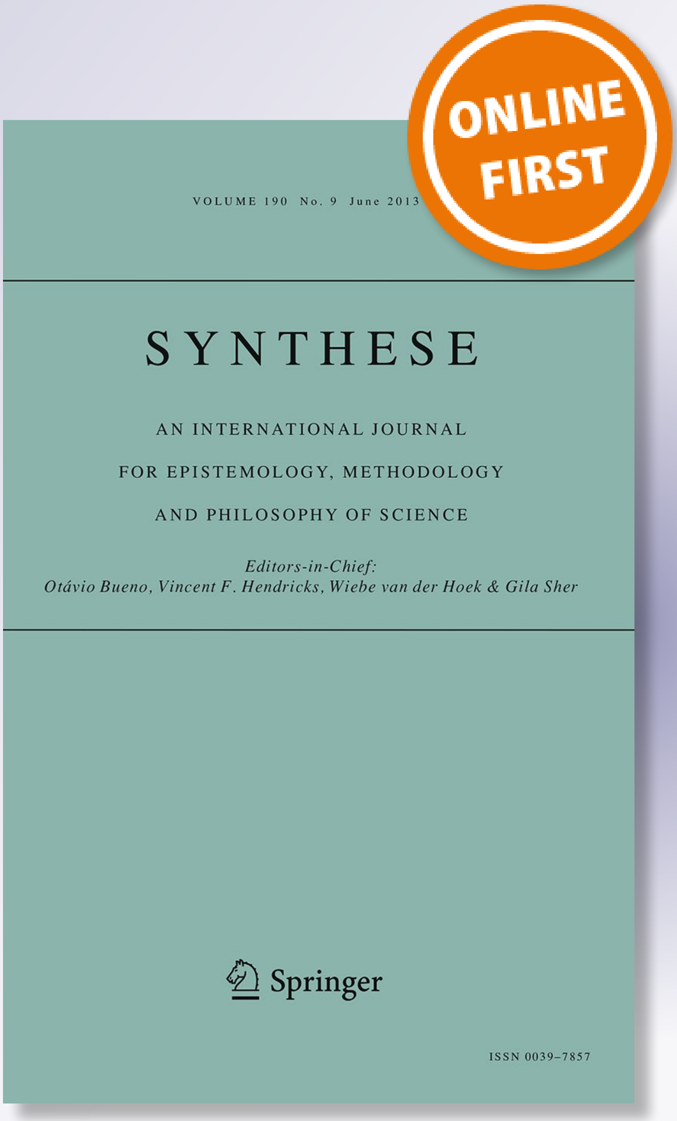
Rethinking correspondence: how the process of constructing models leads to discoveries and transfer in the bioengineering sciences

Sanjay Chandrasekharan & Nancy J. Nersessian

Synthese
An International Journal for
Epistemology, Methodology and
Philosophy of Science

ISSN 0039-7857

Synthese
DOI 10.1007/s11229-017-1463-3



ONLINE FIRST

VOLUME 190 No. 9 June 2013

SYNTHESE

AN INTERNATIONAL JOURNAL
FOR EPISTEMOLOGY, METHODOLOGY
AND PHILOSOPHY OF SCIENCE

Editors-in-Chief:
Otávio Bueno, Vincent F. Hendricks, Wiebe van der Hoek & Gila Sher

Springer

ISSN 0039–7857

Synthese
DOI 10.1007/s11229-017-1463-3

S.I. : MODELING AND REPRESENTATION

Rethinking correspondence: how the process of constructing models leads to discoveries and transfer in the bioengineering sciences

Sanjay Chandrasekharan[1] · Nancy J. Nersessian[2]

Received: 21 November 2016 / Accepted: 6 June 2017

Abstract Building computational models of engineered exemplars, or prototypes, is a common practice in the bioengineering sciences. Computational models in this domain are often built in a patchwork fashion, drawing on data and bits of theory from many different domains, and in tandem with actual physical models, as the key objective is to engineer these prototypes of natural phenomena. Interestingly, such patchy model building, often combined with visualizations, whose format is open to a wide range of choice, leads to the discovery of new concepts and control structures. Two key questions are raised by this practice: (1) how could discoveries arise from building external representations for which there is wide latitude in choice of the components from which they are built, and thus can be considered significantly arbitrary (the discovery problem), and (2) how could such discoveries allow engineering a real-world prototype system (the transfer problem). To examine these questions, we present two case studies of discoveries that emerged from the building of such computational models in the bioengineering sciences. We then develop a *process model* that accounts for the discovery and transfer problems raised by both these cases, focusing on the *process of building the model*. Specifically, to account for the discovery problem, we propose that the process of building such models gradually leads to a close coupling between the modeler's internal processes (which we consider a mental model) and the external dynamic model. To account for the transfer problem, we propose that

✉ Sanjay Chandrasekharan
sanjay@hbcse.tifr.res.in

Nancy J. Nersessian
nancynersessian@fas.harvard.edu

[1] Homi Bhabha Centre for Science Education, Tata Institute of Fundamental Research, Mankhurd, Mumbai 400088, India

[2] Department of Psychology, Harvard University, Cambridge, MA, USA

Published online: 15 June 2017

the process of building the model leads to the creation of an *enactive* model that is *generic*, which closely enacts, and thus reveals, the way the system-level behavior of the engineered prototype emerges in time through the interaction of its parts. This enactive replication process leads to the model and the prototype forming a new class, which allows concepts and control structures developed for the computational model to be transfered to the real-world prototype. We argue that this account requires rethinking correspondence in engineering sciences as a plastic and enactive relation. A closer focus on the process of building models is required to develop a general account of this emerging approach to discovery.

Keywords Discovery · Computational modeling · Bioengineering sciences · Enactive models · Distributed cognition · Incorporation

1 Introduction

Recent discussions in the philosophy literature have suggested that computational models developed in the context of established theories contain elements—often arbitrary—that provide such models with a semi-autonomous status (Winsberg 2003, 2010; Lenhard 2007; Fox Keller 2003). In contrast to such cases of model building within the context of theory, frontier research in the bioengineering sciences provides a domain where model building regularly takes place in the absence of established theory of the phenomena under investigation (MacLeod and Nersessian 2013), which can predict the system behavior or provide design constraints. A major objective in this domain, and engineering sciences in general, is to gain sufficient understanding and control of engineered models, so as to eventually be able to intervene in the natural system. Examples of interventions include controlling neurological disorders (such as epilepsy and Parkinson's disease) or cancerous cells, or being able to produce biofuels cheaply. These grand engineering objectives are a long way off, and much of the research effort is aimed at building physical exemplar models of selected aspects of the natural phenomena. These models themselves are the objects of interest, and the objective of modeling is to achieve the engineering goals of attaining control of, and redesigning, the built target systems in desirable ways.

In this analysis we focus on a relationship between two kinds of external representations that is unique to the engineering sciences: that between engineered exemplars or *prototypes* (such as synthetic blood vessels, cultured neuronal networks, and genetically modified plants), which work like scale models of selected aspects of the sought final system (and thus have representational features), and computational models built to understand in detail the working of these built prototypes. In this context, computational models are typically used to find ways of controlling, optimizing, or extending some desired behavior of the prototypes. As we will show, such computational models contain even more arbitrary elements (such as choice of modeling platform, selection of data, choice of theoretical/mathematical model, simplifying assumptions, choice of visualization, etc.) than in the reported cases where models stem from established theory. Yet, as we show in the case studies reported here, investigations with these external representations lead to new concepts and unexpected discoveries that transfer to the prototype, and, in some instances, to the broader natural phenomena (see also Knuttilla 2005). In this paper we examine two cases where building external representations with significant arbitrary elements led to discoveries that apply to real-world systems (both built ones and natural systems). These cases of discovery raise questions about how such transfer to physical/natural systems is possible from models with significant arbitrary elements.

