have been recognized by either simulation reseachers or practitioners. The questions below provide the core issues for the panel discussion.

1. To what extent are statistical validation tests applicable for large, complex modeling studies?

2. With large, complex models, how does one assure that the variables constituting the experimental frame [ZEIGLER 1976, p. 30] are sufficient? (Are there any ways to detemine if we have the "best" ones or enough?)

3. Can the verification and/or validation of model components provide a basis for claiming the validity of the entire model when behavioral comparisons of model and real (referent) system are not possible?

4. How can the validation responsibility (including credibility) be discharged properly when a referent system does not exist, such as with an SDI system?

5. If an expert system is coupled with a model in the experimental setting, must the expert system be validated?

6. How are expert systems validated? To what extent is that responsibility recognized and described in the AI literature?

7. Does an expanding knowledge base in an expert system mandate an on-going validation responsibility?

8. Can part or all of the validation responsibility be vested in an expert system?

9. Who bears the responsiblity for the validation of simulation support tools in simulation model development environments -- the seller, the buyer, or the user?

10. Can the claim of model validity be made in the absence of validation of the model development environment?

11. Can an invalid support tool (utility) produce valid models?

12. What should be the recourse following the discovery that a model development utility has a serious error? Who bears what responsibilities?

13. How should the validation responsibility be incorporated within the model life cycle? As changes take place within both the referent system and the model, what responsibility exists for assuring the continuing validity of the model? (Must this model management responsibility be discharged in the same way as the initial validation?)

POSITION STATEMENT
Jerry Banks, Georgia Institute of Technology

Issue: To what extent are statistical validation tests applicable for large, complex, modeling studies?

Appropriate output analysis of a simulation provides a specific degree of confidence on accuracy. However, there are many limitations to the use of statistical methods in large scale simulation models. The two most important limitations are the (1) high cost (in terms of time and computer usage) of performing replications of the simulation and (2) the lack of real world data from which to draw comparisons about the simulation results. Because of these limitations, it is usually impossible to perform a complete statistical analysis on a large-scale simulation.

However, rather than develop new statistical methods I propose that existing methods can be used to increase the understanding of the model and obtain some level of confidence that the model is correctly imitating the system under investigation. Four statistical methods which have potential application are control charts, acceptance sampling, fractional factorial analysis, and cluster analysis.

Because of space limitations, only the first of these methods, control charts, will be discussed in the response. If time is available during the Panel Session, the other methods will be discussed.

The uses of control charts in large scale simulations are many and are limited only by the imagination. There are two advantages to using control charts to analyze simulations. First, replications are not required. If a long simulation is run, where the factors to be studied are calculated frequently (or could be calculated frequently), the control chart can measure the variability and mean of the factors as the simulation progresses and the user can be confident that these factors are being developed correctly. (Problems of serial correlation must be considered!) Second, by monitoring the mean and average of important factors, understanding of the model should increase. Some sample applications are listed below.

(1) Analysis of input - This appears to be an excellent area for application. For example, in a complex military simulation, the user presumably knows the maximum, minimum, and median number of a certain weapon type that a unit of a specific size may have. If this data is being used as input, a control chart can easily highlight any change in the median or in the specified range so that appropriate action can be taken. For example, if a typical Blue force division has $x \mp y$ tanks, and the number of tanks being input lies outside of this bound, the use of a control chart can cause an error or warning message to be printed and/or stop the program.

Attribute charts also could be implemented if a user is willing to have a percentage of the data be incorrect. Attribute charts are normally used to measure the number or proportion of items that are nonconforming. In inputting data to a simulation, the goal is to have all data conform to the standard. So, a variable control chart that stops the program as soon as a nonconforming input is detected, instead of keeping track of the number or percentage of nonconforming input, is more appropriate than an attribute control chart.

An extension to monitoring a one time input of data at the beginning of a simulation is to use control charts when data is input frequently to a program. For example, some low resolution Army models use input from a second model whenever a simulated battle occurs. By monitoring each input an analyst can insure that the input remains within specified bounds over time.

(2) Analysis of model processes - This method appears to be a viable use for control charts. Again, we concentrate on complex military simulations. While a program is running, key factors such as attrition, casualty rate, travel rate, etc., can be monitored by a control chart. An error measure can be printed if these values exceed the control limits. For example, it is known that dismounted infantry cannot move more than x kilometers per day. If a control chart monitors the movement of a dismounted unit during the course of a simulation, and on a specific day the unit moves more than x kilometers, the amount of distance the unit moved can be printed, along with the factors that went into the movement equation. An analyst would then be able to determine if the input to the movement equation was incorrect or if the equation itself is overestimating travel.

