COMPUTATIONAL MODELING FOR INDUSTRIAL-ORGANIZATIONAL PSYCHOLOGISTS

Edited by Jeffrey B. Vancouver, Mo Wang and Justin M. Weinhardt

First published 2024

ISBN: 978-1-032-48375-7 (hbk)
ISBN: 978-1-032-48385-6 (pbk)
ISBN: 978-1-003-38885-2 (ebk)

1

BETTER THEORY, METHODS, AND PRACTICE THROUGH COMPUTATIONAL MODELING

Jeffrey B. Vancouver, Justin M. Weinhardt, and Mo Wang

DOI: 10.4324/9781003388852-2

Routledge
Taylor & Francis Group
NEW YORK AND LONDON

SIOP
SOCIETY for INDUSTRIAL and ORGANIZATIONAL PSYCHOLOGY
ORGANIZATIONAL FRONTIERS SERIES

1

BETTER THEORY, METHODS, AND PRACTICE THROUGH COMPUTATIONAL MODELING

Jeffrey B. Vancouver, Justin M. Weinhardt, and Mo Wang

It has been said that human behavior is much like the weather—totally unpredictable. Indeed, as recently as the 1980s predicting where, when, and how intense a hurricane might be when it hit land was largely based on regression models that were of little practical use, particularly beyond 24 hours. However, after Hurricane Andrew hit the coast of Florida in 1992, the national weather service and other agencies began ramping up and refining computational models of hurricanes. Since then, predictions regarding hurricane trajectories and their intensities over those trajectories have improved dramatically, allowing local authorities to determine evacuation needs with enough certainty to obtain much more compliance than in the past. Meanwhile, climate models, which reside at a much higher level of abstraction and a much grosser timescale (i.e., are less granular) than weather models, have been instrumental in validating the causes and consequences of greenhouse gases, as well as provide a sobering forecast of the future with or without change in human behavior. These models have motivated authorities to make policies and set goals to try and address the issue. On a different front, in 2020 the United States and many other governments shut down their economies to "flatten the curve" projected by a computational model of disease spread and death due to a coronavirus known as COVID- 19. Within a year, a vaccine for the virus was produced, thanks in part to a computational model of coronaviruses. Indeed, the natural sciences rely heavily on computational and mathematical modeling to explicate their theories and provide a basis for applying those theories.

In each of the previous three scenarios, the computational models provided scientists and policy makers with not only better scientific knowledge but also more practical knowledge. However, although scholars got better at predicting

DOI: 10.4324/9781003388852-2

the weather, our ability to predict and understand human behavior has not progressed at the same rate. Thus, although we have useful computational models of climate change, hurricanes, and infectious disease spread, we also need computational models to help us understand and predict human responses to these same events. For example, a computational model of a hurricane and where it will touch down is of limited use if the people in the path of the hurricane do not leave the area. We must not only understand the hurricane, but we also must understand the people subject to the hurricane's wrath.

Thankfully, the quest to understand and predict human behavior with computational models has begun in earnest in psychology (Farrell & Lewandowsky, 2010; Sun, 2008) and other fields relevant to industrial-organizational (I-O) psychology like organizational science (e.g., Lomi & Larsen, 2001) and economics (e.g., Lee et al., 1997). Computational models are a set of rules (e.g., equations or formally specified propositions) used to represent processes. Moreover, the rules in computational models are represented in a way that can be simulated. The simulations reveal how behavior emerges from the processes represented, and how they interact with one another over time. Fortunately, the ability to simulate computational models means one does not need to solve the equations to determine how the system will behave or what states it will predict. Indeed, one does not need to be a mathematician to work with computational models. However, one does need to be a student of the phenomenon and systems they are studying.

One area of inquiry where phenomena arise from single or multiple systems is I-O psychology. Specifically, I-O focuses on the human system operating within an organizational system. Indeed, the field has developed a substantial amount of understanding of the phenomena regarding individuals and sets of individuals (e.g., teams) working within organizations, including a growing recognition of its dynamic, interactive quality. Toward that end, increasing attention has been paid to measurement, design, and modeling issues related to data collection and analysis (Wang et al., 2016). On the other hand, there has been relatively less recognition that dynamics, whether across multiple systems, time, or both, create challenges for theory specification. Fortunately, the field has begun to appreciate the usefulness and value of computational modeling when it comes to developing, representing, and communicating theory and the measurement, prediction, and applications that might emerge from this formal approach to theory representation (e.g., Kozlowski et al., 2013).

Still, that appreciation is more narrowly represented in the field than it should be given the nature of the phenomena studied by I-O psychologists. To help expand appreciation for computational modeling, we developed this edited book. As discussed in more detail later, the book includes chapters that review key modeling achievements in several domains relevant to I-O psychology, including decision making, diversity and inclusion, learning and training, leadership, and teams. Moreover, the book provides specific how-to-chapters on the two most used modeling approaches in the field: agent-based modeling and system dynamics modeling. It also provides information about how to evaluate models qualitatively and quantitatively. Finally, it provides advice on how to read, review, and publish papers with computational models.

Meanwhile, to help motivate the current readers' interest in computational modeling, we provide a description of the myriad of values computational modeling can bring to our science. We also provide a brief history of computational modeling as it relates to the field of I-O psychology. This is followed by a more complete description of our goals for the book. That is, we describe the learning objectives for various levels of computational model aficionados, from scholarly consumers to computational model creators. Finally, we provide an overview of the chapters.

The Value of Computational Modeling

It is not difficult to articulate the various ways computational modeling can serve a science. Foremost, it is important to understand that computational modeling is not a statistical technique; rather, it is a way of theorizing. Most often, the theorizing represented in computational models began more informally as natural language or verbal theories (Busemeyer & Diederich, 2010). Computational modeling can provide formality and thus discipline to such theories (Farrell & Lewandowsky, 2010). This includes confirmation of a theory's viability and a greater degree of transparency regarding a theory's explanation. Moreover, formal modeling can facilitate integration, generalization, and differentiation of theories and constructs, given the abstract universality of mathematics. Computational models also provide tools for critiquing theory and interpreting findings, discovering new phenomena, guiding empirical needs and protocols, and providing better prediction and prescription (e.g., Harrison et al., 2007). We note that the value of computational modeling is not only via the representations of theoretical processes that can be simulated, given their formal rendering, but also in the *process of computational modeling*, from start to finish. That is, computational *models* have value, but computational *modeling* also has value, given the thinking processes, supported by the modeling platforms, that are needed to create a working model of the entities and phenomena of interest (Farrell & Lewandowsky, 2010; Hintzman, 1990).

However, before extolling further on the virtues of computational models, a warning against applying the list of values too strictly is needed. Consider, I-O psychologists attempt to address an array of practical problems via the application of understanding of the phenomena of the individual or individuals in some context (e.g., work). This understanding is often derived from one or more general theories that appear relevant to the particular problem at hand. Additionally, it is possible to articulate multiple understandings or alternative explanations. To support the application of a particular understanding, it is generally required to identify patterns in the data that are consistent with the understanding (i.e., hypotheses). These patterns must be substantiated by data that reject null hypotheses and collected in a way as to minimize the presence of alternative explanations. What is generally *not* required is that the understanding be represented in a form that can *generate* the patterns of data presumed to follow from the understanding/theory. Rather, one must rely on an understanding of the theory and one's mental model to imagine the patterns that might appear in data (i.e., hypotheses generation) because typical understandings are expressed in informal ways.

This is important because when one is evaluating a computational model, one should be wary of asking the computational model to do more than a verbal theory does. This may seem ironic, given we are about to go into detail extolling the value that computational models over verbal theories. Yet, no computational model will be able to realize all the values. Our greatest fear in articulating a list of values is that reviewers will translate the list into a checklist when critiquing a computational model or modeling paper. Such a checklist is almost certainly going to doom the paper. Indeed, if one is tempted as a reviewer to start a sentence "aren't most computational models supposed to . . ." and end it with ". . . but this one does not," *please resist*. Many responses to reviewers' comments have been about explaining why some model or set of models in question should *not* be expected to realize this or that criterion. To be sure, all papers and presumably the models within them should add some value or there is no point to the model/paper. That is, the same criteria for all scholarly contributions should be applied to computational models—some value is added. The expectation is not, however, that a scholarly product (e.g., an empirical paper) could maximize on all the desideratum (McGrath, 1981). This same affordance should be offered to computational modeling products. Meanwhile, we believe a lot of value will be added by a field that embraces computational modeling in general. We begin our discussion of the value of modeling at this very general level.

The Need for a New Paradigm

Heidegger (1927, 1996) proposed that a field is typically absorbed in its practices without questioning them or their limitations. Similarly, Kuhn (1962, 2012) proposed that most of the time scholars operate under "normal science," where scholars problem solve and operate under the dominant paradigm. However, when there are some significant anomalies, breakdown, or disruption, a field might reflect on its practices more closely. Currently, psychology has been reflecting on its practices in the face of issues related to replication, questionable research practices, lack of testable theories, theory proliferation, construct proliferation, weak prediction, and the generalizability of our theory and findings.