We provide a *process account* of how this kind of discovery and transfer could be possible, focusing on the process of building the models, which is currently not discussed much in the literature. Specifically, we discuss how the process of building the model systematically infuses data into the model, and how this infusion process creates an enactive and generic model that captures system-level behavior of both natural systems and built prototypes. We then examine the broader implications of this type of modeling in engineering sciences, and argue that such cases raise some significant questions for philosophy of science. In particular, we suggest that these cases, and the larger engineering science enterprise of which they are a part, require understanding correspondence as *enactive* and *plastic*—a constantly evolving relation between built models (computational and physical) that emerge from the dynamic, incremental, and iterative building processes.

2 Building to discover

Building physical (in vitro) simulation models and computational (in silico) simulation models of phenomena of interest is a signature practice of contemporary bioengineering sciences. However, ethnographic data on this building process, and accounts based on such rich data, are scarce in the literature on modeling in bioengineering, where some studies have examined the way discoveries emerge from building models (Knuuttila and Loettgers 2013; Loettgers 2009; Green 2013). An example of a built physical model is the *flow loop* (Nersessian and Patton 2009; Nersessian 2012a), which is an engineered artifact that simulates, and thus helps examine, different types of blood flow (laminar, oscillatory) and their influence on gene expression and other cellular-level activity in tissue-engineered blood vessels (built-models of human arteries). Another example is a neuronal dish model (Nersessian and Patton 2009; Nersessian and Chandrasekharan 2009; Chandrasekharan 2009), which simulates learning in a living neural network, and enables researchers to examine the neural mechanisms underlying learning— specifically, how neurons with their connections broken apart come together to form new connections and learn new behavior.

Examples of computational models built to mimic biological phenomena of interest include simulation models of the plant cell wall and models of dopamine transport in Parkinson's disease. There are also "second order" computational models, such as a model built to simulate processes in the above-mentioned neuronal dish, which is a computational model of a physical model. Sometimes computational models are also built in a chain structure. The first model (say, a biophysical model) will generate virtual data of a downstream process (say, the generation of a metabolite), and these data help in the building of models of the second process, particularly in the estimation of model parameters. Note that these built systems are different from model organisms (see, e.g., Ankeny and Leonelli 2011), as the central objective of building the above models is to engineer the built system better and not primarily to understand biological mechanisms. Any understanding of biological mechanisms that emerges from these built models is secondary to the control objective.

Building such facsimile models that mimic or approximate engineered phenomena of interest is not a new practice. For instance, wind tunnels have helped mimic different aerodynamics for at least a 100 years (though they are now being replaced by computational versions). Recently, however, a combination of four factors has made the practice of building computational facsimiles more widespread, particularly in the bioengineering sciences:

- The complex, non-linear, and dynamic nature of the problems investigated in contemporary biology (and recent science in general), requires building computational models. This is because it is almost impossible to develop detailed conceptual models of cellular and molecular-level interactions of a large number of variables in your head, or using pencil and paper, as these processes: (1) involve many levels of structure (organs to cells to molecules), (2) occur across different time scales (signaling takes seconds, protein expression can take hours, circulation and organ growth takes days/weeks/years), (3) develop simultaneously, and (4) involve complex feedback loops.
- Massive amounts of data are now generated by experimental work in many areas of biology, such as high-throughput data in genetics, where the interactions among different variables are extremely complex, and cannot be understood without computational modeling. Further, the technology that generates the data relies heavily on embedded statistical models of the distribution of the data. These data usually require complex visualizations based on models, because they are difficult to represent, and comprehend, using traditional external representations such as graphs.
- Data in biology are closely tied to their context (e.g., specific cell lines, animals, diseases), and there is no existing theory that integrate and structure all these disparate and scattered data. Building computational models helps bring these data together in a structured fashion, where each component of the model is built by fitting the data.
- The development and easy availability of new technology that supports modeling and rapid prototyping has made computational modeling more widespread.

These factors, together with the technological resource environment of contemporary science, are driving the practice of building computational models in the bioengineering sciences, particularly in tandem with physical models.

We study the cognitive research practices related to such model building. This work is based on "cognitive ethnography" in the tradition of Hutchins (1995), which uses ethnographic interviews and field observations to study distributed cognitive processes in socio-technical systems. The research group of Nancy Nersessian and Wendy Newstetter has been extending this form of ethnography to investigate scientific research laboratories conceived as "evolving distributed cognitive-cultural systems" (Nersessian et al. 2003; Nersessian 2006). Over the course of 14 years, the group has investigated five bioengineering sciences laboratories located at a major research university in the United States, aiming at an integrative account of cognitive, social, and cultural dimensions of research practices involving different kinds of modeling.

In bioengineering sciences, basic research (such as endothelial cell biology) is pursued in the context of complex application problems (such as creating a living substitute artery). Two kinds of models figure prominently in this research: built in-vitro models of in-vivo phenomena, and computational models, either of in-vivo phenomena or of in-vitro models. The in-vitro models are often composites of biological and engineered materials (called "model-systems" by our respondents), designed to exemplify selected features of a class of in-vitro phenomena (Nersessian and Patton 2009). One case we discuss in this paper is the *dish model-system,* which consists of a network of living neurons plated on a multi-electrode array, for stimulating and recording neural activity, and connected to "embodiments" such as robots and computationally simulated "animats" in a closed loop, to study learning. These in-vitro models, which we here refer to as belonging to a broader class of *prototypes* (engineered exemplars), are complex systems in themselves, and researchers turn to building computational models to gain insight into how to control or redesign these prototypes. In both cases (in-vitro and in-silico), the new understanding of the target system—whether it is how to control the system or finding suboptimal features of the prototype design—is obtained through model building. It would not be far-fetched to conclude that understanding in bioengineering sciences, and engineering sciences in general, thus derives largely from building. It is no accident that the famous saying by Feynman ("what I cannot create, I do not understand") is misquoted by bioengineering scientists as "what I cannot build, I do not understand" (famously encoded in the first synthetic cell by Craig Venter; see also Voit et al. 2012). However, this *process* of "building to discover" (Chandrasekharan 2009) has not received much attention in philosophy. As we will argue, investigating this process provides novel insights into the *achievement* of correspondence, which is an evolving and dynamic relationship in this domain. The generation of insights through building of new biological systems has been discussed in the literature (Nordmann 2015; Gramelsberger 2013), particularly in relation to synthetic biology and the role of building computational models in this domain. However, ethnographic studies examining the *process* of building computational models, particularly to help develop new synthetic prototypes, and how discoveries emerge from this twin building processes, are rare (for an exception see Carusi 2014).

