Validating Computational Models

Kathleen M. Carley

April 28, 2017
CMU-ISR-17-105

Institute for Software Research
School of Computer Science
Carnegie Mellon University
Pittsburgh, PA 15213

Center for the Computational Analysis of Social and Organizational Systems
CASOS technical report.

This report/document supersedes
"Validating Computational Models", September 1996

This work was supported in part by the Office of Naval Research (ONR), United States Navy Grant No. N00014-93-1-0793 (UConn FRS 521676). The views and conclusions contained in this document are those of the authors and should not be interpreted as representing the official policies, either expressed or implied, of the Office of Naval Research or the U.S. government. Additional support was provided by the center for Computational Analysis of Social and Organizational Systems and the Institute for Software Research at Carnegie Mellon University.

Keywords: computational models, social and organizational theory, model validation

Abstract

The use of computational models in the social sciences has grown quickly in the past decade. For many these models represent a bewildering and possibly intimidating approach to examining data and developing social and organizational theory. Few researchers have had courses or personal experience in the development and building of computational models and even fewer have an understanding of how to validate such models. And while many papers extort the relative advantages and disadvantages of the computational approach, and many call for the validation of such models, few provide insight into how to validate such models and the issues involved in validation. This paper represents an attempt at redressing this oversight. An overview is provided of computational modeling in the social sciences, types of validation, and some of the issues in doing model validation.

Table of Contents

Validating Computational Models

There is a growing trend in the social and organizational sciences to employ computational models[1] in developing and testing theories. Such models are uniquely valuable for addressing issues of learning, adaptation, and evolution. The value of these models for theory building, however, will require an increased understanding of the potential of these models, and when and how they should be validated. For researchers not trained in modeling or computational techniques such models may appear bewildering; i.e., it may be difficult to understand when to believe a model, and when not to, and how to interpret and use a model’s results. Consequently, there are often well intentioned, but somewhat misplaced, calls for model validation without understanding what validation entails. On the other hand, as noted by Richard Cyert (1994, p. viii), ‘‘[S]ocial scientists, particularly economists, have a fatal attraction for working on theoretical propositions with mathematical models or simulation models and avoiding the real world.’’ In other words, there are models that absolutely require validation that are never validated. What then, is the happy median?

This paper is a first step in locating this median. The goal of this paper then is not to teach programming or even how to build models; rather, the goal is to provide information for the modeler and for the general reader about the process of model validation and techniques for performing such validation. Part of the argument, as will be seen, is that not all models need to be validated and that the level of validation chosen depends on the model’s purpose. In a sense, this paper can be thought of as a consumer’s guide to model validation. One sense that the reader may emerge with is that validation is a complex issue. Another sense, may be, that more research needs to be done on how to validate computational models.

What is Validation?

Often formal models, including computational models, are evaluated in terms of their clarity, parsimony, generality, and testability (Mayhew, 1984). Herein, the focus is on only the last of these issues: testability or as it will be referred to in this paper - validity. No claims are being made about the relative value of clarity, parsimony, and generality. Rather, the point herein is simply that regardless of where a model falls on these other dimensions issues of validation will arise. Moreover, as will be noted, the type and degree of validation needed will in some sense be dependent on the level of parsimony and generality claimed for the model.

General discussions of validity for computational models point to one or more of the following six types of validation: conceptual, internal, external, cross-model, data, and security (Knepell and Arangno, 1993). Each type of validity is assessed in terms of whether or not there is an acceptable degree, where acceptable is defined based on the needs of the researcher or user. Conceptual or theoretical validity refers to the adequacy of the underlying conceptual or theoretical model in characterizing the real world. Internal validity refers to whether the computer code is correct. A model is internally valid if the underlying program is free of coding errors. External or operational validity is concerned with the linkage between the simulated and

[1] The term computational model is used rather than simulation model as it is the more general term and encompasses both enumeration based models, Monte-Carlo models, and symbolic models (such as expert systems). Whereas, the term simulation model has frequently been employed either just for Monte Carlo models or to include human-in-the-loop or gaming models. In this paper, the focus is on all formal models that can be operationalized as a set of computer code.

the real. External validity refers to the adequacy and accuracy of the computational model in matching real world data. Another type of validation is cross-model validation or docking (Axtell, Axelrod, Epstein and Cohen, 1996) where the goal is judge the degree to which two models match. Data validity refers to the accuracy of the data (real and computer generated) and the data's adequacy for addressing the issue of concern. And finally, with respect to security, the issue is one of providing adequate safe-guards or assurances that tampering with the model will be minimized or procedures for determining if the model has been tampered with or fundamentally altered through subsequent reconfigurations. In this paper, the focus is largely on conceptual or theoretical validity and external or operational validity.

In this paper, the focus is on external validation. Thus, in the remainder of this paper, the term validation will be used to refer to various processes and techniques for addressing the comparability between the simulated world of the computational model and the ''real'' world. For simplicity of exposition in the remainder of this article I will use the term ''real'' data to refer to information gathered through experimental, field, archival, or survey analyses of actual human, animal, physical systems, groups, or organizations.

The emphasis in this paper is on validation as the comparison of simulated and real data. Sometimes, methods for exploring the predictions of models such as sensitivity analysis, response analysis, and response surface modeling are often described as validation techniques. Such techniques, since they do not require the comparison with real data, will not be described herein. However, they are important parts of the computational theorist's toolkit.

Before continuing two important caveats need to be made. The first concerns process and the second concerns presentation. First, validation typically requires a team of researchers and is often a multi-year multi-person endeavor. The argument here is based both on practical as well as training issues. Computational models are currently sufficiently complex that a single researcher in a single research period, e.g., six months to two years, generally cannot build, analyze and validate a computational model. This is due, in large part, to the fact that even intellective models are today much more sophisticated than those presented even 15 years ago, often are built out of multiple sub-models, and often take multiple people-years to build and analyze. A second reason is that the analysis of computational models, particularly stochastic, parameterized, or Monte Carlo models requires doing a series of virtual experiments[2]. These virtual experiments may take less time than a human experiment to run, but the results are empirically comparable. Thus examination of the results often requires the same level of statistical training and analysis as human experiments, and the same amount of time. This problem is exacerbated by the fact that computational models can easily be built, and often need to be built, to generate much larger quantities of empirical data than do human experiments. Such massive amounts of data can, in and of themselves, generate particular analysis problems. Thus, the amount of time and level of research needed to bring a computational model to fruition and to examine its predictions necessitates teaming with other researchers if model validation is to be done within a reasonable time frame. Third, the level of training required of computational theorists is as detailed and specialized as that required of a field researcher or experimentalist and few scientists acquire all the requisite skills for both computational theorizing and field work (or laboratory experiments). The skills and training needed to design and build simulation