One outcome of such reflection can be a new paradigm. For us, that new paradigm is computational modeling. To be sure, we are not naïve enough to believe that computational modeling is the only solution to all the disruptions noted previously. Indeed, to some extent, some may think that computational modeling is more unique from current ways of doing science than it in fact is. However, as explained in the following paragraphs, we believe computational modeling provides a path to addressing many of the issues or at least a new perspective on them.

A Different Way of Thinking: Need for a Working Model

The first major specific benefit of modeling is that it forces one to think more completely about a phenomenon and the explanations forwarded about it. Indeed, Farrell and Lewandowsky (2010) point out that simply trying to construct a computational model may substantially improve one's reasoning about a phenomenon and/or a theory because simulations of the model directly test one's reasoning. That is, rather than just saying that if T theory is true, one ought to see effects E_1–E_k, a computational modeler must show that effects E_1–E_k occur in simulations of theory T. Such a process is a form of theoretical proof unavailable to natural language theories unless they are translated into computational models. Moreover, Farrell and Lewandowski note that our higher-level reasoning skills are not typically up for the task of reasoning about how combinations of simple, dynamic processes work (see also Hintzman, 1990). Thus, if one does get a formal representation of a combination of such processes to work, one has at least a viable explanation, if not a valid one.

An example of the aforementioned value of computational modeling can be found in the debates among self-regulation theorists spanning decades. That is, critics of motivational control theories (e.g., Bandura, 1986; Bandura & Locke, 2003) argued that such theories could not explain discrepancy production (i.e., raising one's goal following previously successful performance) because the theory's core process is one of discrepancy reduction (i.e., applying resources until a goal level is reached). In contrast, they claimed, social cognitive theory could explain discrepancy production. Yet, Scherbaum and Vancouver (2010) noted that social cognitive theory only claimed that discrepancies were produced. Other than naming a construct involved (i.e., self-efficacy), it did not explain the *mechanism* that produced the discrepancies. Meanwhile, Scherbaum and Vancouver built a computational model that produced discrepancies using a combination of a few basic discrepancy-reducing structures (i.e., consistent with motivational control theory, Powers, 1973).

In another case, Vancouver et al. (2020) translated goal theory, which is a presumably robust, practical theory of motivation (Locke, 1997; Locke & Latham, 2002), from a static, semi-formal model (i.e., a path diagram with their statistical equations at least implied) to (1) a static computational model and (2) a dynamic computational model. In the process of translating the model from static to dynamic, several presumably reasonable process functions at a key point in the dynamics (i.e., at the return in a feedback loop) were tested to see which might be viable, given goal theory was ambiguous on this point. Only one survived. That is, the process of building the models revealed gaps in specification, and simulations of possible formulations allowed the modelers to test what could and could not account for the phenomena the theory presumably explained. Indeed, modeling platforms, particularly ones devoted to theory development, can help substantially in this process, and the chapters in the second half of this book discuss these processes extensively.

The Benefit of Computational Models: Theory Integration

Several of the complaints about the state of I-O psychology and management science relate to theory proliferation. Yet, science is about simplifying nature to facilitate understanding, as well as to help predict and navigate nature better. Toward that end, theory proliferation has arguably become an impediment rather than a facilitator of that process. One way to reduce theory proliferation is theory integration. Fortunately, computational models can often facilitate theoretical integration by reducing theories to sets of mathematical equations and key, recurring structures. For example, Vancouver et al. (2010b) integrated classic theories of behavior (i.e., self-regulation theories) and classic theories of choice (i.e., subjective expected utility/expectancy theories) using one simple information processing structure. To be sure, the structure was repeated and with different inputs to address the different purposes each individual structure had (e.g., multiple instances of one structure within one model). Subsequently, Vancouver and Purl (2017) used a variant of the aforementioned model to integrate the forethought processes key to Bandura's (1986) social cognitive theory with the feedback processes central to control theory models of self-regulation.

In another case of high-level integration, Vancouver et al. (2014) showed that the math used to represent the simple structure noted earlier was identical to the math used to represent another psychological process: supervised learning. Supervised learning occurs when one can compare an expected state with an observed state, known as the supervisor signal, to correct the internal, mental model used to create the expected state. Vancouver et al. (2014) added these supervised learning components to the 2010 model and thus provided an integrated model of goal pursuit and learning using only a single, simple, but repeated architecture. Indeed, this simple computational architecture matches, mathematically, connectionist (i.e., neural network) architecture (Rumelhart & McClelland, 1986). Meanwhile, specific neural network models have been applied to lots of specific learning problems. For example, Van Rooy et al.

(2003) showed that the outgroup homogeneity effect (i.e., the impression that members of outgroups have less within-group variation than one's in-group) could be modeled as a function of learning and the differences in exposure to members of the respective groups. Higher-order processes (e.g., attributions) were not needed to explain the phenomenon. To be sure, the connectionist architecture likely does not account for all that is involved in human learning, but it does a good job of emulating how we do it most of the time. Indeed, it often learns better (or at least faster) than humans, such that it is now the basis of machine learning and artificial practical applications.

Continuing the process described earlier, Ballard et al. (2016) integrated a cognitive theory of dynamic decision making into the original goal pursuit model of Vancouver et al. (2010b). This was followed by Ballard et al. (2017) showing how another variant of the structure could account for how humans avoid unwanted states (i.e., avoidance goals) as well as desired states (i.e., approach goals). Finally, Ballard et al. (2018) generalized the original model to include the elements needed to explain how humans handled goals with different deadlines. To be sure, despite the repetition of the simple structure, most of the models build look very complicated. Indeed, it would be foolish to expect anyone to look at the structures and equations and be able to explain what they could do without a detailed explanation or, better yet, interacting with the computational model within the software platform. That is, the core of a theory can be parsimonious, but the scope of the theory comprehensive. This paradox arises because the results of the interaction of the model parts (i.e., subsystems), even if simple, are beyond our imaginations.

The Benefit of Computational Models: Experimentation and Exploration

We noted earlier that computational models provide a proof of concept via a working model that produces the behavior one is attempting to explain with a theory. We also noted that to get to a working model, one might need to explore alternative functions or processes to find one that works. This exploration involves experimenting with the model (Davis et al., 2007). Another form of exploration and experimentation with a model can happen after a working model is produced. It often involves determining which subprocesses and parameters are key. This *sensitivity analysis* involves running simulations with various parameter values to assess the role the parameters play in the behavior of the model. Some might substantially affect the model behavior, and some might not. When the parameters can be linked to construct measures or manipulations, the results of sensitivity analyses can be used for developing hypotheses for empirical studies, including null hypotheses where variation in the parameter is found to have little effect on model behavior.

Beyond parameters, subprocesses can be tested by turning them on or off across simulations of the model to reveal whether they are key or not to some phenomenon. For instance, Vancouver et al. (2016) assessed several possible explanations for positively skewed distributions of performance using a dynamic computational model. The possible explanations were added as they built the model, stopping when they were able to represent a process that reproduced a referent data set (i.e., an empirical observation of the phenomenon). In contrast, Vancouver et al. (2020) removed components that were shown to be unnecessary when a static model was made dynamic. In this way, the addition of a dynamical perspective increased the parsimony of the theory without needing additional empirical work. Indeed, where exploration and experimentation are especially useful is when empirical data is difficult to acquire. This issue is more common with macro-level models, which Davis et al. (2007) describe to great effect.

Meanwhile, when empirical model testing is available, computational modeling can be very valuable in revealing the empirical patterns to look for when challenging or pitting theories. For example, Vancouver and Scherbaum (2008) used a computational model to first explicate the difference between self-regulating behavior and self-regulating perceptions, including how the models make different predictions, given an experimental protocol. They then implemented the protocol to see what the data revealed. Similarly, Vancouver et al. (2010a) used a computational model to illustrate how the typical empirical protocols used to evaluate a dominant theory of proactive socialization were not up to the task because researchers did not know that they needed to take the rates of processes into account when collecting data. Moreover, they showed how even time-series data might not be diagnostic because an alternative theory, which Vancouver et al. (2010a) also modeled, produced the same pattern of results. However, they subsequently used the models to illustrate what empirical protocols might differentiate the competing theories. These examples show how computational modeling can be used to critique and as well as guide empirical protocols.

The Benefit of Computational Models: Falsification and Improved Prediction

According to Popper (1963, 2014), what makes a model scientific is that it can be proven wrong (i.e., falsifiability). Of course, all models, whether verbal or computational, are wrong (Sterman, 2002), so usefulness is a better criterion. Still, unlike verbal theories, computational models must be internally consistent or simulations of them would not run, or they would produce behavior inconsistent with the phenomenon they presumably explain. Thus, the need to produce a working model provides an initial opportunity to falsify a theory (as well as undermine its usefulness). In addition, computational models are more transparent than verbal theories (Adner et al., 2009; Weinhardt & Vancouver, 2012). Thus, whether mathematical or propositional, the explicit functions provide an opportunity to question the scientific plausibility of a function or process (i.e., set of functions). Moreover, via exploring alternative functions or processes, one might find alternatives that also work, alternatives that might (or might not) change the behavior or predictions of the model as revealed in simulations of the alternatives. Via these differences, one might be able to develop protocols to see which or when (i.e., under what conditions) models make better predictions. We described this process earlier in the section on exploration and experimentation.