The level of detail of the control chart is flexible. For example, a control chart can be implemented when a unit is moving up hill. By having the program measure the average distance traveled, the analyst can insure that the movement equation correctly accounts for hilly terrain. It should be pointed out that the first portion of this example relied upon the analyst knowing the maximum movement of a specific unit; while the second portion was based on the program calculating the average and range of movement in specified terrain.

(3) Analysis of output - All outputs with reasonable (calculable) means and variances can be monitored with discrepancies noted. One advantage in employing control charts for output is provision of the mean and range. This allows a user to understand the model by observing what ranges on key factors are generated by the model and how these factors vary as compared to the outcome of the simulation.

POSITION STATEMENT
Jorge Haddock, Rensselaer Polytechnic Institute

Validation of simulation systems such as *simulation generators* is different than validation of simulation models. The effort in the validation of simulation models is concentrated on validating the distributions used to model the input processes, and validating the input-output transformations being carried out within the model. In the case of simulation systems the validation of input-output transformations gains major significance. In fact, since the simulation generator is not designed for a specific scenario, validation of input data processes does not form any part of the validation exercise for a simulation generator, rather it becomes the responsibility of the simulation analyst using the simulation generator for modeling a specific system. The purpose of the validation process in the case of simulation generators is to ensure that it correctly models the real system as described in the input dataset. Given a particular dataset, it should lead to such performance measures as would have been realized in a real system described by that dataset. (The validity of the input data is assumed.)

The importance of verification and validation of simulation generators cannot be understated. Obviously, if the generator itself is not valid, regardless of the effort spent on validation of input data, the results will be misleading. Moreover, while a simulation model is used to study one specific system, a generator may be used for studying many different systems.

Verification of simulation generators can follow the same techniques as followed for simulation models; for example, modular program development and structured walk through. Models produced by the simulation generator can be verified and thus verify the generator. Trace outputs and graphic animations for such models can be examined to verify the generator.

Validation of simulation generators requires using different approaches than traditionally used for validation of simulation models. Simulation generators are a rather recent development and only a few validation techniques have been reported in the literature. There is a need to address the issue of validation of simulation generators. Some approaches are described here.

Sensitivity Analysis: Sensitivity analysis consists of comparing the effect of change in major input parameters, indicated by simulation results, to the expected trends. The expected effect on the output factors will be realized only when the respective base cases are being significantly affected by the particular input factor.

The sensitivity analysis for validation requires the following:

(1) identifying the major input factors in the system;

(2) estimating the effect of changes in values of major input factors on major output factors;

(3) identifying base case configurations for examining the effect of changes in the input factors;

(4) carrying out the simulation runs and analyzing the outputs;

(5) exploring and explaining the unexpected trends. If required, modify the model and repeat runs.

When all the effects of changes are shown to be as expected, confidence in the validity of the simulation model is increased.

Extreme Situations: This method consists of carrying out runs to simulate extreme situations and to verify that the model performs as intended in such situations. Successful completion of these extreme situation runs will verify the ability of the computer coded model to perform correctly in such situations. This method complements other methods, which concentrate on validating/verifying the ability to model typical situations.

Comparison with Other Models: Simulation analysts agree that the best way to validate a simulation model is to compare the results predicted by the model with performance of the real system, i.e., predictive validation. In the process of validation of a simulation generator, comparison of simulation results with a particular real system (assuming access to one is available) might change the emphasis from validating the input-output transformation in the generated models to input data validation for the system (testing the distributions, etc.). Unless the input data processes are validated, the validity of input-output transformations cannot be established. Also after spending all the effort, the generator will be validated only for that particular dataset. A better approach to validating the input-output transformations in the generator would be to compare the results against results obtained using other models operating with the same datasets. Models which may be used for comparison purposes can be any of the following: (1) analytical models; (2) simulation models developed for specific systems; (3) other simulation generators for similar systems.