[2] The term virtual experiment refers to an experiment in which the "thing" or "agent" whose performance is being monitored is modeled computationally. Even as virtual reality is a computationally generated computational analog of reality, the virtual experiment is a computationally generated computational analog of a human (laboratory) experiment or a human field experiment (natural experiment).

models goes well beyond programming (Salt, 1993), and includes a wide range of activities including, but certainly not limited to, historical analysis, data validation, use analysis, and requirements analysis (Knepell an d Arango, 1993). Even as one would not expect a mathematician to do field work (or experiments) to validate their models (as few would expect mathematicians to have training in such endeavors), one should not expect the computational theorist to do so. In other words, the set of research skills and knowledge necessary for building and validating models are sufficiently distinct that teaming with other researchers is often necessary for validation. Finally, validation is only a small, though significant, component of computational analysis. Computational modeling involves many considerations including providing an appropriate user interface, determining the optimal level of simulation complexity (what should be included in the model, defining a tutorial strategy; selecting a tool for building the simulation, assessing the hardware requirements, identifying the needs of the user, determining the pedagogical goals, validation and verification, and system evaluation (Bergeron and Greenes, 1988). Of these aspects of computational modeling, verification and validation is in many ways the most doable by individuals other than the computational theorist. Indeed, it can be argued that computational theories should in fact be tested by others than the creator as the creator may be biased (albeit unintentionally) in interpreting the results. In summary, teams are necessary for doing model validation, at least given the currently available technology for model development and validation.

The second caveat is that computational models and their output should generally be described and presented independent of, and generally prior to, external validation. Clearly, researchers should take every precaution to ensure that the results they are presenting are produced by error free code; in other words, the models should be internally validated. However, external validation should be presented separately from the model. From a purely presentational point of view, most models cannot be adequately explained, results presented, and validation technique and results described within the pages appropriate to a single journal article. For example, in presenting a computational model the researcher should explain the representation scheme, the information flow, the modular decomposition of the model, and the (where appropriate) underlying parameters. In presenting the results of, e.g., a Monte Carlo model the virtual experiment, the variables, and the method of analysis should be described. The detail required to present a model and the analysis of its results often precludes providing additional information on validation. Further, the provision of validation material in addition to model details and results can often defocus the article and lead to confusion. From a practical point of view, as noted the time required to build and analyze a computational model is quite substantial and validation may require teams. To delay model presentation until validation has occurred retards the development of the scientific field. Finally, many computational models are formal representations of theory and as such require multiple tests. The predictions of many computational models can be thought of as theoretical propositions or hypotheses. In a sense, a computational model can be thought of as a hypothesis generation machine. For computational theories as with most theories, the researchers who test the theory are generally not those who propose the theory. Rather, the computational theory as with non-computational theories may require many different tests, in many different venues. In this sense, validation can become a process of theory verification and extension (Hanneman, 1988). In summary, validation should not be held up as a pre-requisite for the presentation of a computational model and its predictions.

An important interlude here is to discuss the role of Turing tests in the validation of computational models. The essential goal behind a Turing test, as it is generally applied, is to see whether or not the observer can distinguish between results generated in two fashions. These results may be generated by two alternative computational models, by a computational model and an analytical model, by a computational model and an experiment, and so on. Turing tests may or may not be rigorous, may or may not employ the use of quantitative data, and may or may not be carried out statistically. Turing tests have typically been employed in simulations of machines or of single humans. Carley and Newell (1994) suggest that when the computational model is meant to act as a social agent or a group of social agents it is more appropriate to use a revised version of the Turing test which they refer to as the social Turing test. In a sense, the various types and levels of validation discussed in this paper can be thought of as ways of clarifying what is meant by the phrase "whether or not the observer can distinguish between results." The issue is not doing a Turing test, but defining what is meant by a Turing test, and determining what level of test results will be acceptable.

Types of Models

Wide arrays of computational models have been, and are being, used in the social and organizational sciences. These models can be, and have been, classified on a number of dimensions. For example, models can be thought of as: intellective[3] or emulation based, stochastic or deterministic, parameterized or heuristic, enumerative or Monte Carlo. Illustrative models in each of these categories are presented in Table 1. The categories in table 1 are neither exhaustive nor exclusive of each other. Other classifications are assuredly possible and the foregoing typologies are given, in part, for illustrative purposes. The main point at this juncture is that computational models with different characteristics require different evaluation and validation schemes. Further, no single approach to validation is universally applicable to all types of computational models.

Table 1: Illustrative Models by Category

Intellective	Garbage Can Model (Cohen March and Olsen, 1972)
Emulation	Virtual Design Team (Levitt et al. 1994)
Stochastic	Social Exchange (Macy, 1991)
Deterministic	Diffusion (Krackhardt, 1997)
Parameterized	Cultural Transmission (Harrison and Carrol, 1991)
Heuristic	AAIS (Masuch and LaPotin, 1989)
Enumerative	CORP (Lin, 1994)[4]
Monte Carlo	ELM (Carley, 1992)

[3] Cohen and Cyert (1965; p. 308) additionally talk about human-in-the-loop or gaming simulations. Such simulations are not discussed herein as they are not completely computational.

[4] Actually in this model, enumeration is used only when there are no information distortions and the task has nine pieces of information. See also Carley and Lin, forthcoming.

Descriptions of the types and levels of validation will be provided later, but for now, a single example will serve to illustrate the relation between model type and validation. Specifically, engineering or emulation models (also called wind tunnel and kitchen sink models) require high levels of validation as they are typically built with the purpose of providing explicit advice to a particular corporation or on specific problem. The essential motto of such models can be thought of as ‘‘everything critical in the model and model everything that is critical.’’ These models are characterized by a large number of parameters, many modules, and often a detailed user interface. The high level of detail in these models makes it possible to capture the nuances of various trends, technologies, management styles, etc. within the area of concern. Validation is extremely critical as the intent is to provide practical and detailed advice. In contrast, intellective models are characterized by being smaller, modular, and often simplistic in their assumptions. These models require less and lower levels of validation as their purpose is generally to show proof of concept or to illustrate the relative impact of basic explanatory mechanisms. The essential motto of such models can be thought of as ‘‘keep it simple." These models are characterized by few, if any, parameters and simplistic user interfaces. The lack of detail in these models increases their generalizability but decreases their ability to generate specific predictions in applied settings. For intellective models, validation is somewhat less critical.