Yet another benefit of computational models is improved prediction (Adner et al., 2009). This not only increases the ability to falsify, but it also increases the usefulness and thus reduces the likelihood the model will be discarded. To explain, Meehl (1990) noted that where many constructs tend to be interconnected, it is relatively easy to find support for theories. As such, demonstrating a relationship between constructs is often a straightforward task, and thus theories that simply predict some relationships are likely to be confirmed. However, computational models improve our ability to make predictions (i.e., ones that move beyond predictions of mere association). Computational models can provide predictions not only of the relationships and their direction but also of the levels of variables over time and the shape of relationship between variables and/or over time. To be sure, this feature depends on the quality of inputs when constructing a model (e.g., are the rates and delays of processes well-known?) as well as the quality of inputs into simulations or model fitting exercises. For instance, computational models of emotion intensity onset and decay are hampered by imprecise knowledge of such rates (Hudlicka, 2023). Still, a computational model will be much better positioned to render precise predictions when the measurement models allow it.

Yet, even when measurement and design are lacking, computational models can improve prediction via competitive prediction. In a field that boasts a wealth of theoretical and empirical contributions, it frequently lacks competition among theories. That is, merely identifying a theory that accounts for a particular phenomenon in isolation does not align with the fundamental principles of scientific inquiry (Gigerenzer & Brighton, 2009). Instead, the pursuit of science requires the identification of theories that surpass existing ones in accounting for a particular phenomenon or expanding the scope of what has been explained (Popper, 1959, 2002). Therefore, it is crucial to evaluate theories by contrasting them with alternative theories to determine their validity and comprehensiveness in explaining a phenomenon.

To realize the benefits of falsifiable models and improved prediction, one needs to look no further than our earlier discussion of the many competing models of multiple-goal pursuit. In less than 10 years, the original multiple-goal pursuit model (Vancouver et al., 2010) has lost in competition with modifications of the model or simply expanded its predictive power via integration with other models across six papers to better account for human behavior. This rapid progress would not likely have occurred if not for the theories being computational. Meanwhile, computational models clearly eliminated some explanations for phenomena. For example, on the way to providing a sound person-environmental interactional explanation for the positive skew typically found in the distribution of performance across individuals, Vancouver et al. (2016) eliminated several possible person-centered explanations.

Meanwhile, pitting theories against each other empirically assumes that the explanations are not both true. For example, Ballard et al. (2018) pitted two explanations for increasing motivation as a deadline approached. They found evidence supporting both explanations. More important, they were able to represent the two explanations within a single, relatively simple (i.e., conceptually parsimonious) computational model of goal choice and striving. As noted earlier, this redundancy of processes is commonly found in biology, but it is rarely recognized in verbal theories in psychology.

Even in cases where alternative descriptions of human behavior presumably predict different effects, computational modeling can provide the path to unification. For example, Vancouver and Purl (2017) used their computational model to explain when one might see a positive, negative, null, and curvilinear relationship between two constructs as a function of what appeared to be two competing theories. The model revealed the logic, and contingencies, where the various empirical models might arise while also reconciling the theories. More recently, Shoss and Vancouver (2023) presented a model that included three countervailing processes that lead to opposing hypotheses and yet likely work in concert. Such configurations of apparently incongruous parts operating harmoniously with a single system are difficult to describe verbally or believed workable without demonstration. Computational modeling can address those limitations to conventional theorizing.

The Benefits of Computational Modeling: Final Thoughts

To summarize the earlier arguments, computational modeling is needed to realize the promise of science, which is to simplify nature. Ironically, computational models are often seen as so complex as to be of little scientific value (i.e., the Bonini paradox). However, upon further consideration, the ability of computational models to integrate and generalize via repetitions of simple agents, whether subprocesses or actors with a small rule set, offers the desired advantage of conceptual parsimony. To be sure, sometimes the modeling reveals where differentiation is needed (i.e., self-regulated behavior ≠ self-regulated perception; Vancouver & Scherbaum, 2008) or where systems have functional redundancy across two or more subprocesses (e.g., Ballard et al., 2018), but that too is part of science. However, more often they reveal conceptual overlap via the abstract math used to build the model.

One Nobel Prize-winning chemist described the desired state of formal modeling as where "the models become modules in the theoretical erector set, shuttled into any problem as a first (not last) recourse" (Hoffmann, 2003, quoted in Rodgers, 2010, p. 9). The sentiment expressed by this Nobel Prize-winning scientist in a mature field of inquiry is the goal we have for I-O psychology and management science. Like many sciences, psychology and management science focuses on and tries to understand complex systems interacting over time, nested in a cascade of higher-order systems, and utilizing a cascading set of subsystems. Computational models are built to evaluate explanations of some small part of that network of systems over some presumably relevant timescale, depending on the problem at hand. Eventually, the templates of the models can be used for new parts or problems, or the models themselves can be combined to address more complex parts of some whole. Moreover, because the models are computational, we do not need to rely on human's limited (but still very impressive) computational prowess because we have built the supporting tools needed to utilize the understandings developed. Of course, to get there we need to train ourselves and future scholars in our field to understand the tools and what they can do for us.

Unfortunately, for the uninitiated, little about computational modeling seems simple. Moreover, no one person will likely be able to understand all modeling approaches, much less all models. As we attempted to articulate earlier, the value of computational modeling is not merely in its products but also in its process. No paper on computational modeling should be expected to realize any more than one or two of the values we described earlier, and some papers might not even explicitly realize even one. That is, once one learns to computationally model, one might merely use it to check one's thinking about how a theory or process would play out and thus whether the paradigm one wants to implement to challenge some explanation is up to the task but without publishing the model. Indeed, one might not want to share with the world their own skepticism about their reasoning prowess, even though most modelers will tell them such humility is well-placed (Sterman, 2002). That said, in the next section, we review key examples of I-O psychologists laying bare their skepticism regarding their reasoning prowess.

Brief History of Computational Modeling in I-O Psychology

The history of computational modeling is a relatively short one, especially in I-O psychology. In general, computational modeling only emerged in psychology and management science in the second half of the 20th century (Simon, 1969). For example, Forrester (1961) pioneered system dynamics modeling, which was used to address macro-organizational issues like managerial decision making (e.g., Sterman, 1989) and organizational learning (e.g., Senge, 1990). Using a more general modeling approach, March (1991) considered some dynamics related to organizational learning. Lomi and Larsen (1996) examined the dynamics of organizational populations. Hanisch et al. (1996) modeled the macro (i.e., organizational) implications of different possible withdrawal processes. In that same year, Martell et al. (1996) published a paper in *American Psychologist* describing simulations of a model where small but accumulating adverse effects of bias on promotions for women could possibly account for the absence of women in the highest offices of organizations. Also around this time, papers related to dynamic decision making and feedback processes at the individual level were being published in what was the precursor to *Organizational Behavior and Human Decision Processes* (e.g., Gibson et al., 1997; Sterman, 1989). On a more basic, psychological level, John R. Anderson (1983) developed a propositional computational model focused on human learning and behavior called ACT theory that informed theories of learning and training. At an even more granular level, a great deal of attention was paid to neural network models as a way of understanding learning (e.g., McClelland & Rumelhart, 1981; Anderson, 1996). As noted earlier, those models are now the backbone of machine learning algorithms.

Seeing the promising but relatively low use of computational modeling in the field at the turn of the century, Ilgen and Hulin (2000) published an edited book about computational modeling of behavior in organizations. The book included several chapters from well-known researchers presenting relatively simple computational models on many topics of potential interest to I-O psychologists, like the effects of faking on personality measures in selection (Zickar, 2000), effects of pay-for-performance systems (Schwab & Olson, 2000), group decision making (Stasser, 2000), applications of a modeling system (i.e., Petri nets) on various problems (Coovert & Dorsey, 2000), cultural norm formation (Latané, 2000), organizational change (McPherson, 2000), organizational adaptation (Carley, 2000), and other implications of the withdrawal processes model (Hanisch, 2000). Unfortunately, the book had limited impact on the field. Part of the problem may have been that the learning curve seemed too steep and the benefit too obscure to put the time in to learn about computational modeling and what it might do for one's science and career.

In another attempt to promote computational modeling to the mainstream of organizational and management research, *Academy of Management Review* (AMR) published a special issue on computational modeling in 2007. Two important and highly cited articles focusing on the process of computational modeling were published in that edition (Harrison et al., 2007; Davis et al., 2007). Unfortunately, this did not lead to an increase in computational models in AMR largely because of a limited audience amenable to appreciating such models. Indeed, it seems a critical mass of scholars who can understand, appreciate, and develop computational models are needed to realize their potential and mature the discipline.

Still, macro computational models were being published in other well-respected journals in management like *Administrative Science Quarterly* and *Organization Science*. Yet, most were published in journals specializing in computational or mathematical modeling, like *System Dynamics Review*, which started in 1985, and *Computational and Mathematical Organizational Theory*, which started in 2004. Indeed, with the occasional exception in *OBHDP*, top I-O and management journals were not publishing micro- or meso-level models.