Comparison with analytical models results is perhaps the most credible option here. The only drawback is that anlytical solutions are available only for simple scenarios. The other two options, namely comparison with other simulation models and other simulation generators, suffer from a major drawback - the validity of the other model/generator needs to be established first. In addition, the option of comparing with simulation models for specific system can prove to be very time consuming if such models do not already exist and thus the exercise may not be worth the effort. For these reasons, the latter two options are not recommended.

Comparison with analytical model requires dealing with another issue - point estimates given by analytical models are to be compared with confidence intervals generated by analysis of replications of simulation runs. If the point estimate falls outside the confidence interval, the simulation model needs to be reexamined. However, if the point estimate falls inside the interval, does it indicate that model is valid regardless of exact placement of the point within the interval?

Acknowledgement

I would like to thank Sanjay Jain, a Ph.D. Candidate, for his research and assistance, as well as Professor Bob O'Keefe for his comments, both of Rensselaer.

POSITION STATEMENT

Kenneth N. McKay, University of Waterloo

The following responses refer to the numbered questions addressed to the panel. The responses were generated by the following individuals associated with the WATMIMS research group: John A. Buzacott, Elizabeth M. Jewkes, Kenneth N. McKay, and John Trebish. The responses are in the context of our experience with medium to large scale manufacturing models.

1. Statistical validation tests are applicable for sub-models of the complete system, or very special cases of the general problem for which there are analytical results. Medium to large models are usually too complex and have too many variables for a detailed statistical validation.

2. If a referent system exists, and there are recognized experts, it may be possible to identify the probable 'best' variables and their range of acceptable values. It is then possible using experimenal design methods to determine the contribution of the variables. The referent system can be studied as to the significance of the variables and their interactions. However, there are three major difficulties: finding the right expert, setting the right level of detail to include (the more detailed the question, the more variables are required), and establishing the deterministic criteria with which to validate the inclusion or exclusion of variables. While it may be possible to claim sufficiency for complex models, it is probably infeasible to perform the necessary work to substantiate that claim. If a referent system does not exist, it is not clear how a confident claim of sufficiency could ever be made.

3. If there is little or no feedback from one stage of the model to another, it is possible to sequentially validate and thread the sub-components together. However, if there is feedback which results in changes in the submodel execution, the serial approach is inadequate and a macro approach is warranted.

4. Yes, if the problem has sufficiently few enough interactions and interdependencies that a panel of experts can 'understand' what is happening. Depending on interactions, it might be possible to work with a higher level than molecular - this depends on problem context and if macro modelling is possible. There are other issues of course: there might not be any true experts yet in a new field (they may be competent but not experts), or there may be erroneous assumptions made about interactions between model components that are never challenged nor brought to light.

5. Yes, against its functional requirements. The extent of validation depends on what the expert system is doing: database control, deterministic data selection, stochastic manipulation, operational heuristic generation, or output analysis. It is possible for the front-end (expert or otherwise) to bias and confound the simulation - therefore, it must be analyzed with the same care that would be afforded other software. The less it does, the closer validation gets to the normal software verification procedure.

6. The topic of expert system validation and verification appears to be poorly addressed in the AI literature. Validation and verification seem to be blended together because of the computational complexity inherent in expert systems (i.e., large decision trees). In the AI literature we have reviewed, mathematical and/or statistical validation is not mentioned. Two forms of validation are often mentioned: face validation to the original expert, and a peer review by other experts of the computer and original expert's solutions to problems. In our opinion, validation of expert systems is not any different from validation of simulations: careful analysis of cause and effort relations through the controlled manipulation of variables and their values. The face validation is also important but can be very myopic and misleading - it is necessary but not sufficient.

7. Yes, if the expanding knowledge is interrelated to other elements and can result in a different scenario being played out. Ongoing validation is required to ensure that biases and confounding factors are not introduced.

It is possible to add information to a knowledge base that will not contaminate the system, but this has to be decided on a case by case basis.

8. Verification, yes; validation, no! It is possible to construct automated test suites that can expertly analyze and subsequently exercise software to ensure that the software executes correctly (boundary cases, loop execution, etc.). However, the validation process depends a great deal on 'know-how' and often what is missing is more important than what is present. Expert systems are not capable of understanding what the implications are of observed and unobserved events unless every implication is explicitly coded; hence validation by expert systems is not feasible.