In a sense, the underlying issue here is balance. It is generally recognized in building computational models that it is important to keep a balance between keeping a model simple an d attaining veridicality; however, the balance point must depend on the purpose of the model. It is important to realize that the balance point between simplicity and veridicality depends on purpose (Burton and Obel, 1995). For example, intellective models with the purpose of demonstrating the theoretical adequacy or inadequacy of some assumption, or concerned with the impact of a specific principle, have the balance point shifted away from veridicality and towards simplicity. Whereas, engineering or emulation models with the purpose of demonstrating the feasibility of a specific approach or measuring the impact of a specific change on an actual system have the balance point shifted away from simplicity and toward veridicality. Again, the approach for validating such models, and the need for validation, varies somewhat with the type of computational model. Essentially, the higher the claimed veridicality, the greater the need for more in-depth and higher levels of validation. As the validation approaches are described, when there are specific requirements of certain types of models those requirements will be described.

Levels of Validation

Model validity can be assessed at various levels and through a series of techniques. At least eight different levels of external validity can be distinguished: face, parameter, process, pattern, point, distributional, value, and theoretical. Face validity requires that the computational model has an appearance such that taken at face value the model seems to jive with reality. Parameter validity occurs when the parameters of the model match reality - values observed for parameters in field, survey, archival or experimental settings. Process validity occurs when the process described by the computational model corresponds to real processes. Pattern validity requires that the pattern of results generated by the computational model matches real patterns of results. Point validity requires that the behavior of the model on each dependent variable, taken one at a time, has the same mean as the real data. In contrast, distributional validity requires that the distribution of results generated by the computational model has the same distributional characteristics as the real data; e.g., means, standard deviations, and shape of results are the same. Whereas, value validity requires that the specific results from the computational model match on a point by point basis the real data. Finally, theoretical validity occurs when the underlying theoretical constructs in the computational model provide a better predictive indicator of real data than does a linear model.

Face, pattern and process validity form a hierarchy of stringency in terms of validation with respect to validating the internal workings of the model. That is, models which have process validity typically have parameter validity and those which have process and para meter validity have face validity Pattern, point, distributional, and value validity form a hierarchy of stringency in terms of validation with respect to validating the model’s results. In this case, models which have value validity have distributional validity. Having distributional validity guarantees point validity. And point validity guarantees pattern validity. Theoretical validation is a joint approach to simultaneously addressing validity of the internal workings of the model and the results that it generates.

Types of Validation

A variety of validation techniques have been used by researchers in various scientific fields. Roughly, these techniques fall in to the following categories: grounding, calibrating, verifying, and harmonizing. Each of these techniques will be described in turn. Many of these techniques can be used at one or more levels of validation.

Grounding

Grounding involves establishing the reasonableness of a computational model. This approach is generally used for establishing the face validity of a model and sometimes its parameter or process validity. This approach is more often used with intellective than with emulation models. Grounding involves the use of storytelling, initialization, and evaluation techniques. When the focus is on initialization grounding provides face validity, and may establish partial parameter or process validity. When the focus is on evaluation, grounding generally only establishes the validity of results at the pattern level. For examples of grounding see Cohen, March and Olsen (1972), Glance and Huberman (1993), Kaufer and Carley (1993); Levinthal and March (1991); and Carley and Svoboda (1996).

The basic goal of grounding is to establish that the simplifications made in designing the model do not seriously detract from its credibility and the likelihood that it will provide important insights. At one level, grounding is largely a matter of storytelling. That is, the author sets forth a claim for why the proposed model is reasonable. This claim is enhanced by not over-claiming the applicability of the model and by discussing the models limitations and scope conditions. Grounding can be enhanced by demonstrating that other researchers have made similar or identical assumptions in their models. Thus, explaining how the proposed model extends, is a special case of, is a generalization of, or competes with one or more other computational or mathematical models is a rhetorical technique for increasing a models grounding. Finally, the grounding claim is enhanced by demonstrating, typically through some type of ethnographic analysis, that the proposed computational model captures the key elements of a specific group, organizational, or social process, or the core ideas in a verbal theory.

Another type of grounding is provided through initialization. Initialization is the process of setting the initial or starting parameters or procedures for the model. This technique is typically used with stochastic, parameterized, and Monte Carlo models. On the initialization front, grounding requires setting the various parameters and procedures so that they match real data. For example, if a computational model is concerned with the impact of advice networks on organizational performance, then setting the range of initial densities for the simulated organizations to include those observed by other researchers who collected network data within organizations grounds the model. Grounding can also be achieved by establishing the boundaries on a process. For example, if empirical studies have shown that individual learning tends to follow logarithmic rather than exponential growth, then setting the growth equation in a model to be logarithmic grounds the model.

Finally, grounding involves simple performance evaluation. Simple performance evaluation is the process of determining whether the computational model generates the stylized results or behavior expected of the underlying processes. First the researcher locates one or more stereotypical facts or stylized behaviors. These facts or behaviors might be thought of as general empirical regularities that have been repeatedly observed with real data. An example is that most studies of populations of organizations exhibit a liability of newness, that is young organizations are more likely to perish than are older organizations. Another example is that most studies of diffusion show an S-shaped adoption curve. Second the researcher demonstrates that the proposed model generates data or exhibits behavior consistent with the stereotypical fact or stylized behavior. The ability of the computational model to generate these stylized results is prima facie evidence for grounding. Such results from the computational model generally are not, and should not be, the only results that can be generated from the model. The point is simply that establishing such non-surprising results first is a form of model validation.

Calibrating

Calibrating is the process of tuning a model to fit detailed real data (see Figure 1). This is a multi-step, often iterative, process in which the model's processes are altered so that the model's predictions come to fit, with reasonable tolerance, a set of detailed real data. This approach is generally used for establishing the feasibility of the computational model; i.e., for showing that it is possible for the model to generate results that match the real data. This approach is more often used with emulation than with intellective models. Calibrating a model may require the researcher to both set and reset parameters and to alter the fundamental programming, procedures, algorithms, or rules in the computational model. Calibrating establishes, to an extent the validity of the internal workings of the model and its results (at least in a single case). The researcher may choose to halt calibration after achieving either a parameter or process level of validation. Further, in terms of results, calibration may halt at any level - pattern, point, distribution or value. For examples of calibration on an emulation model see Levitt et al. (1994) and on an intellective model see Carley (1990).