However, something changed in 2010. First, Vancouver et al. (2010a) published a computational model in the *Journal of Management*. In the paper, three related computational models were used to explain a paradox in the socialization literature by highlighting how the empirical paradigms used to evaluate the major theories of proactive socialization were not diagnostic, given the dynamics inherent in the theories. Two of the models were at the individual level, and one was a meso-level model representing an employee-supervisor dyad. Second, Vancouver et al. (2010b) built a computational model of an individual pursuing multiple goals and used it to explain an interesting but unexplained finding in an earlier empirical paper (Schmidt & DeShon, 2007). The model, though invented little new theory, addressed a large theoretical gap in the literature on work motivation. The model was built using combinations of a simple, single architecture (i.e., the negative feedback control system) that integrated and made dynamic several traditional theories of motivation (e.g., expectancy theory, goal theory). The paper also thoroughly explained the model-building and model-evaluating process. Finally, Dionne et al. (2010) published a computational model in *The Leadership Quarterly* that focused on leadership, shared mental models, and team performance. This meso-level model also provided a way to address some longstanding theoretical gaps, while employing a new, dynamic twist on existing theory on leadership and teams.

Since 2010, many more computational modeling papers have been published in I-O's top journals. Indeed, the first set of chapters in this book review computational modeling work in decision making as it applies to organizations (Cooney et al., Chapter 2, this volume), diversity and inclusion (Samuelson et al. Chapter 3, this volume), training and socialization (Hardy, Chapter 4, this volume), leadership in teams (Zhou, Chapter 5, this volume), as well as teams and groups more generally (Kennedy & McComb, Chapter 6, this volume). In some cases, like the work on decision making, most of the modeling has been around for a while but not applied to I-O psychology problems. In other cases, like the work on diversity and inclusion, training, leadership, and teams, much of the modeling is more recent (e.g., past 10 years). Still, this work has both computational and theoretical roots that have been long established. Perhaps most importantly, all the chapters highlight the vast amount of work that needs to be done, given the nascent nature of the computational approach within the field. Fortunately, top I-O journals have now opened their doors to computational modeling papers, making the presumably steep learning curve worth climbing not only for those interested in applying the tool to their projects but also for those reviewing and reading those journals. Still, to get a critical mass of modelers and a sophisticated audience, training is needed.

Computational Modeling Training: Levels of Schooling

As noted earlier, computational modeling has great potential to facilitate our understanding of organizational phenomena. However, the extent to which these potentials can be realized depends on whether there exist "well-trained craft persons" to leverage the strength of the "tool." Apprentices and journey-level workers, in the form of sophisticated readers, reviewers, and researchers, through masters, in the form of model builders, need to be trained before realizing the trade that is computational modeling. Indeed, there may be heterogeneous training goals contingent upon one's specialized role within the trade. In the following sections, we summarize what is expected of readers and reviewers, researchers, and modelers to support the development of computational modeling in applied psychology and management.

Apprentices: Sophisticated Readers and Reviewers of Computational Modeling Papers

The first type of craftsperson are readers and reviewers of computational modeling papers. It might seem odd to include reviewers, who are supposed to be experts, with readers, who are not expected to have such qualifications. However, reviewers are often chosen for their expertise in the topic but not necessarily the method or analysis used. Still, we want any reviewer, like most readers, to be able to (1) know the conventional vocabulary of computational modeling; (2) understand what goes into creating, validating, and evaluating the model; and (3) understand how the model can help in understanding, studying, predicting, or controlling the phenomenon of interest.

First and foremost, computational modeling involves a systematic repertoire of vocabulary to ensure conceptual ideas and theoretical logic are communicated formally and precisely. Some vocabularies speak to the nature of constructs. For example, *dynamic variables* (also called *level* or *stock variables*) refer to variables that exhibit inertia and only change values when external forces are applied (Forrester, 1968; Vancouver & Weinhardt, 2012; Wang et al., 2017). This feature is extremely important in the modeling of time but is difficult to illustrate precisely in verbal theories. Other vocabularies are borrowed from related domains but carry particular importance in computational modeling, such as *exogenous* and *endogenous variables* that delimit the core mechanisms included in the model. That is, endogenous variables are affected by other variables in the model, whereas exogenous variables are not.

Also, like informal theories, computational modeling uses specialized vocabularies and graphics to depict relationships among variables. The central difference is that in addition to verbal/graphic depictions, variable relationships are also represented in mathematical forms. However, the mathematical functions can be more complex than the four simple operations (addition, subtraction, multiplication, and division), and they are conventionally presented in a way that variables involved in a function all point at the construct (i.e., the result of the function). Thus, when depicting a computational model, one should not expect to see an arrow point at an arrow to depict a multiplicative or moderating function. Instead, consult the actual function. Also, the functions can reveal what variables are dynamic, given that integration and/or differentiation with respect to time is frequently used to capture dynamic processes. Further, there are specialized vocabularies in computational modeling for model evaluation (Taber & Timpone, 1996), including *internal validity* (i.e., how well the model successfully translates the corresponding verbal theory), *outcome validity* (i.e., model fit), *process validity* (i.e., the correspondence between the model's mechanisms and the actual processes being modeled), and *sensitivity analysis* (i.e., how changes in parameters values affect model predictions).

Arguably, the conventional vocabulary of computational modeling can be an entry barrier and requires some specialized efforts invested, but such efforts are worth it not only for those seeking to convey precise, unambiguous, and comprehensive theories, but for those seeking to benefit from, build upon, or question such models. As Vancouver and Weinhardt (2012: 605) stated, "If mathematics is the language of science, and computational models are an expression of that language, scientists should be at least familiar with it," and "the best way to do this is to use the language."

After learning the conventions (e.g., vocabulary), the second objective is to learn the processes. Computational modeling involves systematic procedures in terms of building and evaluating the model. In this regard, Vancouver and Weinhardt (2012) offer a comprehensive summary starting from problem identification and ending with model evaluation. Weinhardt (Chapter 9, this volume) offers a comprehensive evaluation framework that outlines the work one must do to evaluate models. Here we offer a quick overview.

To begin, the problem of interest is mostly based on an existing conceptual framework that depicts a dynamic process (Busemeyer & Diederich, 2010). Further, the problem is usually defined in a narrow but precise way. This reflects the likely complexities that will arise once one begins to represent the processes needed and clarifies how the computational model contributes to theory refinement. Problem selection is followed by system definition, typically including determining the units of analysis, the time scale and boundary of the dynamic process, and the variables involved. These aspects can be partly determined by the theoretical framework adopted. However, some theories (especially verbal theories) can have much broader scopes than the bounds of typical computational models, requiring decisions on which part(s) of the theories to be modeled. Then the model is built based on the variables and parameters identified. Specifically, relationships between variables are expressed both graphically and mathematically, and multiple software platforms are available to facilitate the process (e.g., C++, Python, MATLAB, Vensim, R, etc.).

The model is then evaluated via various approaches and criteria. For example, simulations can be run to see if variables exhibit reasonable dynamic patterns (e.g., reaching an equilibrium level as opposed to increasing or decreasing infinitely, unless that is what is observed [e.g., wealth accumulation]). The model prediction can be tested against empirical data qualitatively (e.g., whether the empirical data exhibits the patterns predicted by the model) or quantitatively (e.g., the accuracy or fit of the model after parameter estimation). The model can also be compared to alternative models to identify the relevance of different theoretical mechanisms. In this book, we cover both qualitative (Weinhardt) and quantitative (Ballard et al.) model evaluation, but as is always the case with scholarly work, the evaluation depends on the purpose and the purpose depends on where value is likely to be added to the existing literature. That said, in general, a formal rendering of theory adds value when so little formal modeling exists.

Third, besides the terminology and modeling process, it is also important to understand what the model does to illustrate the phenomenon of interest. Generally, as a particular type of formal theory, computational modeling has unique advantages in terms of theorizing about and representing complex systems. This can be illustrated by (1) comparing formal theories to informal theories and (2) comparing computational modeling to analytic formal theories (Adner et al., 2009). For example, compared with informal theories, formal theories enable theoretical clarity and a deeper understanding by providing descriptions of process details of complex dynamic phenomena. That is, because the set of processes needed to produce the phenomena must be specified at least somewhat, computational models have less "handwaving" (i.e., processes left unexplained) than informal theories. Of course, computational models will often have to live with black boxes and simplifying assumptions, even when attempting to illuminate a black box or test an assumption (e.g., any information processing theory must contend with our vague understanding of how information is represented in the mind).

Meanwhile, when compared with analytic formal theories, an advantage of computational modeling is that they can handle complex dynamic processes with causal loops and nonlinearities. These features of the phenomena that I-O psychologists study are often intractable using pure analytic approaches (Wang et al., 2016). Thus, to understand what a computational model does, readers and reviews should pay special attention to the unique and incremental theoretical contribution of computational modeling above and beyond informal theories and analytic formal theories. Relatedly, the processes (the mechanisms) specified in the model should be carefully examined against the scientific plausibility of the processes depicted, which sheds light on what the model does, whether the model has fidelity, as well as how the model refines existing theories.

Of course, a more sophisticated reader and reviewer of papers describing computational models will have experience working with such models. This can include working with existing models to find ways to validate or challenge them; however, to truly understand models one should build them. This book will help you do that. Starting with some simple examples is necessary to build the skills needed to start creating your own models. Still, most of the work of model building is likely more about extending the models of others via generalizing existing models or applying existing architectures to new problems. Eventually, one may feel existing architectures are insufficient for tackling certain kinds of problems and feel inspired to create one's own architecture, but it is more likely that one will use simple functions and modules to represent somewhat standard processes or rules that when cobbled together in a coherent formal description of a process will produce a value-added contribution to the field.