Based on the two case studies of building computational models in bioengineering sciences, we will show that, going significantly beyond the development of a representation of the target system's structure, building the computational model involves a coherence enterprise, where parts of the system are first built using data and theory from a wide variety of disparate sources. These parts, when brought together as a model, generate coherence between these disparate parts (essentially a thread that connects general patterns in data from disparate sources), and reveal how different variables interact to generate configural (i.e. system-level) patterns that emerge across time. The model thus generates an exemplar of the system-level dynamic *behavior* of the target system. For instance, parts of a metabolic pathway model of atherosclerosis would be developed using available pathway data based on neural cells, even though the atherosclerosis model requires modeling endothelial and smooth muscle cells.[1] The replication of a wide range of such related, and partial, experimental results from literature iteratively "infuses" real-world data into the model, while also synthesizing what is common about the data from such disparate systems. This systematic infusion of data at the element level, over time, turns the final model (of atherosclerosis) into an *enactive* exemplar, which can act out the system-level behavior of the target system (such as how plaques form, a feature that is not available in the metabolite level data used to build the parts). This scaling up to the behavior level is achieved partly through the internalization of a range of real-world experimental data about parts, and partly through the *enactive* capability of computational models.

There are two senses in which a computational model is enactive. One is the operational level idea that the computational model 'runs', while paper-pencil models do not. The running allows the modeler to examine in detail how different dynamic patterns emerge, through interaction between elements, across time. This is very difficult to do using pencil-paper models, as it is impossible to recreate such detailed configural dynamics in the imagination when the number of elements is high. This sense of enactive is part of our proposal, particularly in the discovery account.

The second sense of enactive builds on this idea, but it highlights how the running is crucial to understanding the part-whole relationship. The central point here is that the phenomena the modeler is interested in are at the whole system level (neuronal firing patterns, pathway dynamics), and these system-level dynamic behaviors emerge only in interaction and in time, and the model thus needs to act them out for validation. A rough analogy would be a model of walking that is based on muscle and biomechanics data. For validation, the model would need to act out walking (using an animat or robot), which is a system-level behavior in time, emerging from element level interactions. Without the enaction of walking, the modeler would not be able to understand how the elements interact to generate walking, which is a configural pattern that emerges in time through the interaction of various components (see O'Malley and Soyer 2012 for a related discussion on integration). Without the enactment of this configural pattern, the modeler would not be able to validate the model, or transfer insights from the model to a built system.

When the model is built, each replication of results from the literature, at the part level and the system level, changes the final model's parameters in the direction of a fit between model-data and real-world data. This gradual process allows the model to eventually enact the system-level behaviors of the target biological prototype system, as well as counterfactual scenarios that emerge from parameter values not seen in nature. These two features (internalization and synthesis of partial data from disparate literature, and the acted out system-level behavior of the target system) exists only in the model, and therefore, here-to-fore unknown behaviors can arise in the model.

As discussed above, enaction here refers to the model's ability to accurately act out dynamic scenarios at the behavioral level (i.e. system level), based on the synthesis of components (theory elements—such as equations—and data) taken from a wide

[1] Interestingly, although not modeling a prototype system directly, this model used some data that was generated through simulations of a physical model-system, the flow-loop-construct model-system, which mimics the flow of blood through the lumen.

range of literature. During the process of building the model, the enaction process allows the modeler to see patterns that are not available to other researchers with access to the same data, because the data in the latter case are not synthesized, and they do not cohere together to generate a dynamic behavior-level model. This makes imagining the way the whole system moves across time impossible without a running model, particularly the imagination of novel counterfactual scenarios based on many parameter changes.

Importantly, this enactive status develops slowly through the building process, and creates a *generic model* of the behavior of the prototype (rather than a model of the specific built system under study). There are two reasons for this. One, at the elemental level, the data infused into the model come from a range of experimental domains and cell types, which makes the structural level generic. Secondly, the model enacts system-level behavior (such as plaque formation, or patterned firing of neural populations) after each infusion of data, and these system-level behaviors, resulting from the interaction of data from a range of domains and cell types, become part of the model. This makes the model generic at the system level, as a range of system-level behaviors are exhibited by the model, based on disparate element-level data.

Note that in both the cases we report, data from disparate sources are used for the same reason: lack of good data. As shown in the flow chart of the modeling process in case 2 (Fig. 2), computational models are built in a number of steps. First the available data is split into two sets, training data and test data. The model is built up using training data, and it is tested using test data. For case 1, the test data was the built model of the neuronal dish, the training data was the data from the wider neuroscience literature. For case 2, the training data was from two wild type plants, the test data was from the same plants and their genetically modified versions. In the atherosclerosis case we discuss, neural cell data is used because enough muscle cell data was just not available.

Such disparate data are thus not chosen by the modeler while building the model—the modelers have to make do with such data. This means part of the generic nature of the model, particularly the part emerging from the disparate nature of the data, is an accidental feature that arises from operational constraints of the model building process.

Since one core component of the infused data, and the replicated system-level behavior, comes from the physical prototype, the infusion of data and the iterative building of the computational model creates a new class of enactive system. This system is now generic, as it subsumes data from the natural system and the built target system (prototype), and can replicate the system-level behavior of both these systems. These two systems now become instances of the generic model system. Once a new pattern is identified in this generic model, and this holds across many different scenarios generated by the model, the generic and class relations between the model and prototype make it possible to extend the new pattern to the target instance (case study 1 below).

The generic feature comes from the assimilation of data from other domains, which is accomplished by the model's capacity to enact (i.e. mimic the behavior of) both the parts and the larger system, using "borrowed" structural features from many domains.

When the model replicates the behavior of the prototype system as well, the generic character includes the prototype, thus making the model a generic structure that can replicate (1) the parts (say neurons and synapses), (2) the behavior of the real world system as a whole (neural firing in zebrafish), and (3) the built prototype system (the network of neurons in a dish).

Based on this analysis, we address the discovery problem, arguing that the development of the enactive status incrementally leads to changes in the cognitive system of the modeler, particularly in her ability to examine many possible scenarios in detail and mentally simulate how these scenarios could be generated by different possible mechanisms. This internal/external representational "coupling" is central to the discovery process (Chandrasekharan and Nersessian 2015; Chandrasekharan 2009; Chandrasekharan and Stewart 2007; Nersessian 2002, 2009a, 2012a), and allows the model to be used in such a way that it can make predictions about unknown elements of the very system it is modeling (case study 2 below). The transfer problem is addressed by appealing to the generic character the model develops over time, through the building process. Thus for both the discovery and transfer accounts, the *process* of building is crucial.

We develop this two-fold process account using the following structure. In Sect. 3, we report a case study of how a computational model was built to understand and control an in-vitro model (the dish model described above) and how new concepts and control structures were discovered through the building process. We then discuss a central question raised by this case, namely how a new concept developed for, and using, a significantly arbitrary representation (visualization of a computational model) could transfer to the real-world built model-system. We propose that the enactive correspondence set up through the building process is key to answering this question. Section 4 reports a second case study, where building a computational model of the biological pathway of lignin production led to an even more remarkable discovery, that the scientifically accepted pathway requires an extension, to include a new element that contributes significantly to the production of lignin. We offer an account of how the process of building a model could lead to the model identifying unknown elements of the very system it is modeling. Section 6 concludes the paper with a general discussion of the theoretical and methodological issues raised by these two case studies. All quotes within the case studies are from interview data.