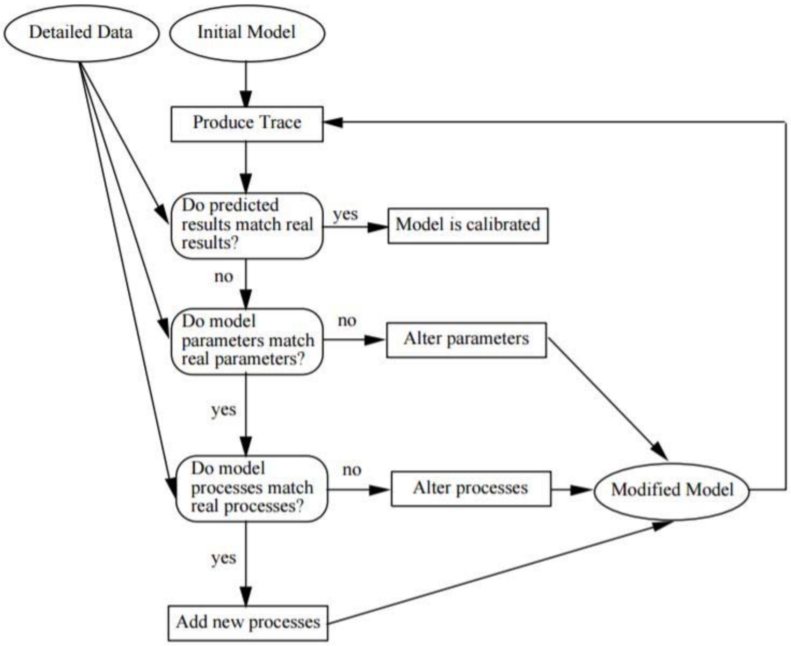

Figure 1: Calibration

To calibrate a model the researcher begins with the uncalibrated model. Then a trace of the model's predictions and the processes that generated them is generated. This information is then checked against real data. If the simulated predictions of the dependent variable(s) matches the real dependent variable the model is considered to be calibrated. Otherwise, first the parameters and then the processes are checked for accuracy. This check may involve going back and talking to experts at doing the task the model seeks to simulate or collecting new observational detail to fill in details or to check the accuracy of the original real data. Once both parameters and processes are accurate, if the model predictions are still not matching the real data, the modeler typically moves to adding additional lower level or auxiliary processes that were originally thought to be less important.

Calibration occurs at two levels. At one level, the models predictions are compared against real data. This can be characterized as analysis of the dependent variable(s). At another level, the processes and parameters within the model are compared with data about the processes and parameters that produced the behavior of concern. This can be characterized as analysis of the independent (and control) variable(s). To calibrate a model it is important to have access to detailed data on one or more cases. Participant observation or other ethnographic data is often the best possible data for calibrating as typically only such data provides the level of detail needed by the modeler at both the process and outcome level. Calibrating models of subject matter experts typically requires interacting with an expert and discussing whether or not the model matches in its reasons and its results the behavior of the expert , and if not, why not.

In calibrating a model, the level of match required between the model and the real data depends in part on the research goals. The level of match also depends on the quality of the real data and the degree to which that data does not represent a pathologic or extreme data point. How should the cases for calibrating the computational model be chosen? The ideal is to use a set of cases that span the key categories across which the model is expected to operate. The next best option is to choose two to four cases that represent typical behavior and one to two that represent important extremes. The basic idea here is that by looking at both the typical and the extreme the boundaries on processes, parameters and outcomes can be set with some degree of confidence. In practice, however, the researcher who wishes to calibrate a model is often lucky to even have one case with sufficient detail. That case, moreover, is often more a matter of opportunity than plan.

Critics of calibration often argue that any model with sufficient parameters can always be adjusted so that some combination of parameters generates the observed data. Thus, the argument proceeds, calibration does not establish the validity of a model in a meaningful way. At one level, this criticism has some truth in it for some models. In particular, large multi-parameter models often run the risk of having so many parameters that there is no guarantee that the model is doing anything more than curve fitting. However, for many computational models this criticism is less appropriate. In particular, for models where the process is represented not by parameterized equations but by rules, interactive processes, or a combination of procedures and heuristics there are often few if any parameters. There is no guarantee that a sufficiently large set of procedure and heuristics, that often interact in complex and non-linear ways, can be altered so that they will generate the observed data. For procedural models, calibration becomes a process of altering ‘‘how things are done’’ rather than ‘‘how things are weighted.’’ This distinction is critical as it separates process matching from curve fitting.

Verification

Verification is a set of techniques for determining the validity of a computational model’s predictions relative to a set of real data. To verify a model, the model’s predictions are compared graphically or statistically with the real data (Kleijnen, 1995b). Verification, though rarely used, is a necessary step in moving a model from the theoretical to the applied realm and is a necessary step in establishing the accuracy of the theory embodied in the model. Verification demonstrates that a model’s predictions match real data. During verification the focus is on validating the model’s results not its internal workings. Verification can be done on any type of computational model. In the verification process, unlike calibration, the model is not altered. Further, the level of detail needed in the real data for verification is less than the level of detail needed for calibration. The type of real data available will determine whether the models is validated at the pattern, point, distribution, or value level, or some combination of these. Further, verification is sometimes done on uncalibrated models, particularly for intellective models. For examples of verification see Cyert and March (1963), Carley (1990), Gibson and Plaut (1995).

The type of statistical analysis used for verification depends on the nature of the data being examined. For example, Gibson and Plaut (1995) use graphical techniques for demonstrating pattern level verification for their connectionist based dynamic learning model. Whereas, in Gibson, Fichman, and Plaut (1996), show the real and predicted data side-by-side in graphs, but base their conclusions on statistical hypothesis tests. These tests are at both the pattern level (e.g., is the linear trend predicted by the computational model present in the real data?) and the point level (e.g., is the level of real results significantly different from that predicted by the model?).

Verification can be done in one or more stages (see Figure 2). The basic idea, is that a particular set of real data may have some type of bias or pathology in it. Thus, verification against multiple data sets is an even stronger validation of a model than verification against a single data set. These multiple verifications can be done on data sets drawn for different purposes or at different times, or if the sample of real data is large enough on two mutually exclusive subsets of a single data set. Sometimes, verification is combined with calibration. That is, some of the data used for calibration is re-used as part of the first data set the model is verified against. It should be noted, that this type of re-verification is useful mainly if the model is to be used in an applied setting or if the point of the verification is to test the theory embodied in the computational model. Baligh, Burton and Obel (1994; see also 1990) employ a similar multi-stage approach in their validation of the Organizational Consultant which is a heuristic model.