For example, I-O psychology and management are replete with process theories of domain-relevant phenomena. A key mission of "crafts persons" is to vet these process theories computationally. Such vetting requires (1) demonstrating whether core processes of the model can explain the phenomenon of interest and (2) providing core structures that can be expanded for evaluating interventions and boundary conditions or extensions (i.e., moderators). To find where extensions are needed, one need only consult the discussion sections of published computational models. That is, like empirical papers whose discussion sections talk about needed future theory and research, modeling papers talk about research that might challenge the model and the extensions needed to further the scope of what the model can handle. Of course, researchers are free to identify other substantial theoretical limitations or extensions. For example, presumed moderators should not suggest the bounds of a theory or model, but the territory into which the model can expand. Meanwhile, no modeling endeavor can handle all that needs to be represented just as no empirical paper should be considered the final say in some empirical question. Modelers need to provide ideas regarding where a model needs to go as well as what research might challenge it. They, and the reviewers of the papers, need to be okay with leaving this work to future research (e.g., Zhou et al., 2019). At the same time, readers and reviewers need to retain a skeptical attitude. An unsophisticated readership or reviewer may be too easily impressed by a computational model because it is beyond their comprehension. We see this happen too frequently with sophisticated statistical techniques, especially when new (e.g., interpreting causal conclusions drawn from structural equation models as internally valid). We do not want readers and reviewers rejecting a computational model due to myths regarding computational models (i.e., that they are more complex than the phenomena they seek to explain), but we do not want them acquiescing to a computational model due to some blind faith in the accuracy and integrity of the process. It is just a tool like any other.

Journey-Level: Researchers Who Can Design Studies to Test Models Empirically or Via Simulations

The second type of craft persons are researchers who can design studies to test computational models empirically or via simulations. This is needed because science is a system just like the phenomena scientists study. Just as computational models often depict iterative or circular processes, so too would a good model of the scientific process. Thus, the linear process described by Vancouver and Weinhardt (2012) is merely an arc in a scientific problem-solving process. Few vetted models should be considered settled models. Imperfection should not keep a model from being published, and a published model should not be the last word. Indeed, one reason for a large set of sophisticated readers-to-builders is to make sure the self-correcting nature of science can operate. In between readers and builders are those who vet computational models derived from process theories. Here again, vocabulary can be daunting because of the negative transfer interference with convention and the use of qualitative terms when clean distinctions are hard to come by.

For example, there exists, and we have chapters devoted to quantitative (i.e., parameterizing) and qualitative (i.e., pattern matching) fitting methods for model evaluation (Busemeyer & Diederich, 2010). Quantitative fitting involves estimating the free parameters in the model. In line with the common procedures in statistical modeling, parameterizing for computational models is usually done via minimizing some type of loss function (e.g., least squares estimation) or maximizing some type of likelihood function (e.g., maximum likelihood estimation). Notably, since each unit (e.g., individual) tends to exhibit a unique dynamic pattern, parameter estimation may be repeated for each individual, though Bayesian methods are now commonly employed to handle this multilevel analysis issue (see Ballard et al., Chapter 10, this volume).

Parameters obtained from the optimization algorithm can be examined directly (e.g., the distribution of each parameter, the covariation between different parameters). However, a more important function of parameter estimation is to provide a model fit index capturing the extent to which the model fits the observed data (Vancouver & Weinhardt, 2012). Specifically, by freely estimating some parameters, the optimization algorithm can find the best fit between model-expected outcomes and empirically observed outcomes. A fit significantly better than chance suggests that the model is effective (e.g., Vancouver et al., 2005), especially if the modeling fitting algorithm can *recover* the parameters used when data is generated from the model (see Ballard et al., Chapter 10, this volume). That is, best practice includes fitting data generated from a model that represents the nature of the measures, the sample size, and the design elements *before* attempting to fit observed data. This will show if the model fitting algorithm will work reliably given the model. This is most important if using the model for application (i.e., for obtaining parameter values for a unit to be used for understanding or prediction regarding that unit). When assessing the validity of the model, knowledge of the reliability of parameter estimates is useful to prevent over- and under-interpretation of the results of model fitting. It can also be useful for assessing the diagnostic power of an empirical design for assessing parameters. Typically, the data generated from a protocol might help allow for the reliable estimate of one or two parameters, but it may not do a good job allowing other parameters to be estimated because some subprocess may or may not be involved in the protocol (e.g., Ballard et al., 2018). In general, typical experimental and passive observational designs, even if longitudinal, will struggle to produce reliable estimates if very many processes are involved. For example, Vancouver et al. (2010b) illustrated how multiple parameters could explain the same individual differences in goal-striving behavior over time.

Another stringent test is to pit the proposed model against alternative models (not limited to a null or saturated model) to demonstrate its superiority. One challenge of this approach is that computational modeling has not been widely adopted in the field of organizational science, so there are hardly existing models to be compared with. Nevertheless, researchers have sought to develop alternative models on their own. For example, Zhou et al. (2019) compared their proposed model to several alternative models derived from different theoretical perspectives. Such comparisons are valuable in deepening our understanding of core theoretical mechanisms behind observed phenomena. They, like the model fitting studies described earlier, are likely not the last word on a model, given the theoretically infinite number of alternative models and the inability of most data collection enterprises to constrain the possibility of fitting models.

As noted, quantitative model fitting is a sophisticated and often difficult enterprise, given the qualities of the data needed, especially if the model has several unknown (i.e., free parameters) that must be estimated. Fortunately, there is much one can do under the qualitative model fitting moniker to evaluate a computational model. Indeed, qualitative approaches to model fitting is a broad category with some elements looking quite close to what one would consider a quantitative procedure in typical research projects. For example, in our field, a hypothesis derived from theory and/or past research merely describes a pattern (i.e., some effect exists and is of a certain sign, as in the correlation should be positive). That is, the question asked is if one or more patterns found in the data match what a theory would expect the pattern[s] to look like. Toward that end, some statistical model is used based on the pattern one wants to examine, and then the resulting statistics are compared to the predicted patterns (i.e., the hypothesis). Thus, qualitative model fitting often involves quantitatively estimated effects.

Meanwhile, which comes first, the data or the model, determines mostly only whether they call the process prediction or postdiction (Taber & Timpone, 1996; Vancouver et al., 2010b). Given that theories are derived to explain observations (i.e., inductively or abductively), and computational models are representations of theories, postdiction is often a fine choice, especially if there is a lot of research on the topic. Indeed, model builders often need to confirm that their computation model can reproduce phenomena and effects found in the existing literature (Vancouver et al., 2010b; Vancouver & Weinhardt, 2012; Vancouver et al., 2020). At the journey level, we would expect to see researchers applying existing models to existing data sets to assess the generalizability or applicability of the model to phenomena. This typically requires tweaking little in the model of the unit, which is typically what the model is about, but something in the model of the environment or object (e.g., task) with which the unit is interacting. For example, Vancouver et al. (2014) assessed a model against a dataset described by DeShon and Rench (2009) after tweaking it to represent the DeShon and Rench protocol. The model was built by Vancouver et al. (2010b) and assessed originally against a dataset presented by Schmidt and DeShon (2007). Such testing of the generalizability or applicability of models should help reduce theoretical clutter. Indeed, as more models appear in the literature, the postdiction exercise of matching an existing model with an existing study (actual data from the study not required) could be an experiential learning opportunity for any modeling trainee.

Besides serving as a replication to verify the fidelity or generalizability of the computational model, postdiction can also shed light on the underlying processes through which certain phenomena and effects occur. This can complement informal theories, especially when informal theories fail to recognize dynamic factors at play and cannot account for the phenomenon. In this regard, Vancouver et al. (2010b) offered an example where a poorly understood phenomenon discovered in the existing literature was not only reproduced but also explained elaboratively based on the dynamic pattern generated by their model.

Meanwhile, prediction is more likely to be used when the existing data may not be up to the task of challenging a model. This can happen because the model produces some surprising pattern no one thought to look for, but more likely it is because computational models produce data not typically considered. For example, computational models can produce trajectories with nonlinearities (e.g., bifurcations, discontinuities); something few verbal theories would dare propose. And though qualitative, one can use quantitative methods for pattern matching. For example, Vancouver et al. (2005) used interclass correlation coefficients (ICCs) to fit progress trajectories across time for each individual from a study created to assess a computational model of goal striving. They also matched trial performance with two versions of the model for each individual. Finally, they matched the results for each condition that emerged from the model with the data, averaged across individuals and trials, across the ABA design. The model data emerged from a single run of the data, but it could have been from multiple simulations where one or more parameters are pulled for each run by specified distributions. Either way, this latter type of fit could be considered postdiction, given the predicted effect is exactly what the model was built to explain. It just happened to be a new study was commissioned to acquire the moment-by-moment trajectories that the informal theory did not speak to.