9. This depends on who has claimed what. For example, if the vendor of the software tool being used claims accuracy in a specific domain or application, the vendor is primarily responsible for the major validation effort. The consumer has the responsibility, during the tool acceptance phase, of ensuring that the tool is suitable for their domain. If the vendor is supplying general purpose software, the vendor assures the accuracy of individual statements and the major validation responsibility resides with the user (for their specific domain and application).

10. No, some form of validation must be performed on the model development environment. It is necessary to know where the weak areas are and where care must be taken in the model. For example, if a random number generator is known to generate inadequate sequences, an alternate routine may possibly be added to circumvent the problem. If validation is not performed (by someone), the model's execution can never be said to be valid (i.e., is something a problem, or is it a symptom?). The tools must be known to be deterministic and capable of repeatable performance.

11. Possibly, it depends on the model's nature and the characteristics of the tool. If the model builder is aware of weak areas in the tool, it is possible to avoid the problem spots, or perform extra validation in those areas. A high degree of user sophistication is required to make a claim of model validity when using a tool that is questionable. An analogy may explain why it is possible to have a valid model when the tool isn't perfect: imagine an aircraft that is only capable of making cold coffee - is this an indication that the craft will not fly - hopefully not.

12. This depends on the claim of the vendor - if the tool has a functional error or theoretical weakness - then it is the vendor's responsibility. If the user is using the tool for something other than what it was intended for, the user has the problem and should pay the vendor to enhance the utility. The user has the responsibility to provide a reproducible problem with all of the necessary data/files to the vendor and also has a responsibility to validate the correction.

13. Developers and users of the model must jointly agree upon a strategy for ongoing model validity. Models are rarely one-shot exercises, yet are usually budgeted as if they were; nor are they usually revalidated when used for a different purpose. Developed internally, simulation models are not given the same scrutiny and planning that other software would be given within the organization. That is, a company may have a software development life cycle established for software they create for end-users, but the life cycle will not be followed for internal support software (i.e., simulations). The simulation life cycle must have clearly identified phases where major updates are performed and the system completely re-validated. Minor changes and bug fixes, may not require a complete re-test. The model developer is responsible for monitoring the referent system and detecting the significant changes. The people directly involved in the referent system are probably too close to the slowly evolving system to recognize what should be changed in the model.

POSITION STATEMENT

Nelson Pacheco, Mitre Corporation

1. To what extent are statistical validation tests applicable for large, complex modeling studies?

We should first differentiate between modeling for the purpose of understanding an existing system and modeling for the purpose of making inferences about a system which does not yet exist. In the case of existing systems, classical statistical validation tests which compare simulation versus actual system data are well known and proven (at least for terminating simulations). Large, complex modeling studies, however, are often used in the concept evaluation phase of proposed systems which do not yet exist. In this setting, the concept of a simulation as an experiment is more applicable. In physical experiments, there is no a priori truth known; only predictions from theories and assumptions. The experimenter tests a view of reality which can be tested further by other experiments. The validity of these experiments depends on their reproducibility; any other experimenter should be able to observe the same results, given a similar set of experimental conditions.

In an analogous fashion, the modeling of large complex systems which do not yet exist may be viewed as an experiment. As in physical experiments, the validity of these modeling studies can be based upon their reproducibility. Reproducibility in this sense implies that other modelers working independently, perhaps using other paradigms, frameworks, tools, etc., but with the same conceptual system specification and initial conditions, should be able to reproduce the same output measures from their simulations. Failure to do so, of course, does not necessarily imply incorrectness, but may instead lead to a challenge on the correctness of the other simulations.

In the case of deterministic simulations, reproducibility can be measured by establishing a priori tolerance intervals, with simulation outputs within the tolerance interval assumed to be equivalent. Given several independently-produced terminating Monte Carlo simulations, statistical validation tests such as Analysis of Variance (ANOVA) for testing equality of means, or multiple comparison procedures (e.g., Least Significant Difference, Duncan's Multiple Range Test, etc.) for comparing confidence intervals are appropriate. For steady-state Monte Carlo simulations, the output sequences may be viewed and tested as being independent sample paths from the same stochastic process.

3. Can the verification and/or validation of model components provide a basis for claiming the validity of the entire model when behavioral comparisons of model and real (referent) system are not possible?