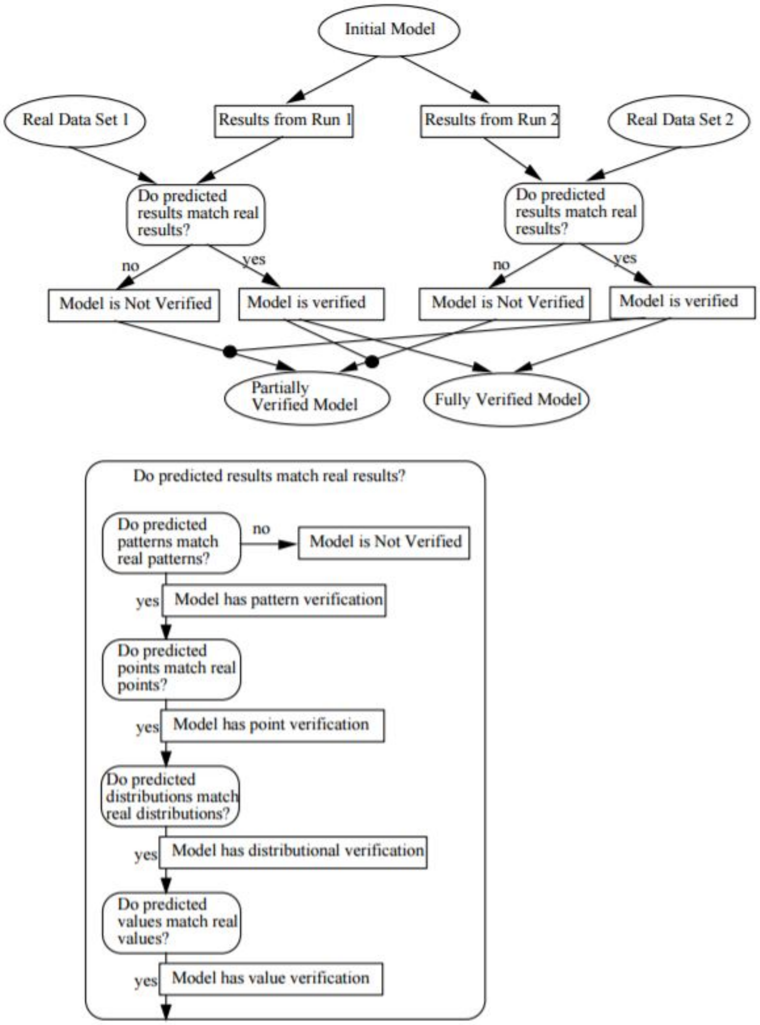
Initial Model
Real Data Set 1
Results from Run 1
Results from Run 2
Real Data Set 2
Do predicted results match real results?
Do predicted results match real results?
no
yes
no
yes
Model is Not Verified
Model is verified
Model is Not Verified
Model is verified
Partially Verified Model
Fully Verified Model
Do predicted results match real results?
Do predicted patterns match real patterns?
no
Model is Not Verified
yes
Model has pattern verification
Do predicted points match real points?
yes
Model has point verification
Do predicted distributions match real distributions?
yes
Model has distributional verification
Do predicted values match real values?
yes
Model has value verification

Figure 2: Verification

A special issue in verification occurs with respect to multi-agent models. Multi-agent models can potentially undergo dual level verification; i.e., verification at both the individual and group level. To wit, does the model accurately predict group level behavior, individual level behavior, or both? In this case, the purpose of the model determines the requisite level of verification. If the purpose of the model is to explain group level phenomena based on the actions of generic agents, then the model should be verified at the group level. In this case, the researcher would verify the model by collecting data on multiple groups about the phenomena of interest and then statistically compare model predictions to actual data on groups. Alternatively, if the purpose of the model is to examine what group level phenomena might emerge given the specific actions of particular agents and the known variation in individual behavior, then the model should be verified at the individual level. In this case, the researcher would verify the model by collecting data on individuals about the actions of interest and various characteristics and then statistically compare changes in individual behavior predicted by the model over time with the actual data on individuals.

Dual level verification is rarely done; i.e., rarely are models verified at both the individual and the group level. In fact, there are ongoing debates about whether verification should be done at both levels. There are several arguments against dual-level verification. The first is that group level behavior is an emergent phenomena that requires only on-average understanding of individuals and not specific behavior and that individual differences get averaged out and are not important at the group level. The second line of reasoning is that group level behavior is more than the aggregate of individual behavior and that holistic effects come into play over and above individual actions. In contrast, those arguing for dual-level verification argue that group level behavior is an aggregate of individual level behavior and correct group level behavior can only emerge from correct individual level behavior.

Verification is a rigorous statistically based approach to validation. A potential error in verification is the use of non-comparable data. That is, for comparability the results of the computational model and the real data should be obtained under comparable situations and environmental conditions (Kleijnen, 1995a). Such comparability can be enhanced by setting input data and parameters in the computational model to resemble as closely as possible those in the real situation. When the situation conditions are not known, the researcher should run a sensitivity analysis in order to determine whether variations in the model's inputs generate variations in the model's outputs that correspond with the expert's intuition.

Harmonization

Harmonization is a set of techniques for determining the theoretical adequacy of a verified computational model relative to a linear model and a set of non-computational data (Kaplan, Miller and Carley; 1996). The goal of harmonizing is to show that the theoretical assumptions embodied in the computational model are well grounded, or in harmony with the real world. This is done through a multi-step validation process and two sets of real data (see Figure 3). First the computational model is calibrated against detailed data and then verified against the first set of real data. Then the model is re-verified against a second set of real data. Next, a linear model is estimated on the first set of real data. The linear model serves as a benchmark[5] and is used to generate well understood and robust results which are not constrained by the theoretical assumptions embodied in the computational model. Rather, these results from the linear model are simply designed to provide the best fit of the data given the constraints of linearity. The next step is cross-validation. Cross-validation involves using the first set of real data, estimating the parameters for a simple linear model on that data, and then, given those parameters, predicting the behavior of a second set of real data (Stone, 1994). The final step involves statistically

[5] These linear models can also be thought of, and may indeed represent, baseline models. Although, they are rarely Mayhewian style baseline models (Mayhew, 1984).

contrasting the computational model's predictions and the linear model's predictions for the second set of real data.