3 Case 1: Neural engineering

This case outlines an episode of conceptual innovation, which occurred over a 2-year period in a neural engineering laboratory. It involved constructing a computational model of a dish of cultured neurons, which itself is a living in-vitro physical model of cortical neural network activity. The computational model was developed to understand spontaneous "bursts" that arose in the in-vitro model, but not in properly functioning in-vivo animal brains. The computational model led to a reconceptualization of "burst" phenomena in the in-vitro model and the development of "programmable" neurons. (See Nersessian and Chandrasekharan 2009; Nersessian 2012b for fuller discussions.)

In the period of our investigation, the lab's central research problem was to develop an account of learning and plasticity in networks of neurons, which were considered to more closely model learning in the brain than single neurons. For this, the lab had developed a physical model-system: an in-vitro network of cultured neurons, referred to as "the dish." Building this in-vitro model-system involves extracting neurons from embryonic rats, dissociating them (breaking the connections between neurons) and plating them in a single layer on a dish with embedded electrodes (MEA: multi-electrode array), where the neurons regenerate their connections to form a network. The researchers interact with the dish by stimulating this neuronal network with different electrical signals (electrophysiology), and feeding the output to "embodiments" (robotic devices, or visualized creatures that move around in simulated worlds) that support closed-loop feedback (using a translator program that maps the dish signal to motor commands).

The stated goal of the research was to understand the dynamics of learning in the neuronal network, in such a way that a control structure could be developed, which would allow the dish to be trained to control the embodiment systematically. The design of the physical dish incorporated constraints from the current understanding of neurobiology and chemistry, as well as those relating to electrical engineering and other technologies used in the lab. To begin with, the researchers stimulated the neuronal network using different electrical signals and tracked the output. Researcher D4 then tried, unsuccessfully, to replicate a plasticity result reported by another group. One key problem she faced was what she called "bursting"—a form of network-wide electrical activity spontaneously exhibited by the in-vitro neuronal networks, an extension of the term as used in single neuron studies. Bursting was thought to be a problem in achieving learning ("plasticity"), because the network-wide activity prevented the detection of any systematic change that arose in response to controlled stimulation. Whenever D4 conducted a plasticity experiment, the network produced bursts. The lab interpreted those as "*noise in the data … noise interference in the way … so it is clouding the effects of the learning that we want to induce*".

The group hypothesized that bursting arose because the neurons lacked the sensory inputs they would ordinarily get if they were in a live animal's brain. Based on this conceptual framing, and the engineering conception of noise as something to be eliminated, D4 began working on quieting bursts in the dish, by providing the network with artificial sensory input, using electrical stimulation. After trying a wide range of stimulation patterns, she achieved a breakthrough, managing to quiet bursts entirely in a neuronal network. However, she tried inducing learning in the quieted network over the next 6 months, but was unsuccessful. This was mostly because the activity pattern evoked by a stimulus did not stay constant across trials, but "drifted" away to another pattern. This drift prevented her from tracking the effect of a stimulus, because the network never responded in the same manner to a constant stimulus.

As D4 was trying to quiet the network, D11 branched away from working with the in-vitro model-system, to develop a computational model that mimicked it. D11 felt that to understand the phenomena of bursting, he needed to "see" the dish activity, make precise measurements of variables such as synaptic strength, and run more controlled experiments than could be conducted with the physical dish. Interestingly, D11 built the initial in-silico model not using the experimental data from their dish, but drawing from other studies in the broader literature, involving single neurons, brain slices, and computationally simulated networks. Importantly, the model did adapt constraints from the lab's dish system, but these had to do only with the construction and structure of the dish, not its system-level behavior. The other parameters (constraints) of the model, such as type of synapses, synaptic connection distance, percentage of excitatory and inhibitory neurons, conduction delay, conduction velocity, noise levels, action potential effects, and spontaneous activity were based on results reported in the broader neuroscience literature. Further constraints came from a neuroscience modeling platform. The computational model was thus based on the physical design features (such as thickness of neuron layer) of the lab dish, but it was tested and optimized with data reported in the wider literature. Only after the model's behavior replicated the experimental results in the wider literature did D11 replicate the lab's own data. The model, as developed, was thus a second-order in-vitro system—an in-silico model of the activity of a *generic* dish. The model is generic because it incorporates replications of not just the lab's dish, but also disparate experiments from the wider literature.

D11 built many tentative versions of the computational simulation, and probing these, he started to get what he called a "feel" for how the network behaved under different conditions. He then developed a visualization that captured the activity of the network as it ran. The visualization enabled D11 to observe—literally see—interesting spatial patterns in the way the model responded to different stimuli. These patterns were novel and distinct from what was known about the behavior of cultured dishes. These visualized patterns figured centrally in the development of the novel concept of "center of activity trajectory" (CAT). Figure 1 provides an outline of this iterative and incremental modeling process.

The model's visualization of the network's activity showed the movement of an *activity pattern* across the entire network, in real time. This visualization was significantly different from the visual representation that was used for representing the activity in the in-vitro dish, which only tracked activity at each electrode of the in-vitro system, using an oscilloscope-like representation of spikes of electrical activity developed for this purpose. The individual neuronal activity was hidden in this representation, as it only captured which electrode (cluster of neurons) was activated and by how much. This representation did not have a *network* structure, and thus it could not show *burst movement across the network*. It was thus not possible to see from the dish display whether there were patterns moving across the network. Such patterns, however, were revealed by D11's visualization of the in-silico model.

The computational model offered other advantages in exploring bursts. For instance, the network could be stopped at any point to examine its states, and started again from there. It was possible to manipulate significant variables, such as synaptic strength, which were not accessible using the in-vitro model. A large number of such counterfactual experiments could be run at no cost, since the computational model could be changed easily, without the detailed, laborious, and expensive processes involved in setting up and maintaining a living dish. These features proved to be a powerful combination, particularly when coupled with the visualization that enabled tracking the activity across the network as it was happening. They gave D11 immediate access to a range of configurations and data that the living dish could not provide, and these could be examined and reexamined, and comparison experiments run instantly. From this