Still, other times the predicted hypothesis is derived to challenge a parameter related to a mechanism added to fill a gap between the theory formalized into a model and thus not previously assessed, or that reveals itself as central to some mechanisms only heretofore described verbally. An example of this latter form can be found in Ballard et al.'s (2018) test of two mechanisms thought responsible for increasing motivation for a task as its deadline approaches. In this kind of case, the modeling can identify or confirm a diagnostic test of the explanation depicted in the theory/model. That is, the researcher might "ask" the model to confirm that some effect would arise were the explanation valid or the researcher might observe an effect by conducting sensitivity analyses that would be unlikely to arise from a different process and that had not yet been systematically observed. Science has a special fondness for confirming such predictions, but they are only as special as the uniqueness of the explanation for them. That is, prediction is not as critical as the absence, to the degree possible, of alternative explanations. Toward that end, the construction of explanatory models is needed. This is the highest level of craft person the field needs.

Masters Level: Builders of Computational Model

Most of the time academics and practitioners know what statistical analysis they need to use to examine some dataset. Still, it is very useful to be able to consult a statistics expert or two in one's department, whether that department is in a university, an organization, or a government agency. Likewise, a mature science will and should have computational modeling experts who specialize in one or more computational architectures, and who can help others build, extend, and evaluate computational models. Preferably, they are familiar with the verbal theories and existing empirical work surrounding some phenomenon, but they also may work with others who broaden that expertise. At the current time, these individuals tend to come from other disciplines or subdisciplines where computational modeling is more prevalent (e.g., cognitive psychology or organizational theory). This provides opportunities for cross-pollination and thus a broader integration of modeling platforms. However, such individuals are typically hired to further their own discipline's computational work, need to be trained in the substantive phenomena, or are wedded to the vocabulary of their subdiscipline. Also, there just are not enough of them.

Given that I-O psychology has always been one of the more quantitatively sophisticated subdisciplines of psychology, we would hope that the field can also begin to produce these model builders. Still, it is not the intent of this book to produce a master class of computational modelers. Rather, it is to illustrate the potential of computational modeling for our field and to begin the apprenticeship of a new set of scholars, some of whom will become master computational model builders. Indeed, we believe that the information one can gain from the chapters in this book will inspire you to pick up a new tool and that the "how to" chapters will reveal that the learning curve may not be as steep as originally presumed.

Indeed, building models will provide the best background for reading, reviewing, and assessing models. Building models related to a topic of interest will also provide a perspective not always appreciated by modelers naïve to the subject matter. For example, modelers from more sophisticated subdisciplines tend to focus on quantitative model fitting. As a result, they emphasize the value of limiting the number of free parameters. In contrast, someone trying to represent a process explanation of a phenomenon is likely to consider several sources of differences among units (e.g., individual differences) or unknowns regarding processes (e.g., rates of change) that might matter in important ways. Assessing the effects of these differences and or different values for unknown parameters in the model is likely a prudent and useful step. One will likely find, via sensitivity analysis, that some can be removed because they do not affect model behavior much. Still, others are needed to remain true to the phenomenon and mechanisms represented, even if they are more than someone wanting to fit the model to data would like. To be sure, parameter proliferation can easily get out of hand such that one does not have a very useful model in terms of understanding. The question a modeler must answer, which may be unknown at the beginning but often reveals itself in the building process, is what the purpose of the model is, at least for some presentation of it. The purpose, which will need to make some contribution to the literature, will also dictate what free parameters might be set to 1 (if a weight, i.e., multiplier) or 0 (if a bias, i.e., additive term), which might be set to some constant (e.g., 0.5, as in "shoulder shrug") at least temporarily, allowed to be free, or represented as multiple conditions for some key construct (i.e., high and low on x). How things will shake out in the end (or along the way) may often be hard to tell.

Indeed, model builders should be "lazy." That is, one can and should use existing architectures (e.g., neural networks, control system models) when building new models, if possible (Hoffmann, 2003). This will simplify the building process and increase the likelihood that the model will be a special case of a general theory (i.e., a paradigm). Similarly, a modeler can model existing verbal theories applied to the phenomena of interest. Modeling existing theories means that one likely has an empirical and conceptual foundation for the model. It also reduces theory proliferation and can possibly lead to theory trimming if, upon attempting to model a theory, the logic laid bare comes up wanting (see, e.g., Vancouver & Purl, 2017; Vancouver et al., 2020). If there are multiple theories existing within the domain, one might model more than one to provide a way to pit theories or demonstrate the viability of the mechanisms, whether opposing or complimentary, coexisting in a reasonable way (e.g., Ballard et al., 2018).

It is important to acknowledge that modeling is not easy. One must translate an explanation into a set of functions and get them to all work together well and in line with the conceptual space that the mathematical symbols and operations are supposed to be representing. One also must defend the translation decisions to an audience often unfamiliar with process explanations of phenomena. One likely must identify errors in coding when the model produces peculiar behavior or clearly track down the source of the behavior to understand and explain why the observed behavior of the model makes sense (i.e., that it is not a coding error, but it is a prediction of the model [but it is usually a coding error]). To accomplish these things with reasonable efficiency, one must know the modeling platform well. It takes time and practice to develop these skills, especially across multiple platforms or in very flexible platforms like R, Python, or MATLAB. Indeed, none of the editors of this book have achieved such a level of proficiency. However, it is a brave new world, and some readers out there will achieve such a level of proficiency. The science is depending on it.

Book Chapter Summaries

In total, this book reflects where we have been, where we currently are, and where we are going regarding computational modeling in organizational psychology. Still, where we go is up to you, the reader. We hope that this book begins a new scientific journey for you and for our field. We have broken down the chapters in this book across two broad themes. In the first section, "The Call for Computational Modeling in I-O Psychology," the chapters provide overviews of how computational modeling has been and could be applied to a range of organizational issues. In the second section, "Creating and Validating Computational Models," the chapters provide the reader with the how-to knowledge to develop, validate, publish, and review computational modeling papers. Next, we summarize each chapter.

The Call for Computational Modeling in I-O Psychology

Computational models of decision making have been a staple in the applied cognitive literature since the beginning of computational modeling (Simon, 1969). Indeed, the field of decision making, and its embrace of computational models, is a good roadmap for how we can move our science forward. Cooney, Kaplan, and Braun (Chapter 2) provide a review of computational models of decision making and their application to organizational decision making. The authors present a typology of computational models of decision making broken down by whether the model is static or dynamic, whether the model is descriptive or prescriptive, and whether the model is micro, meso, or macro. This typology will be instrumental for researchers seeking to learn which decision-making models might be useful for their research questions.

Samuelson, Lee, Wessel, and Grand (Chapter 3) overview how computational modeling has impacted and can further impact diversity and inclusion research. These models also have a long history in the relatively short history of computational models. For instance, one of the first agent-based models was Schelling's (1971) model of residential racial segregation. This model showed that even a slight preference for living by people who are similar to oneself can lead to severe segregation over time. Chapter 3 reviews past work on computational modeling regarding diversity and how these models have benefits ranging from understanding basic psychological and social processes such as those underlying stereotypes and stigma to more applied issues such as optimizing interventions to reduce bias. Finally, the authors outline how specific modeling approaches (e.g., agent-based, neural network models) can spur future computational research on diversity and inclusion.

Hardy (Chapter 4) reviews applications of computational modeling to the literatures on learning, training, and socialization. Like the fields of decision making and diversity, this is another topic area that has embraced computational modeling. For example, both March's (1991) learning model and Anderson's (1996) adaptive control of thought (ACT-R) model come from this literature and are two of the most influential computational models in psychology and management. After reviewing previous models and insights from models regarding learning, training, and socialization, Hardy outlines a future research agenda for computational models focused on adult learning, informal learning, learning interventions, and how models can be used to highlight the utility of training for organizations.

Zhou (Chapter 5) reviews computational models of leadership in teams. Leadership is a complex and dynamic process, which when mixed with the complexity and dynamics of teams necessitates computational modeling. Zhou reviews past computational work on topics such as the emergence of leadership structure and group member participation among other important leadership topics.

Zhou's review shows that leadership is well suited for computational modeling and that multiple different computational modeling architectures have been applied to understanding leadership. Finally, Zhou shows how computational modeling has implications for advancing empirical and theoretical research on leadership. Moreover, computational modeling of leadership has important practical applications such as testing how different leadership structures are likely to affect group performance.

Kennedy and McComb (Chapter 6) conducted a systematic review of simulation research on groups and teams from 1998 to 2018. Their review shows that there has been an increase in simulation studies on groups and teams during this period and that multiple different modeling architectures have been used to study teams and groups. For each simulation paper, they identify the focal variable, the modeling technique, the simulation approach, and the insights gained from the study. The review emphasizes the theoretical considerations and opportunities for pursuing the examination of complex and emergent phenomena, as well as the flexibility simulation offers for tackling different types of research questions. Overall, the review suggests that simulation is a burgeoning approach that has the potential to advance group and team research.

Creating and Validating Computational Models

Tang and Liu (Chapter 7) kick off the section on how to create and validate computational models with their chapter on agent-based modeling (ABM). ABM is a powerful computational modeling technique that can help researchers gain insights into the dynamics of interactions among agents in a complex system, such as organizations. ABM involves revealing how the interactions among a collection of heterogeneous and adaptive agents following a (typically) small set of rules lead to complex behavioral patterns. Thus, ABM allows researchers to study emergent phenomena that arise from the interactions among agents. Tang and Liu provide an overview of ABM, including its defining characteristics, strengths, and limitations. They then review articles published in premier organizational and psychological journals that used ABM to model organizational phenomena. Finally, the chapter provides a detailed walkthrough of building a simple ABM for the scenario of newcomers joining a team of seasoned organizational members. This example illustrates how ABM can be used to model the dynamics of organizational phenomena and build organizational theories.