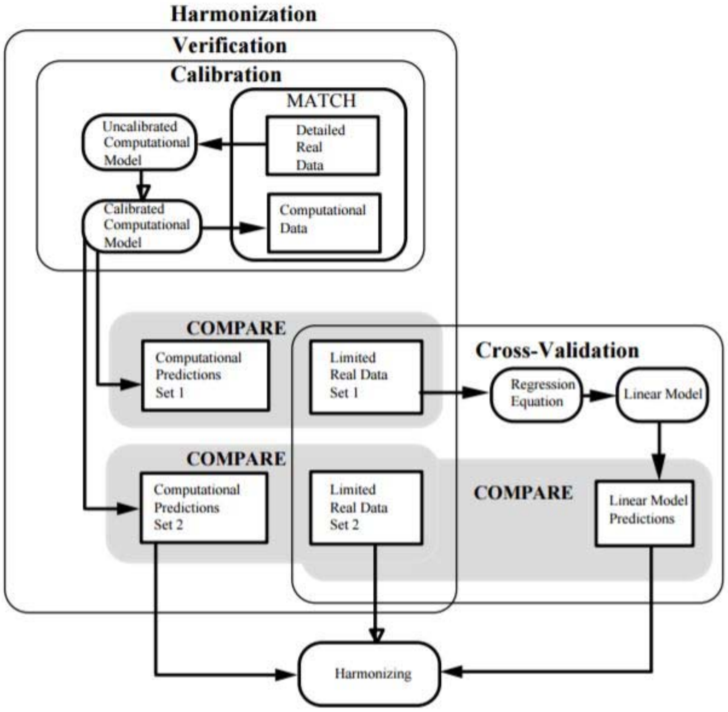

Figure 3: Harmonization

Linear models have been characterized as the best understood and most robust set of models that are statistically sound but without contextual theoretical assumptions. For linear models, the cross-validated models should have good predictive power (Hammond, 1955; Hoffman, 1960; Slovic and Lichtenstein, 1971; Dawes 1979). Computational models are often developed to take into account known non-linearities and are based on, often quite strong, contextual theoretical assumptions. Comparison of the predictions of the computational model with the predictions of a linear model is a way of statistically addressing the adequacy of these theoretical assumptions and their predictive value over and above that achievable through a simple linear benchmark model. In this sense, one might argue that harmonization allows the researcher to explore both the predictive and theoretical adequacy of the computational model. How the actual comparisons should be made, however, depends on the type of data. Harmonizing can be a valuable step in changing a computational model from an intellective or normative device into a prescriptive device.

Harmonization is certainly a valuable empirical technique for locating areas or assumptions of the model that need to be improved. The classical, more common, but non-empirical approach, for locating areas of a computational model that need to be improved, is the Delphi test. Delphi tests are non-rigorous holistic tests of a computational model's adequacy with the real world. In a Delphi test a panel of experts on the item being modeled evaluate the adequacy of the computational model. Basically, the panel of experts look for areas in the computational model that in their judgment are not sufficiently comparable to the real world. Delphi tests generate a series of areas in which the computational model should be improved. Delphi tests are often used for large or emulation style models.

In doing harmonization, the researcher should have available two samples of real data both sufficient in size for statistical analysis. Note, these two samples can be two halves of the same data set. One approach to harmonization (Kaplan, Miller, and Carley, 1996) requires there to be a set of dependent variables, and then for each dependent variable multiple sets of predictions, multiple R^2s, for both the computational and linear models. The actual comparison between the computational and linear model can be made by running a Pearson correlation on the R^2s reported by both the computational and the cross-validated linear model across a set of conditions. In this approach, a positive correlation suggests that both models are doing well and poorly in the same areas; whereas, a negative correlation suggests that the models mirror each other and so a strong prediction for one model translates into a weak prediction for the other. A correlation near zero suggests that the models have different strengths and weaknesses.

A key aspect of harmonization is that it requires there to be a reasonable linear model. There are many sources for such model. One option is to use the inputs to the computational model as the elements of a linear model. A second option is to use data that may or may not be used in the computational model but that is easily, and typically, collected in order to predict the dependent variable (such as age, education, and parent’s education in predicting a person’s income) by the user. A third option is to use as the linear model, a model that has been presented in the literature by other researchers. Regardless of the option chosen for generating the linear model, the researcher should then cross-validate that model against real data.

Discussion

Researchers have long called for the validation of computational models. Indeed, early modelers themselves called for validation (e.g. Cyert and March, 1963). The importance of model validation for scientific advancement cannot be denied. But, what is the state of model validation in the social and organizational sciences? Further, under what conditions should models be validated?

To begin with, few computational models are validated. Indeed model validation within the social and organizational sciences is in its infancy in many ways (Andreoni and Miller, 1995). Within computer science, validation is also an emerging concern. For example, recent work on RISC processors (Bose, 1995) points out that verification is becoming increasingly important as the single-chip processors are becoming sufficiently complex that designer intuition is no longer sufficient for ensuring performance bugs. Most artificial intelligence models, for example, are typically only validated using calibration techniques. The state of validation is somewhat more advanced within engineering and operations research (Knepell and Arangno, 1993; Gass 1980; Kleijnen, 1995a). Indeed, much of the work on the validation and verification of computational models has occurred in the engineering, operations research, and in military contexts (see for example, Naylor and Finger, 1963, Gass 1983 or Gados, 1989). For additional reviews on models outside of the social and organizational sciences see Banks, Gerstein and Searles (1987, 1990) and Balci (1987).

Within the social and organizational sciences when validation is done, the type of validation typically done depends on the nature of the model. For example, calibration is generally done only for emulation models and grounding is generally done only for intellective models. Rarely are other types of validation used. In a sense, grounding and calibration serve similar functions;

i.e., they both establish the reasonableness of a model and its potential for predictive accuracy. Grounding is uniquely suited to the intellective model where the goal is to examine general principles and not to provide detailed guidance on a specific example. For intellective models, calibration may be neither feasible nor warranted. In contrast, calibration is particularly appropriate and necessary for emulation models whether the goal is to provide detailed guidance in a specific situation as calibration establishes the particularity of the results. Neither validation technique establishes the general accuracy and overall performance of the computational model. To move computational models into the applied realm, particularly into the realm where they will be repeatedly applied, other validation techniques such as verification are needed.

A review was provided of the basic nature of computational models in the social and organizational sciences, types of validation, and some of the issues in doing model validation. Both specific validation techniques and levels of validation were described. In the foregoing discussion, for the sake of clarity, validity was often discussed as an all or none proposition. That is, a model either is or is not valid. However, as Law and Kelton (1991) argue, validity is more a matter of degree. A model is not simply valid or not valid; rather, it has a certain degree of validity. The reader should keep this caveat in mind in interpreting and applying the foregoing discussion to any particular model.

In the foregoing discussion, it was suggested that the validation of computational models typically is best done by teams of researchers and separate from the development and analysis of the computational model. Further it was suggested that not all models should be validated. Often computational model are built to show proof of concept or to show that the predictions made from a verbal theory cannot follow from, or are inconsistent with, that specific theory. In these cases, validation is not of essence as the model itself is not serving as a new theory. Whereas, if the computational model is thought to embody a new theory, then validation is important as it is with any theory. In this case, however, the process of validation is, as with any theory, a drawn out process, often requiring the collecting of new types of data in new ways, and involving multiple researchers, over a period of years.

Many times the designing and building of a computational model requires the researcher to move well beyond extant knowledge of the system being simulated. For example, to model population ecology requires operationalizing concepts such as performance. In cases such as this, the modeler proceeds by making reasonable assumptions about how the system works. These assumptions have the same epistemological standing as axioms in a mathematical theory. Validation helps to determine the theoretical adequacy of these assumptions.