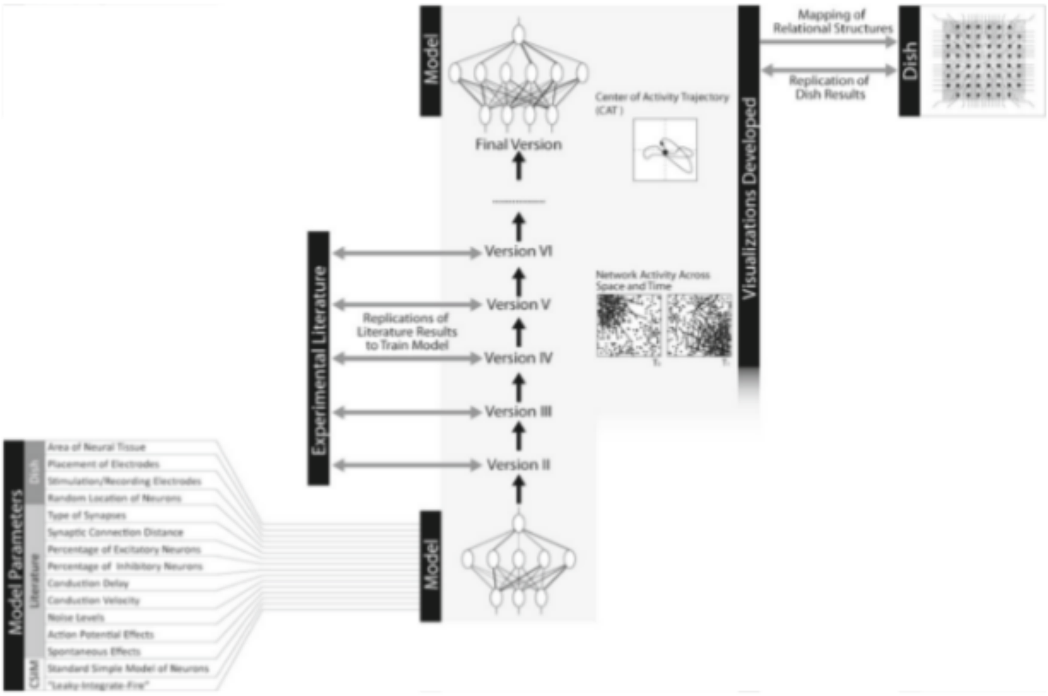

Model
Final Version
Version VI
Version V
Version IV
Version III
Version II
Model
Experimental Literature
Replications of Literature Results to Train Model
Center of Activity Trajectory (CAT)
Network Activity Across Space and Time
Visualizations Developed
Mapping of Relational Structures
Replication of Dish Results
Dish
Model Parameters
Dish
Literature
CSIM
Area of Neural Tissue
Placement of Electrodes
Stimulation/Recording Electrodes
Random Location of Neurons
Type of Synapses
Synaptic Connection Distance
Percentage of Excitatory Neurons
Percentage of Inhibitory Neurons
Conduction Delay
Conduction Velocity
Noise Levels
Action Potential Effects
Spontaneous Effects
Standard Simple Model of Neurons
"Leaky-Integrate-Fire"

Fig. 1 The iterative modeling processes that led to the change in the understanding of bursts and the new CAT concept

process, he noticed that there were repeating *spatial* patterns in activity, as the activity propagated across the network. The spatial patterns were seen both when spontaneous bursts arose in the model network, and when the model network responded to stimuli. He found that there were "similar looking bursts" that propagated across the network, and there appeared to be a limited number of what he called "burst types." D11 then began collaborating with D4 and D2, a graduate student who had been working on dish embodiments. The group then decided to investigate bursts further together, as a possibly interesting pattern, that is, *a signal*. The radical change in perspective here is worth noting: "burst" transforms from something akin to noise (that needs to be eliminated), to a pattern, a signal that could possibly lead to a control structure for training the network. Note also that this change arose from *the group* running many simulations with different situations and parameters over time, a process leading to a group consensus on the conceptual shift. This group process was possible because of the external, *manifest* nature of the built model.

From this point, potential problem solutions were mapped from the in-silico model to the in-vitro model. In particular, the computational model helped them get past the drift problem. The group came up with a range of ways to quantify the spatial properties of moving bursts (using clustering algorithms and statistical techniques), and these measures proved to be immune to the drift problem, i.e. the activity pattern evoked by a stimulus now stayed constant across trials. These measures were first developed for the computational model and then equivalents (by analogy) were developed for the in-vitro model. They included several conceptual innovations: "burst types," "spatial extent" (an estimate of the size and location of the burst in the dish), and "center of activity trajectory" (CAT), a vector capturing the spatial location of the electrode along with the firing rate). These spatial measures of bursts were then shown to be better indicators of plasticity than plain spike responses to probe stimuli. So in the matter of a year, based to a significant extent on the spatial patterns generated by the computational model's visualization, the group's theoretical position shifted from bursts as noise to bursts as signals, and this in turn, led to a possible control structure.

Of these spatial measures, CAT is the most noteworthy, for two reasons. First, because it is a novel concept, articulated from both the computational visualization and an analogy with the physics notion of center of mass, which was applied to the neuronal activity pattern seen in the visualization. Second, because it emerged entirely from the visualization of the computational model and would likely be impossible to conceptualize and properly formalize without it. CAT is an averaging notion, similar to the notion of a population vector, which captures how the firing rates of a group of neurons that are only broadly tuned to an action or stimulus (say an arm movement), when taken together, provide an accurate representation of the action/stimulus. However, CAT is more complex than the population vector, because it tracks the *spatial* properties of activity as it *moves* through the network. For instance, if the network is firing homogenously, the CAT will be at the center of the dish, but if the network fires mainly at the left corner, then the CAT will *move* in that direction. CAT thus tracks the flow of activity (not just activity) at the population scale, and on a much quicker time scale than population vectors. When applied to the in-vitro network, CAT provides a novel representation of neuronal activity—a new concept.

The CAT is like a signature for a burst type, in that each burst type has a corresponding range of similar-looking CATs specific to that type. While the CAT concept was developed from the visualization, the relation between the CAT concept and the visualization is not a direct mapping link. The visualization worked as a generator of many types of activity, which, when put together, led to the new CAT concept. Our account ends here, though in later work the researchers combined CAT and techniques developed for burst quieting to develop a set of stimulation patterns (a control structure) for the dish that led to supervised learning by the living neuronal network, in effect making the in-vitro dish neuron network programmable. Using these stimulation patterns, the living network in the dish was trained to control a computationally simulated creature and a robotic drawing arm. Thus the CAT concept, which was developed based on an arbitrary spatial representation (the visualization) of the data produced by the computational model, transferred to the engineered physical model (the Dish) as well as to a real-world control structure, where the Dish was used to control a robotic arm.

3.1 Discussion

The most perplexing aspect of this case is that the network visualization used by D11 is largely *arbitrary,* as there are many different ways to visualize the data. There is no necessary reason why a network representation should be used—he could have used a visualization similar to their spike graph representation of the living dish. Even after choosing a network representation, there are many ways in which the network activity could be visualized. This raises the central question posed by this case of discovery-by-building:

> How could concepts developed from the **spatial** analysis of an **arbitrary** representation transfer to the actual dish model and provide a control structure for supervised learning?

The answer to this question lies partly in the iterative building process and partly in the nature of the visual representation, and the relation these two elements establish with the modeler's imagination. The underlying network's parameters are continually being reconfigured by replications of various experiments, from different domains in neuroscience. Each of these replications can be viewed as *infusing* data into the network, as the inputs and the outputs change the network's parameters. This process adds complexity to the network in a cumulative fashion, as each replication builds on the network created by the previous replication. The final network structure, including parameter settings, is based on a synthesis of all the data used from the literature, as well as the lab. This global synthesis of data only existed in the model, and this synthesis could literally be seen at the behavioral level, because the visualization was dynamic and captured the network structure *as a whole.* This global pattern was fine-tuned by the replications, to more closely mimic the behavior of the network, until the system gained sufficient complexity to *enact the behavior* of the dish network as a whole, including counterfactual cases where the modeler changed parameters to explore the network's global behavior. The model also provided the option of exploring the network locally, as the model could be examined at different levels of resolution (such as 20 neurons, 10 neurons, and so forth) and different parameter settings (such as number pf inhibitory/excitatory neurons, synaptic distance, and synaptic connection strength)

Since the visual representation captured the underlying artificial network's structure in an integrated fashion, making the interrelationships between the different points and their movements across the network clearly visible, it lowered the amount of integration in imagination the modeler needed to do, which was extremely high in the case of the MEA spike graph representation. Seeing the network's behaviors directly and having the ability to stop-and-poke them *as behaviors*, while also examining the numbers that generate the behaviors, allowed D11 (and others in the lab) to generate a close coupling between their mental and external representations.