Vancouver and Li (Chapter 8) introduce readers to system dynamics modeling, which is another computational modeling platform useful for a large set of computational modeling opportunities. Systems dynamics has been around for a long time (e.g., Forrester, 1968), and researchers in this field have developed a user-friendly software platform for rendering computational models. After a short history of system dynamics thinking, the chapter focuses on how to build a model in the platform as well as some simple structures responsible for common dynamics, like growth, learning, calibration, and goal striving. They also describe how these structures and other common issues can be applied to address typical theoretical issues and phenomena in I-O psychology.

Switching from how to build models to how to validate them, Weinhardt (Chapter 9) outlines a framework for evaluating computational models called the Model Evaluation Framework (MEF). MEF specifies that a well-justified and useful model has three criteria: (1) logical consistency, (2) accurate, crucial predictions, and (3) generalizability. This provides the reader with a guide on how to qualitatively evaluate computational models. The MEF specifies that the work of evaluation requires two roles: the modeler and the evaluator. The modeler's work is to specify and justify their model to the full extent of their ability. The evaluator's work is to evaluate the usefulness and validity of the model—a process that will likely require iterations with the modeling role to accomplish the necessary work. Weinhardt guides the reader not only on the step-by-step process of model evaluation but also guides the reader through the philosophical underpinnings of this model evaluation process.

Ballard, Palada, and Neal (Chapter 10) provide a tutorial on quantitatively fitting computational models to data. Fitting computational models to data can be important not only when testing the validity of a model but also when estimating free parameters that are needed to make predictions for the specific system(s) being simulated. That is, the fitting process allows researchers to quantify the degree of correspondence between the model and the observed data, as well as compare the fit of alternative models. The chapter provides a tutorial on the process of model fitting, using the multiple-goal pursuit model as an example. The tutorial covers the steps required to code a model, estimate its parameters, and assess its fit to the data. The tutorial is aimed at readers who have some basic familiarity with computational modeling, but who may have limited experience with the R programming language often used for quantitative fit and parameter estimation. By the end of the tutorial, readers will have gained practical experience in model fitting and be equipped to extend the approach to more complex research questions in their own work.

Finally, Neal, Ballard, and Palada (Chapter 11) provide practical insights and recommendations on how to publish and review computational models. As the practice of computational modeling is still in its infancy within our field, there is often a lack of clear guidance on how to write and assess papers that utilize this approach. The chapter first outlines the steps involved in creating a computational modeling paper and recommendations for how these papers can be published in top journals within the field. They then discuss the key issues that authors and reviewers need to consider when publishing and reviewing these types of papers, with special attention paid to the similarities and differences between traditional papers and computational modeling papers. By the end of this chapter, both authors and reviewers will have a better understanding of how to approach and evaluate computational modeling research in the organizational sciences.

Conclusion

Computational modeling is exploding within the general field of psychology. In 2007, the *Cambridge Handbook of Computational Psychology* included 26 chapters on computational modeling. In 2023, the new edition has 38 chapters, including one devoted to I-O psychology. Indeed, like the trajectory illustrated in our brief history of computational modeling in I-O psychology and management, the references in the new handbook include a smattering of classic computational work within each subdiscipline, though going back a bit (i.e., the 1950s or so). Still, many more references are to work done in the past 10 to 15 years not because the charge was to review recent computational modeling work, but because so much more work is being done. Some of this work is very relevant to I-O psychologists but likely largely unknown. For instance, Read et al. (2017) created a neural network model relating to the activation of three motivational systems and used it to recreate the psychometric structure of the Big Five personality dimension. Indeed, it likely reveals a whole new set of worlds with which few in I-O psychology are familiar.

Likewise, we think this book will surprise and, dare we say, delight readers regarding the value and opportunities computational modeling can provide the individual researcher, research teams, and the fields of I-O psychology and management. Computational modeling provides a way of thinking about phenomena in the field that is refreshing and much needed. The book describes this tool that supports the scientific enterprise to a level not seen since the introduction of statistical methods. Using the tool provides a way for one to have confidence in the internal consistency of one's mental models and other's verbal models of the way things are thought to work. If all goes well, the field might be able to say that it is as good at what it does as meteorologists!

References

Adner, R., Pólos, L., Ryall, M., & Sorenson, O. (2009). The case for formal theory. *Academy of Management Review*, *34*(2), 201–208.

Anderson, J. R. (1983). *The architecture of cognition*. Hillsdale, NJ: Lawrence Erlbaum Associates.

Anderson, J. R. (1996). ACT: A simple theory of complex cognition. *American Psychologist*, *51*(4), 355–365.

Ballard, T., Vancouver, J. B., & Neal, A. (2018). On the pursuit of multiple goals with different deadlines. *Journal of Applied Psychology*, *103*(11), 1242–1264.

Ballard, T., Yeo, G., Loft, S., Vancouver, J. B., & Neal, A. (2016). An integrative formal model of motivation and decision making: The MGPM*. *Journal of Applied Psychology*, *101*(9), 1240–1265.

Ballard, T., Yeo, G., Vancouver, J. B., & Neal, A. (2017). The dynamics of avoidance goal regulation. *Motivation and Emotion*, *41*(6), 698–707.

Bandura, A. (1986). *Social foundations of thought and action: A social cognitive theory*. Englewood Cliffs, NJ: Prentice-Hall.

Bandura, A., & Locke, E. A. (2003). Negative self-efficacy and goal effects revisited. *Journal of Applied Psychology*, *88*(1), 87–99.

Busemeyer, J., & Diederich, A. (2010). *Cognitive modeling*. Thousand Oaks, CA: Sage.

Carley, K. M. (2000). Organizational adaptation in volatile environments. In D. R. Ilgen & C. L. Hulin (Eds.), *Computational modeling of behavior in organizations: The third scientific discipline* (pp. 241–273). Washington, DC: American Psychological Association.

Coovert, M. D., & Dorsey, D. W. (2000). Computational modeling with Petri Nets: Solutions for individual and team systems. In D. R. Ilgen & C. L. Hulin (Eds.), *Computational modeling of behavior in organizations: The third scientific discipline* (pp. 163–188). Washington, DC: American Psychological Association.

Davis, J. P., Eisenhardt, K. M., & Bingham, C. B. (2007). Developing theory through simulation methods. *Academy of Management Review*, *32*(2), 480–499.

DeShon, R. P., & Rench, T. A. (2009). Clarifying the notion of self-regulation in organizational behavior. In G. P. Hodgkinson & J. K. Ford (Eds.), *International review of industrial and organizational psychology* (Vol. 24, pp. 217–248). West Sussex: Wiley-Blackwell.

Dionne, S. D., Sayama, H., Hao, C., & Bush, B. J. (2010). The role of leadership in shared mental model convergence and team performance improvement: An agent-based computational model. *The Leadership Quarterly, 21*(6), 1035–1049.

Farrell, S., & Lewandowsky, S. (2010). Computational models as aids to better reasoning in psychology. *Current Directions in Psychological Science*, *19*(5), 329–335.

Forrester, J. (1961). *Industrial dynamics*. Cambridge, MA: The MIT Press.

Forrester, J. (1968). *The principles of systems*. Cambridge, MA: Wright-Allen Press.

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.

Gigerenzer, G., & Brighton, H. (2009). Homo heuristics: Why biased minds make better inferences. *Topics in Cognitive Science*, *1*(1), 107–143.

Hanisch, K. A. (2000). The impact of organizational interventions on behaviors: An examination of models of withdrawal. In D. R. Ilgen & C. L. Hulin (Eds.), *Computational modeling of behavior in organizations: The third scientific discipline* (pp. 33–68). Washington, DC: American Psychological Association.

Hanisch, K. A., Hulin, C. L., & Seitz, S. T. (1996). Mathematical/computational modeling of organizational withdrawal processes: Benefits, methods, and results. *Research in Personnel and Human Resources Management*, *14*, 91–142.

Harrison, J. R., Lin, Z., Carroll, G. R., & Carley, K. M. (2007). Simulation modeling in organizational and management research. *Academy of Management Review*, *32*(4), 1229–1245.

Heidegger, M. (1927). *Being and time*. Southampton: Basil Blackwell.

Heidegger, M. (1996). *The principle of reason*. Bloomington, IN: Indiana University Press.

Hintzman, D. L. (1990). Human learning and memory: Connections and dissociations. *Annual Review of Psychology*, *41*(1), 109–139.

Hoffmann, R. (2003). Marginalia: Why buy that theory? *American Scientist*, *91*(1), 9–11.

Hudlicka, E. (2023). Computational models of emotion and cognition-emotion interaction. In R. Sun (Ed.), *The Cambridge handbook of computational cognitive sciences* (2nd ed., pp. 973–1036). Cambridge: Cambridge University Press.

Ilgen, D. R., & Hulin, C. L. (Eds.). (2000). *Computational modeling of behavior in organizations: The third scientific discipline*. Washington, DC: American Psychological Association.

Kozlowski, S. W., Chao, G. T., Grand, J. A., Braun, M. T., & Kuljanin, G. (2013). Advancing multilevel research design capturing the dynamics of emergence. *Organizational Research Methods*, *16*(4), 581–615.