Although the validation of model components is necessary for validation of a system model, it is not sufficient. The very concept of a system model implies the existence of interactions between components which may be highly complex and non-linear. This is particularly true for military systems wherein command and control elements typically introduce human interactions with the various components which are highly scenario-dependent. For example, a valid model of a sensor and a valid model of a weapon may not combine into a valid model for the employment of the sensor-weapon combination within a command and control context. Not only must the sensor and weapon "work" individually, but they must be effectively employed within a set of threat environment and enemy countermeasures. Without observing the referent system, it is not possible to unequivocally state that the system will behave as the model predicts.

On the other hand, validation of model components can lead to identification of those critical functional interfaces between components whose behavior strongly influences system performance. At this point, explicit assumptions can be stated and agreed upon regarding interactions between these components. Although this does not validate the system model as such, it may enhance the credibility of the simulation study to a sufficient level to allow informed decisions to be made regarding system development. Analysis of the sensitivity of the system to the assumptions can also lead to a better understanding of the critical assumptions.

4. How can the validation responsibility (including credibility) be discharged properly when a referent does not exist, such as with an SDI system?

For such systems, the term "validation" as classically used does not apply. One can never state that the overall system (which does not exist) will behave exactly as predicted by the model. Credibility, however, is a human judgement which is based upon many other factors: previous modeling experience, support tools available, the model development environment, assumptions made, etc. Whereas validation requires a positive determination of fit between the model and reality, credibility emphasizes a guarding against as many potential errors as possible. In other words, to be credible a model should be able to answer its critics. A criticism often heard in the SDI context is that models can not be validated because of lack of real world data. Viewed in an experimental setting, however (see response to question 1), the role of the model is precisely to generate data upon which to base inferences regarding predicted system performance. These inferences should be based on an explicit and well-examined set of assumptions, and may in turn be used to focus research on those issues which are critical to the overall system effectiveness. As limited tests and demonstrations are conducted, and real world data becomes available, changes in the model and/or the system design may become necessary, increasing the level of credibility of simulation studies.

13. How should the validation responsibility be incorporated within the model life cycle? As changes take place within both the referent system and the model, what responsibility exists for assuring the continuing validity of the model?

Responsibility for initial validation testing rests with the developer; this should include the development of test cases which explore the region of validity. However, this does not absolve the user from the responsibility for independent verification and validation (IV&V). The developer's initial validation testing should be checked and expanded, as required, by an independent agency knowledgeable of the referent system but not dependent on the system in order to maintain unbiasedness.

A model should only be validated against a specific referent system, which should be explicitly defined. As the referent system evolves, the model may become 'dated' and may no longer accurately represent the system. Although it may not be necessary to update the model with every change in the system, the model's validation must be revisited each time that it is used in a study, to determine whether it is still valid, or if it requires modification and revalidation before it can be used.

POSITION STATEMENT
Jeff Rothenberg, The Rand Corporation

It is necessary to distinguish between a model's *validity* and its *validation.* The validity of a model is an objective (though often unknown) fact, its validation is a proof or demonstration of its validity. Our concern here is to show how models can be validated, not to discuss which models are valid.

As suggested in Rothenberg, 1986 (WSC '86) any model can be characterized by three criteria: (1) it must be a model of some real-world referent, (2) it must have some purpose with respect to its referent (such as prediction, comprehension, communication, etc.), and (3) it must be more cost-effective to use the model for this purpose than to use the referent itself (where cost-effectiveness encompasses costs like time, safety, convenience, etc.).

If "validation" is taken to mean the assurance that a model correctly represents its referent (reality), then this is equivalent to the above definition of modeling, which says that a good model must cost-effectively represent its referent for the intended purpose. Validation considered outside of this context is a vacuous abstraction. Validation must mean the proof that a model cost-effectively fulfills its purpose with respect to its referent (according to its given cost-effectiveness criterion).

Validation always refers to a reality outside the model itself, but there are several distinct cases. The first (and easiest) cases involves a known, accessible referent. Here validation can be performed by direct experimentation: whatever the purpose of the model, it can be tried for this purpose and compared to the direct use of the referent for the same purpose, to see which is more cost-effective. For example, suppose the purpose of a computerized model of a bouncing ball is prediction, and that the cost-effectiveness criterion is to minimize the time required for prediction. Then validation consists of using the model to predict the behavior of the ball, while assessing the accuracy of this prediction (fulfillment of the model's purpose) and timing it to see if it is faster (more cost-effective) to run the model than to bounce the ball.