Note that CAT captures the centre of activity of *patterns*, similar to the way centre of mass captures the behavior of objects. CAT can be applied to find the centre of *any* activity with spatial extent, it is thus *generic*. This generic abstraction is based on, but can be separated from, the (arbitrary) visualized network representation. Since it is generic, CAT can be seen as capturing the model's replication of the dish activity as one instance of an activity with spatial extent; its ability to capture the activity of the actual dish is another instance. The CAT concept cannot be directly mapped to the visualization, in the sense that CAT does not exist in the visualization, just as centre of gravity does not exist in objects (Dennett 1992). The visualization and the model are *generators* of spatial activity (behavior), which is accounted for by the CAT concept. This abstract concept is what transfers to the dish.

This account provides a new way of thinking about the transfer problem: the iterative building of the model via the infusion of data from different domains and the wider literature creates a *generic* model, as well as a new class of systems, of which the real world system and the target (built) system are instances. A new concept is then developed for the model instance, and if it holds good, the concept, modified in appropriate ways, is also applied to the target instance. This application process is easier when the target system is also a built one, since it can be redesigned to achieve the transfer involved, in this case to create a control structure for supervised learning that can elicit behavior that was previously not possible.

In the view we have been developing here, building a computational model (in the bioengineering sciences, and possibly other sciences) involves adding real-world data and complexity (in the dish case, the way the network nodes are configured) to a system (through data from the wider literature and the lab) such that it can *enact* the behavior of the related phenomena as well as the system under study. Each run during the building process reconfigures this enactive ability, until the model gains sufficient complexity to enact the target system behavior. The final runs are a *probing* of this enactive ability, the possibilities of which the modeler does not entirely understand, but has a "feel" for. For instance, the control structure used to direct the behavior of the animat and the robotic hand using the dish neurons consists of providing the network with a patterned stimulation, and then, instead of reinforcing this stimulus, the network is provided a random background stimulation to "stabilize" its response to the patterned stimulation. This training method is rather counterintuitive to existing ideas about learning and reinforcement, and comes from the building process. However, the modelers do not understand why/how the random stimulation stabilizes the neuronal network. The intellectual work at this stage is thus not interpretation of the results to generate explicit knowledge, but the use of different probes to generate a range of counterfactual behaviors, and the capture of this range of behavior within a margin of error framework. This range of error is then used to develop a control structure, which is used to manipulate the target system, and generate desirable behaviors.

4 Case 2: Metabolic engineering

This case reports how building a computational model of the biological pathway involved in lignin production led to a remarkable discovery. It was discovered that the standard pathway required an extension, to include a new element which was found to contribute significantly to lignin production. The initial goal of the modeling process, however, was just to tweak the lignin pathway so as to make a better biofuel. A pathway diagram captures the main reactions involved in a metabolic or signaling process, and also the positive and negative regulation effects, which specify how the presence of different metabolites has a positive or negative influence on different reactions. The pathway diagram developed by the modeler brings together pieces of information that are spread over a wide set of papers, databases, and unreported data provided by experimentalists. Computational models seek to capture the way the concentration levels of different metabolites in a given biological pathway change over time. Lab G researchers build a variety of models, but most focus on building ordinary

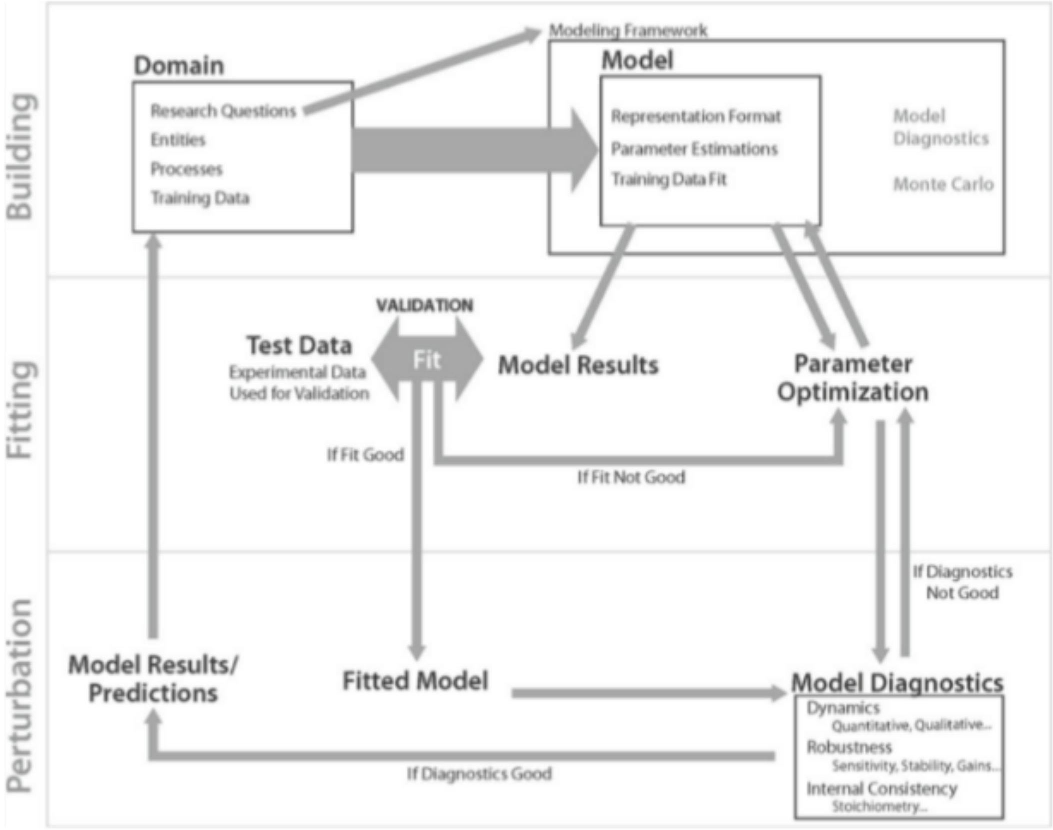


Fig. 2 An outline of the iterative modeling process typically used in Lab G

differential equation (ODE) models of metabolic systems. Lab G models do not use real-time dynamic visualizations. Parameter values are changed using scripts. Results for different parameter values are compared using a deck of graphs, where each graph plots the concentration value of a molecule in the pathway across time. These graphs are used by the modeler while discussing the model with collaborators and other team members. Figure 2 provides an outline of the typical Lab G modeling process.

4.1 Remedying recalcitrance

Lignin is a natural polymer that helps harden plant cell walls. The hardening provides the plant with structural rigidity, and thus supports growth. While this hardening property is biologically useful for the plant, it is a problem for the biofuel industry, because lignin is difficult to breakdown (exhibits "recalcitrance") when biomass is processed into fermentable sugars (using enzymes/microbes). This recalcitrance of lignin makes the extraction of sugars from biomass difficult and costly, and biofuel production is therefore uncompetitive currently in comparison to fossil fuels.