A final question centers around, what factors facilitate model validation? As was noted, validation may often be done by researchers other than the modeler and often long after the model was developed. To date, most successful validations seem to have been done by teams of researchers which include both the modeler(s), and other researchers over a period of multiple years. Examples here include the work on the virtual design team (Cohen, 1992; Levitt, Cohen, Kunz, Nass, Christiansen, and Jin, 1994), soar (Laird, Newell and Rosenbloom, 1987; Rosenbloom, Laird, Newell and McCarl, 1991), the organizational consultant (Baligh, Burton and Obel, 1990; 1994, Burton, and Obel, 1984, 1995), and on the models building off of ELM (Carley, 1992; Ye and Carley, 1995; Carley and Lin, 1995; Carley, Prietula and Lin, 1998). Nevertheless, there are several things that can be done more generally to facilitate validation. First, models that have a flexible user-interface are generally easier to validate. That is, models where the user-interface has been designed so that the user can easily alter the basic parameters, input data, and types of output without having to recode or recompile the underlying software admit validation by researchers other than the modeler. The reason is that such a flexible user-interface makes it easier for these other researchers to simply rerun the model with the specific required parameters, inputs and outputs. Second, it is easier to validate models that have been adequately described in terms of the basic components - input, output, internal mechanism or processes, initial conditions, parameters, boundary conditions, and limitations of the model. Without such description, at least in a manual for the model, other researchers cannot engage in any but the simplest of validation approaches unless they work with the modeler. Third, it is easier to validate models where virtual experiments based on the model are adequately described in terms of experimental design, variable definitions, n-size, basic parameters, and possible biases. A virtual experiment is an experiment run using a computational model. The resultant data, like that from a laboratory experiment, can be meta-analyzed and used by subsequent researchers if sufficient information is provided. Fourth, if the modeler retains archives of model results or maintains a record of the parameters and input data and program version used to generate particular results future researchers can validate these results first by regenerating them and then contrasting them with real data. Finally, models that have been developed recently are more likely to be validated as the code is still likely to be available and runnable with minimal overhead.

The process of validating computational models and techniques for performing such validation that have been and are being used in the social and organizational sciences have been described. There are other possible techniques and more will assuredly be developed over time, particularly as researchers in this area begin to import techniques standard in engineering. The attempt here was not to provide a comprehensive review. Rather, the goal was to show the range of techniques, to present some guidelines for when what techniques are appropriate, and an understanding of how these techniques are currently applied. As has been described, validation is a complex and multi-faceted process; however, it is a critical process in the testing and development of general computational theories and specialized applications.

References

Andreoni, J., & Miller, J .H. (1995). Auctions with Artificial Adaptive Agents. *Games and Economic Behavior*. *10*: 39-64.

Axtell, R., Axelrod, R., Epstein, J.M. & Cohen, M.D. (1996). Aligning Simulation Models: A Case Study and Results. *Computational and Mathematical Organization Theory*, 1(2): 123-142.

Balci, O. (1987). Credibility and Assessment of Simulation Results: The State of the Art. *in Methodology and Validation Proceedings of the conference on methodology and validation, Orlando Fl.*; 6-9 April, 1987, edited by O. Balci. San Diego, CA. Society for Computer Simulation.

Baligh, H. H., Burton, R. M. & Obel, B. (1990). Devising Expert Systems in Organization Theory: The Organizational Consultant. In M. Masuch & W. De Gruyter (Eds.), *Organization, Management, and Expert Systems* (pp.35-37). Berlin.

Baligh, H. H., Burton, R. M. & Obel, B. (1994). Validating the Organizational Consultant on the Fly. In K.M. Carley & M. J. Prietula (Eds.), *Computational Organization Theory* (pp. 179-194). Hillsdale, N J.: Lawrence Erlbaum Associates.

Banks, J., Gerstein, S. & Searles, S.P. (1990). Verification and Validation of Large Scale Simulation Models. *in Proceedings of the 1990 UKSC Conference on Computer Simulation; Brighton, UK*; Sept. 5-7, pp. 1-6. Available from: Mr. K.G.Nock, UKSC Hon. Treasurer, c/o Scientific Computers Ltd. Victoria Road, Burgess Hill, West Sussex, UK.

Banks, J., Gerstein, S. & Searles, S.P. (1987). Modeling Processes,Validation, and Verification of Complex Simulations: A Survey. *in Methodology and Validation Proceedings of the conference on methodology and validation, Orlando Fl.*; 6-9 April, 1987, pp. 13-18, edited by O. Balci. San Diego, CA. Society for Computer Simulation.

Bergeron, B.P., & Greenes, R.A. (1988). Modeling and Simulation in Medicine: The State of the Art. *in Proceedings of the Twelfth Annual Symposium on Computer Applications in Medical Care, Washington D.C.*; Nov. 6, 1988, IEEE Computer Society Press, pp. 282-286.

Bose, P. (1995). Performance Analysis and Verification of Super Scalar Processors. International Business Machines Corporation. Research Division; 20094, Tech report 20, IBM, T. J. Watson Research Center; Yorktown Heights.

Burton, R.M., & Obel, B. (1995). The Validity of Computational Models in Organization Science: From Model Realism to Purpose of the Model. *Computational and Mathematical Organization Theory*, 1(1): 57-72.

Burton, R.M., & Obel, B. (1984). *Designing Efficient Organizations: Modeling and Experimentation*. Amsterdam: Elsevier Science.

Carley, K.M. (1996). A Comparison of Artificial and Human Organizations. *Journal of Economic Behavior and Organization*, 31(2): 175-191.

Carley, K.M. (1992). Organizational Learning and Personnel Turnover. *Organization Science*, 3(1): 2-46.

Carley, K.M. (1990). Group Stability: A Socio-Cognitive Approach. In E. Lawler, B. Markovsky, C. Ridgeway, & H. Walker (Eds.) *Advances in Group Processes: Theory and Research. Vol. VII* (pp. 1-44). Greenwhich, CN.: JAI Press.

Carley, K.M., & Lin, Z. (1997). A Theoretical Study of Organizational Performance under Information Distortion. *Management Science*, 43(7): 976-997.

Carley, K.M., & Lin, Z. (1995). Organizational Designs Suited to High Performance Under Stress. *IEEE - Systems Man and Cybernetics*, 25(1): 221 -230.

Carley, K.M., & Newell, A. (1994). The Nature of the Social Agent. *Journal of Mathematical Sociology*, 19(4): 221 -262.

Carley, K.M., Prietula, M.J. & Lin, Z. (1998). Design Versus Cognition: The Interaction of Agent Cognition and Organizational Design on Organizational Performance. *Journal of Artificial Societies and Social Simulation*, 1, issue 3.

Carley, K.M., & Svoboda, D.M. (1996). Modeling Organizational Adaptation as a Simulated Annealing Process. *Sociological Methods and Research*, 25(1): 138 -168.