In many cases, however, the referent of a model is not directly accessible for such experiments. Stars cannot be exploded at will, airplanes cannot (safely) be flown into the ground, security systems cannot always be tested, etc. In such cases, validation must be done indirectly by extrapolating from the known, observable behavior of the referent or conducting indirect observations or experiments on it by whatever means are available. Similarly, it is often reasonable to use modeling as a way of gaining understanding about a referent, for example, trying various mathematical constructs as *possible* models and attempting to validate each one to find out which one best approximates the referent (as opposed to deriving a model from a pre-existing understanding of the referent).

In some cases, the "referent" of a model is unreal, a fantasy reality with no concrete form at all. For example, a video game might be thought of as modeling an imaginary reality. This is really a misuse of the term "modeling", since there is no referent. Given something that *poses* as a model (i.e., a "pseudo" model), it is possible to "back project" a reality that *might* be modeled by this pseudo model, but this is an empty, non-deterministic, and misleading exercise. A pseudo model is something that poses as a model but for which there is no referent; it is therefore meaningless to speak of validating a pseudo model.

Closely related to this, however, is the case in which a model appears to be built *before* the reality to which it refers; for example, a design or prototype is often thought of as a model for a proposed referent. To the extent that the referent is unspecified initially, it does not yet exist; therefore, the "model" is a pseudo model, which cannot be validated. The design process can then be thought of as one of "back projecting" the pseudo model onto one of its possible referents. A more useful view may be to consider a design (or specification of any sort) to be a referent on which an eventual system is itself modeled. Once a prototype has been built (whether or not it is modeled on some prior specification or design) it then becomes a referent which can in turn be modeled by a refined implementation.

Extending this argument, a (computerized) implementation of a specification can be thought of as a model of that specification. If M is a conceptual model of some real-world phenomenon, then a particular computer program P that implements M can be thought of as model of M. The purpose and cost-effectiveness criterion here involve the ability to run the implementation on a computer, as opposed to interpreting the specification "manually". This view unifies validation with verification.

Verification is usually distinguished from validation as meaning the assurance that an implementation of a model is correct, i.e., that a model meets its specifications (which may, however, be invalid). Verification concerns the internal consistency of a model, whereas validation concerns the model's relation to its referent. To paraphrase Barry Boehm, validation means building the right system, whereas verification means building the system right. However, verification can be viewed as validation with respect to a specification rather than an external phenomenon or artifact. This view is recursive down to the level of machine instructions, where, for example, an *add* instruction using limited precision arithmetic is an imperfect model of true addition.

Validation of a conceptual model may or may not be harder than validation of an implementation of that model (i.e., verification of a program), but both are equally vital if one is to use the implementation as a valid model. This implies the need to verify modeling environments, expert systems used in conjunction with models, etc.

Note that the expert system paradigm assumes that a system is developed in close coordination with an expert (or experts) who can validate the behavior of the system directly. To the extent that this knowledge acquisition process can elicit "meta-knowledge" that provides consistency criteria for the knowledge base, additions and modifications to the knowledge base might be verified against this meta knowledge. Similarly, it may be possible to develop "validation suites" of tests (i.e., verification criteria) that can be run against a model whenever it is changed.

Validation and credibility are logically independent of each other. If a model can be validated (i.e., *proven* valid), it should be credible, provided the proof itself is credible. On the other hand, credibility by itself cannot confer validity; at best (or perhaps worst) it confers "face validity", meaning that the model *appears* to be valid, whether or not it really is. Though it is possible for a model to be credible (or even validated) purely on the basis of behavioral comparison between the model and its referent, this is rarely convincing for complex models, since their range of behavior is difficult to explore exhaustively. A convincing demonstration of validity (or even credibility) in such cases requires that the model be comprehensible. This argues for modern software engineering approaches to maximize comprehensibility, including the explicit, declarative representation of "knowledge" in models, as pioneered by expert systems and related knowledge based techniques. (For the purpose of this discussion, "knowledge" can be defined operationally as the explicit representation of information in a form that is at once *meaningful* to a human reader and *interpretable* by a computer program.)

The lack of comprehensibility in most large models presents an insurmountable obstacle to their validation. Knowledge-based, object-oriented and modular techniques, as well as the liberal use of metadata to provide motivation, rationale, and traceability of data, assumptions, algorithms, etc. can go a long way toward improving comprehensibility, thereby facilitating validation.