Kuhn, T. S. (1962). Historical structure of scientific discovery: To the historian discovery is seldom a unit event attributable to some particular man, time, and place. *Science*, *136*(3518), 760–764.

Kuhn, T. S. (2012). *The structure of scientific revolutions*. Chicago, IL: The University of Chicago Press.

Latané, B. (2000). Pressures to uniformity and the evolution of cultural norms: Modeling dynamic social impact. In D. R. Ilgen & C. L. Hulin (Eds.), *Computational modeling of behavior in organizations: The third scientific discipline* (pp. 189–220). Washington, DC: American Psychological Association.

Lee, H. L., Padmanabhan, V., & Whang, S. (1997). Information distortion in a supply chain: The bullwhip effect. *Management Science*, *43*(4), 546–558.

Locke, E. (1997). The motivation to work: What we know. In M. Maehr & P. Pintrich (Eds.), *Advances in motivation and achievement* (Vol. 10, pp. 375–412). Greenwich, CT: JAI Press.

Locke, E. A., & Latham, G. P. (2002). Building a practically useful theory of goal setting and task motivation: A 35-year odyssey. *American Psychologist, 57*, 705–717.

Lomi, A., & Larsen, E. R. (1996). Interacting locally and evolving globally: A computational approach to the dynamics of organizational populations. *Academy of Management Journal*, *39*, 1287-1321.

Lomi, A., & Larsen, E. (2001). *Dynamics of organizations: Computational modeling and organization theories*. Menlo Park, CA: AAAI Press/The MIT Press.

March, J. G. (1991). Exploration and exploitation in organizational learning. *Organization Science*, *2*(1), 71–87.

Martell, R. F., Lane, D. M., & Emrich, C. (1996). Male-female differences: A computer simulation. *American Psychologist*, *51*(2), 157–158.

McClelland, J. L., & Rumelhart, D. E. (1981). An interactive activation model of context effects in letter perception: I. An account of basic findings. *Psychological Review*, *88*(5), 375–407.

McGrath, J. E. (1981). Dilemmatics: The study of research choices and dilemmas. *American Behavioral Scientist*, *25*(2), 179–210.

McPherson, J. M. (2000). Modeling change in fields of organizations: Some simulation results. In D. R. Ilgen & C. L. Hulin (Eds.), *Computational modeling of behavior in* *organizations: The third scientific discipline* (pp. 221–240). Washington, DC: American Psychological Association.

Meehl, P. E. (1990). Why summaries of research on psychological theories are often uninterpretable. *Psychological Reports*, *66*(1), 195–244.

Popper, K. R. (1959). The propensity interpretation of probability. *The British Journal for the Philosophy of Science*, *10*(37), 25–42.

Popper, K. R. (1963). Science as falsification. *Conjectures and Refutations*, *1*, 33–39.

Popper, K. R. (2002). *An unended quest: An intellectual autobiography*. New York, NY: Routledge.

Popper, K. R. (2014). *Conjectures and refutations: The growth of scientific knowledge*. New York, NY: Routledge.

Powers, W. T. (1973). *Behavior: The control of perception*. Chicago, IL: Aldine.

Read, S. J., Droutman, V., & Miller, L. C. (2017). Virtual personalities: A neural network model of the structure and dynamics of personality. In R. R. Vallacher, S. J. Read, & A. Nowak (Eds.), *Computational social psychology* (pp. 15–37). London: Routledge.

Rodgers, J. L. (2010). The epistemology of mathematical and statistical modeling: A quiet methodological revolution. *American Psychologist*, *65*(1), 1–12.

Rumelhart, D. E., McClelland, J. L., & The PDP Research Group. (1986). *Parallel distributed processing: Explorations in the microstructure of cognition*. Cambridge, MA: The MIT Press.

Schelling, T. C. (1971). Dynamic models of segregation. *Journal of Mathematical Sociology, 1*(2), 143–186.

Scherbaum, C. A., & Vancouver, J. B. (2010). If we produce discrepancies, then how? Testing a computational process model of positive goal revision. *Journal of Applied Social Psychology*, *40*(9), 2201–2231.

Schmidt, A. M., & DeShon, R. P. (2007). What to do? The effects of discrepancies, incentives, and time on dynamic goal prioritization. *Journal of Applied Psychology*, *92*(4), 928–941.

Schwab, D. P., & Olson, C. A. (2000). Simulating effects of pay for performance systems on pay-performance relationships. In D. R. Ilgen & C. L. Hulin (Eds.), *Computational modeling of behavior in organizations: The third scientific discipline* (pp. 114–134). Washington, DC: American Psychological Association.

Senge, P. M. (1990). *The art and practice of the learning organization*. New York, NY: Doubleday.

Shoss, M. K., & Vancouver, J. B. (2023). A dynamic, computational model of job insecurity and job performance. In A. Zhou (Chair), *Novel directions in job insecurity research on work and nonwork domains*. Symposium conducted at the Society for Industrial and Organizational Psychology, Boston, MA.

Simon, J. R. (1969). Reactions toward the source of stimulation. *Journal of Experimental Psychology*, *81*(1), 174–176.

Stasser, G. (2000). Information distribution, participation, and group decision: Explorations with the DISCUSS and SPEAK models. In D. R. Ilgen & C. L. Hulin (Eds.), *Computational modeling of behavior in organizations: The third scientific discipline* (pp. 135–161). Washington, DC: American Psychological Association.

Sterman, J. D. (1989). Misperceptions of feedback in dynamic decision making. *Organizational Behavior and Human Decision Processes*, *43*(3), 301–335.

Sterman, J. D. (2002). All models are wrong: Reflections on becoming a systems scientist. *System Dynamics Review: The Journal of the System Dynamics Society*, *18*(4), 501–531.

Sun, R. (2008). *The Cambridge handbook of computational psychology*. New York, NY: Cambridge University Press.

Taber, C. S., & Timpone, R. J. (1996). *Computational modeling*. Thousand Oaks, CA: Sage.

Vancouver, J. B., Li, X., Weinhardt, J. M., Steel, P., & Purl, J. D. (2016). Using a computational model to understand possible sources of skews in distributions of job performance. *Personnel Psychology*, *69*(4), 931–974.

Vancouver, J. B., & Purl, J. D. (2017). A computational model of self-efficacy's various effects on performance: Moving the debate forward. *Journal of Applied Psychology*, *102*(4), 599–616.

Vancouver, J. B., Putka, D. J., & Scherbaum, C. A. (2005). Testing a computational model of the goal-level effect: An example of a neglected methodology. *Organizational Research Methods*, *8*(1), 100–127.

Vancouver, J. B., & Scherbaum, C. A. (2008). Do we self-regulate actions or perceptions? A test of two computational models. *Computational and Mathematical Organization Theory*, *14*(1), 1–22.

Vancouver, J. B., Tamanini, K. B., & Yoder, R. J. (2010a). Using dynamic computational models to reconnect theory and research: Socialization by the proactive newcomer as example. *Journal of Management*, *36*(3), 764–793.

Vancouver, J. B., Wang, M., & Li, X. (2020). Translating informal theories into formal theories: The case of the dynamic computational model of the integrated model of work motivation. *Organizational Research Methods*, *23*(2), 238–274.

Vancouver, J. B., & Weinhardt, J. M. (2012). Modeling the mind and the milieu: Computational modeling for micro-level organizational researchers. *Organizational Research Methods*, *15*(4), 602–623.

Vancouver, J. B., Weinhardt, J. M., & Schmidt, A. M. (2010b). A formal, computational theory of multiple-goal pursuit: Integrating goal-choice and goal-striving processes. *Journal of Applied Psychology*, *95*(6), 985–1008.

Vancouver, J. B., Weinhardt, J. M., & Vigo, R. (2014). Change one can believe in: Adding learning to computational models of self-regulation. *Organizational Behavior and Human Decision Processes*, *124*(1), 56–74.

Van Rooy, D., Van Overwalle, F., Vanhoomissen, T., Labiouse, C., & French, R. (2003). A recurrent connectionist model of group biases. *Psychological Review*, *110*(3), 536–563.

Wang, M., Beal, D. J., Chan, D., Newman, D. A., Vancouver, J. B., & Vandenberg, R. J. (2017). Longitudinal research: A panel discussion on conceptual issues, research design, and statistical techniques. *Work, Aging and Retirement*, *3*(1), 1–24.

Wang, M., Zhou, L., & Zhang, Z. (2016). Dynamic modeling. *Annual Review of Organizational Psychology and Organizational Behavior*, *3*, 241–266.

Weinhardt, J. M., & Vancouver, J. B. (2012). Computational models and organizational psychology: Opportunities abound. *Organizational Psychology Review*, *2*(4), 267–292.

Zhou, L., Wang, M., & Vancouver, J. B. (2019). A formal model of leadership goal striving: Development of core process mechanisms and extensions to action team context. *Journal of Applied Psychology*, *104*(3), 388–410.

Zickar, M. J. (2000). Modeling faking on personality tests. In D. R. Ilgen & C. L. Hulin (Eds.), *Computational modeling of behavior in organizations: The third scientific discipline* (pp. 95–113). Washington, DC: American Psychological Association.