To solve the recalcitrance problem, genetically engineered plant varieties with lower lignin content (working prototypes) have been developed (for various plant species such as alfalfa and poplar). However, these transgenic species are not optimal, as they exhibit unforeseen consequences, such as lower expression of only one of the three monomer building blocks of lignin (termed monolignols, H, G, S). Computational modeling provides a way of investigating the mechanisms underlying lignin production, and the understanding derived from modeling could contribute to the development of transgenic species (optimized prototypes) that have low lignin content as well as good growth (very low lignin content will lead to plants not having structural integrity, and this could prevent growth). The modeling could also allow developing plants with different ratios of lignin monomers, such as a lower S/G ratio, which also helps in improving the extraction of sugar from plant cellulose.

To model the pathway involved in lignin biosynthesis, Lab G collaborated with an industry research lab in another US state. The model was being developed to understand how the monolignol components (monomers: H,G,S) of lignin are generated by the components of the lignin pathway. This is a new modeling area; most other modeling efforts in the biofuel domain involve developing bio-informatics models, or models of organisms that break up the plant biomass.

G10 was the principal researcher working on the lignin biosynthesis model. He was a bioengineering Ph.D. student, and holds a bachelor's and master's degrees in electrical engineering. G10 started by building a model of the lignin pathway in alfalfa, based on existing knowledge about the pathway from the literature, and data from papers. He faced three significant problems in getting data from his collaborators. One was that the granularity of the data provided was different from the granularity required by the model. A second problem was delays, and not having good access to the collaborators. A third problem was the unwillingness of the collaborators to part with data before publication.

While waiting to get data and feedback on the alfalfa model from his collaborators, G10 developed a model of lignin biosynthesis in poplar, a related species for which a significant amount of data are available publicly. This model helped him understand the lignin pathway better, and also develop a two-step modeling technique that helped in dealing with the complexity of the lignin pathway. The model also led to the development of a novel parameter estimation method, suited to the available data (consisting mostly of S/G ratio in genetically engineered plants). This method involved the following three-step process:

- First, each parameter value was constrained to a *physiologically realistic* range (note the use of a biological constraint—survival—to simplify a mathematical problem; this constraint is invisible in the final model, and becomes explicit only when the process of building the external representation is examined). A Monte Carlo type sampling of parameter values within this range was then done to simulate the model. These simulations generated a large set of model-data (different S/G ratios).
- In the second step, correlations were established between the parameter values used in these simulations and the S/G ratios, to select significant parameters (any parameter where a small change lead to a large change in the S/G ratio).
- In the final step, these significant parameter values were optimized (using linear programming and simulated annealing) so that the difference (SSE: sum of squared error) between model results and experimental data was minimal. This generated an '*ensemble' of models* with low SSE. Here the term 'ensemble' only captures

similar models with different parameter settings (see Parker 2013 for a discussion of ensemble models with wider differences).

This ’ensemble’ of models helped in identifying the key reactions that influenced the S/G ratio. The ensemble of models was then used to simulate two transgenic experimental results that were not included in the training data. The models (note the plural; there was no unique model) were able to approximate these test data, as well as provide some mechanistic insight into the working of the pathway, and these mechanisms were found to be supported by available experimental evidence.

Based on this validation by data-fit as well as mechanism, the ensemble of models was used to make some predictions about how the pathway could be engineered to reduce the S/G ratio. Even though the model was based just on publicly available pathway information and data, it helped G10 create a modeling template for the lignin domain. Particularly, the three-step process was useful in managing the mathematical complexity of the system, and the parameter estimation process was developed in a way that was tailored to data in this domain. These templates were then applied, with variations, to the alfalfa model, to make significant discoveries relating to the lignin pathway.

4.2 The Alfalfa model

Once G10’s collaborators had published some of their experimental work, they shared their data (excel files) with him, which he incorporated into the Alfalfa model. The collaborators did not provide G10 with the pathway structure, only data were given. Some other experimental results that the group had reported in the literature were also incorporated into the model.

The alfalfa model was more complex than the poplar one, since there were more transgenic species available (7, each created by down-regulating an enzyme) than the poplar case. Further, the data included the lignin levels at different points of growth of the plant stem (8 different internodes), and the lignin levels were different for each of these growth points. Accounting for this differential expression of lignin at different growth points was a challenge. However, now there were lots of data available.

The modeling approach used was a variation of the one developed in the case of poplar. This model made the assumption that the lignified stem tissues of wild-type alfalfa plants evolved to maximize the production of lignin monomers, and this biological assumption (once again, invisible in the final model) was used to develop the objective function (a mathematical term used in the optimization of parameter values).

In the second step, the transgenic plant data (for every inter-node of the plant) were modeled using a method where it is assumed that a mutant strain with a genetic modification (the down-regulation of an enzyme in this case) functions as closely to the wild-type as possible within the limitations imposed by the mutation. This is a biological assumption (again, invisible in the final model) that provides a mathematical term, and it helps solve the following modeling problem: how to use most of the pathway of a wild-type plant together with the engineered pathway (and data) of a genetically modified plant. If the modeler considers the genetically engineered plant’s

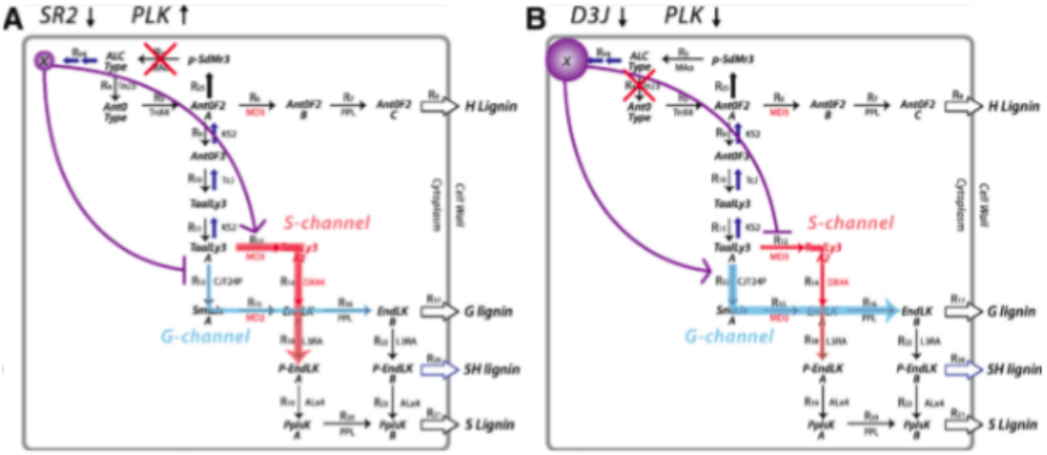


Fig. 3 The revised lignin pathway. The *red*, *blue* and *purple* elements indicate the specific changes proposed by the modeler. Metabolite names have been replaced with alphabet sets. (Color figure online)

pathway to be different (in terms of values or structure), then the model developed for the genetically modified plants could be very different from that of the wild-type, making what the modelers call the "feasible space" of the model very large.

Finally, a Monte Carlo type simulation similar to the poplar case (sampling of many parameter values) was performed, to understand the role of kinetic features of the participating enzymes.

This three-step modeling process led to a series of insights (six postulates). These include the reversibility of some enzymatic reactions (blue arrows pointing up in Fig. 3) and the possibility of independent pathways for the synthesis of G and S monolignols. However, one finding stood out: building the model showed G10 that the traditional pathway—used by almost everyone in the field for decades—is incomplete, and an element (termed X by G10; this naming is significant, as it indicates G10's lack of biological knowledge about potential pathway elements) *outside the standard pathway* has a significant regulatory effect on the behavior of the lignin pathway.