Cohen, G.P. (1992). The Virtual Design Team: An Information Processing Model of Coordination in Project Design Teams. *Ph.D. Dissertation, Stanford University, Department of Civil Engineering*. Stanford, CA.

Cohen, K.J., & Cyert, R.M. (1965). Simulation of Organizational Behavior. In J.G. March (Ed.) *Handbook of Organizations* (pp. 305-334). Chicago, IL: Rand McNally.

Cohen, M.D., March, J.G., & Olsen, J.P. (1972). A Garbage Can Model of Organizational Choice. *Administrative Sciences Quarterly*, 17(1): 1- 25.

Cyert, R.M., & March, J.G. (1963). *A Behavioral Theory of the Firm*. Englewood Cliffs, NJ: Prentice-Hall.

Cyert, R.M. (1994). *Foreword to Computational Organization Theory*, Carley, K.M. and Prietula, M.J., (Eds.). Hillsdale, NJ: Lawrence Erlbaum Associates.

Dawes, R. (1979). The Robust Beauty of Improper Linear Models in Decision Making. *American Psychologist*, 34: 571 -582.

Gados, R.G. (1989). Guidelines for Evaluating Simulation Models of the Strategic Defense System. MITRE Tech. Report 9463, April.

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

Gass, S.I. (Ed.) (1981). Validation and Assessment of Energy Models. *in Proceedings of a symposium held at the National Bureau of Standards*, Gaithersburg, MD: May 19-21, 1980.

Gibson, F.P. & Plaut, D.C. (1995). A Connectionist Formulation of Learning in Dynamic Decision-Making Tasks. *in Proceedings of the 17th Annual Conference of the Cognitive Science Society, Hillsdale, NJ.*: Lawrence Erlbaum and Associates.

Gibson, F.P., Fichman, M. & Plaut, D.C. (1997). Learning in dynamic decision tasks: Computational model and empirical evidence. *Organizational Behavior and Human Decision Processes*, 71(1): 1-35.

Glance, N.S., & Huberman, B.A. (1993). The Outbreak of Cooperation. *Journal of Mathematical Sociology,* 17(4): 281 -302.

Hammond, K.R. (1955). Probabilistic functioning and the clinical method. *Psychological Review,* 101: 252 -254.

Hanneman, R. (1988). *Computer-Assisted Theory Building: Modeling Dynamic Social Systems*. Beverly Hills, CA: Sage.

Harrison, J.R. & Carrol, G.R. (1991). Keeping the Faith: A Model of Cultural Transmission in Formal Organizations. *Administrative Science Quarterly*, 36: 552-582.

Hoffman, P.J. (1960). The Paramorphic Representation of Clinical Judgment. *Psychological Bulletin*, 57: 116-131.

Kaplan, D.J., Miller, M.D. & Carley, K.M. (1996). Harmonization of Computational Models and Experimental Data: An Illustration Using Data from the Impact of Wearable Computers on Cooperative Work. *Working paper, Social and Decision Sciences,* Carnegie Mellon University, Pittsburgh, PA.

Kaufer, D. & Carley, K.M. (1993). *Communication at a Distance: The Effect of Print on Socio-Cultural Organization and Change*. Hillsdale, N J: Lawrence Erlbaum Associates.

Kleijnen, J.P.C. (1995). Statistical Validation of Simulation Models. *European Journal of Operational Research*, 87(1): 21-34.

Kleijnen, J.P.C. (1995). Verification and Validation of Simulation Models. *European Journal of Operational Research*, 82(1): 145 -162.

Knepell, P.L., & Arangno, D.C. (1993). *Simulation Validation: A Confidence Assessment Methodology*. Los Alamitos, CA: IEEE Computer Society Press.

Krackhardt, D. (1997). Organizational Viscosity and the Diffusion of Controversial Information. *The Journal of Mathematical Sociology*, 22(2): 177-199.

Laird, J.E., Newell, A. & Rosenbloom, P.S. (1987). Soar: An Architecture for General Intelligence. *Artificial Intelligence*, 33: 1-64.

Law, A.M., & Kelton, D. (1991). *Simulation Modeling and Analysis. 2nd Ed*. New York, NY: McGraw-Hill.

Levinthal, D., & March, J.G. (1981). A Model of Adaptive Organizational Search. *Journal of Economic Behavior and Organization*, 2: 307 -333.

Levitt, R.E., Cohen, G.P., Kunz, J.C., et al. (1994). The 'Virtual Design Team': Simulating How Organization Structure and Information Processing Tools Affect Team Performance. In K.M. Carley & M.J. Prietula (Eds.) *Computational Organization Theory* (pp. 1-18). Hillsdale, NJ: Lawrence Erlbaum Associates.

Lin, Z. (1994). A Theoretical Evaluation of Organizational Measures Regarding Predictability of Organizational Performance. In K.M. Carley & M.J. Prietula (Eds.) *Computational Organization Theory* (pp. 113-160). Hillsdale, NJ: Lawrence Erlbaum Associates.

Macy, M.W. (1991a). Learning to Cooperate: Stochastic and Tacit Collusion in Social Exchange. *American Journal of Sociology*, 97(3): 808 -43.

Masuch, M. & LaPotin, P. (1989). Beyond Garbage Cans: An AI Model of Organizational Choice. *Administrative Science Quarterly*, 34: 38-67.

Mayhew, B. (1984). Baseline Models of Sociological Phenomena. *The Journal of Mathematical Sociology,* 9: 259-281.

Naylor, T.H. & Finger, J.M. (1963). Verification of Computer Simulation Models. *Management Science*, 14(2).

Rosenbloom, P.S., Laird, J.E., Newell, A. & McCarl, R. (1991). A Preliminary Analysis of the Soar Architecture as a Basis for General Intelligence. *Artificial Intelligence,* 47: 289 - 325.

Salt, J.D. (1993). Simulation Should be Easy and Fun. *in Proceedings of 1993 Winter Simulation Conference, Los Angeles, CA*; Dec. 12-15. G.W. Evans, M. Mollaghasemi, E.C. Russell & W.E. Biles (Eds.). New York, NY: IEEE, pp. 1-5.

Slovic, P. & Lichtenstein, S. (1971). Comparison of Bayesian and Regression Approaches to the Study of Information Processing in Judgment. *Organizational Behavior and Human Performance,* 6: 649-744.

Stone, M. (1994). Cross-validation and Multinomial Prediction. *Biometrika,* 61(3): 509-515.

Ye, M. & Carley, K.M. (1995). Radar-Soar: Towards An Artificial Organization Composed of Intelligent Agents. *Journal of Mathematical Sociology*, 20(2,3): 219-246.