This finding would have been almost impossible without modeling, as the modeling process gave G10 a global view of all existing data on lignin and how they fit together, as well as how the data fit the pathway structure. This comprehensive view that comes from modeling led to him questioning the standard pathway as a last resort. Experimentalists, who lack this comprehensive view, would seek to explain their (isolated) results using the standard pathway. The proposed role of X in the lignin pathway, if correct, would rewrite the scientific consensus on the lignin pathway significantly. The proposal created much interest among G10's collaborators, and significantly increased their willingness to conduct experiments on the model's prediction. Their experiments identified a possible candidate metabolite (salycylic acid) that played the role of X as identified by G10's models. Based on this finding, a paper was written jointly with the experimental collaborators, outlining the *modeling and experimental results*, and published in a high impact *modeling* journal. According to G10, the experimental collaborators knew about the existence of the candidate metabolite (salycylic acid) outside the lignin pathway), but they did not consider the metabolite important in influencing the production of lignin.

This result illustrates clearly the ideal case of modeling—of the model making a significant and non-obvious experimental prediction, which is then tested and validated by experimentalists. This shows how building models in engineering sciences spans both engineering and science, at the same time. It also illustrates how the process of building the model can change the very knowledge from which the model is created.

5 Discussion

> The key question raised by this case of discovery-by-building is this:
> How could a model reveal hidden causal elements in the very pathway it is modeling?

Note that the element X is unknown to the modeler, but he is making *specific* predictions about its *significant influence* on the pathway, where the mechanism he identifies cascades down multiple levels in the reaction. This significant and specific influence is unknown to the scientific community. So this extension of the model is not a simple improvement or an addition, but a re-formulation of the pathway, as the element has a significant influence on the way the reaction proceeds. The discovery and transfer problems are inseparable in this case, as unlike case 1, where a new concept was discovered and *applied* to a target prototype system, here the model generated a hypothesis about the natural world, and it was found to hold true. As in case 1, this discovery can be *applied* to the prototype (transgenic plant), but this application comes from the demonstration (transfer) that it exists in the wider natural system (wild type plant).

It is worth noting that the original goal of the lignin project was tweaking a given pathway, which is an engineering goal. But G10 ended up changing the standardized pathway, which represents the scientific consensus on the mechanism underlying lignin production. This is a basic biological science discovery, generated by a novice biology learner (an electrical engineer), based on a few months of bio-systems modeling. The modeler has read biology papers, but his understanding of biology is very data centric, as he did not understand the biochemical elements involved and how the structural features of the elements constrain the reaction. The differential equation model does not require him to think in biochemical structure terms, which is required to understand the pathway mechanism. The finding is thus remarkable; it is analogous to a novice astronomer building a model of the (known) solar system, and then proposing, correctly, that there is a planet X outside Pluto which is influencing the movements of earth in specific ways. The built model is thus not just a replica of an existing standardized structure (the pathway) that allows tweaking the given pathway—it is a mechanism that allows discovering unknown features of the pathway itself. *The model can thus uncover hidden mechanisms in the very thing it is modeling*. In our view, accounting for this feature of the model requires understanding the building process, particularly how building leads to an enactive system, and the cognitive changes that accompany this shift. As in case 1, the iterative building process involves replicating several results from the literature, and this leads to the development of an enactive model, where the model's behavior eventually comes to parallel the pathway dynamics (when both behaviors are re-represented as graphs).

How does an external representation achieve this ability to enact a real system? This question is all the more interesting because the structure of the external representation is *arbitrary* to a significant extent, as in case 1. For instance, the model could be developed using other techniques, such as agent-based modeling, and even within ODE-based modeling, there are other possible approaches. There are data and theory elements borrowed from other domains. Elements are added and dropped from the pathway based on data fit. The final network used also has an element of arbitrariness, as it is pruned based on computational and biological considerations. This level of arbitrariness is possibly a feature peculiar to bioengineering, and not computational modeling in general. The lack of good data and theory, the overwhelming computational complexity of the problems, and the engineering objective driving model building (building prototypes) are some of the factors that contribute to this arbitrariness.

In our view, in the case of G10 as well, the enactive ability is achieved by *infusing data* into the model, but here it is more complex, and involves a highly recursive process. There are three distinct elements of the model—data fit, parameter values, and network structure—that can be altered in many ways to replicate experimental data. During the building process, these elements are tuned iteratively and in tandem, until they lock together like pieces of a jigsaw puzzle. The lock-in happens because the iterative changes constrain each element and the interactions among them. The engine behind this process is the fitting of data.

The notion of fit is complex, as it is not a point-by-point replication of all experimental data for all variables. Rather, "fit" usually means the model replicates the *trends* (metabolite production going up/down) in the experimental data, for *most* of the major variables. In other words, fit is a global pattern, and it is approximate. While estimating unknown parameters, the modeler uses the fit with the experimental data as an anchor, in the following way. For each change in a parameter, the way the model's output maps to the experimental results (the fit landscape) changes. But only parameter values that improve fit, or keep fit at an acceptable level, are considered. The building process proceeds by using the global behavior of the model (fit) as an anchor to specify the local structure (parameter values), which is involved in generating the fit itself. Since the fit is also used to add/delete components in the pathway, the model-building process can be thought of as a *coagulation process*, where each of the three changeable elements (pathway-structure, parameters, fit) are fluid in the beginning, but get more and more constrained by their interactions. This process is very complex and would be impossible to do in the head, and thus cannot be achieved without building the model. Gelfert (2009) presents a similar example in the domain of mathematical modeling, where abstract-level relations help seemingly arbitrary models cohere with each other. This can be thought as a type of coagulation, where the changeable elements are formal structures.

Typically, fit is understood as the rough matching of graphs (of data generated by the model and the experiment) generated by tweaking the final model. The understanding of fit we have developed, from studying computational modeling practice in bioengineering labs, is different, and considers fit as a dynamic and interactive process that is based on, *and helps build*, an enactive external representation that is generic—*fit is a critical building process*. Each replication of experimental results by the model changes its parameter structure, and thus cumulatively *adds complexity to the model*. This process also makes the model generic, as it synthesizes disparate data, until the model fits all available experimental data well. At that point, the model can *enact* the real system, i.e. play out in time how the interaction of local elements lead to the global and local dynamic behaviors of the pathway that is being examined, including counter-factual scenarios based on stopping and manipulating the simulations part way. The building process brings together, and coagulates, data fit, parameter values, and network structure, and thereby turns the model into what a Lab G researcher called a "'representational machine" that can enact the dynamics of the biological system. In the lignin case, the fit between the model and experimental data graphs (a critical mediating representation) is not a task endpoint; it is the way to add more and more complex behaviors to the model, thus tuning the parameters and creating the representational machine, which can then enact a range of behaviors under investigation. Once this is achieved, the model's behavior can be changed to explore counter-factual cases, which provide insights into redesigning the pathway. This 'representational machine' role of the model is much more limited than the idea of 'representational engines' (Ihde 2000), where technology acts as models that create theoretical and epistemological frameworks, and 'performative models' (MacKenzie 2008), where models become part of institutions such as markets.

What are the cognitive changes that help the modeler discover the unknown structure of the thing he is modeling? Our tentative answer is that this discovery is based on two critical cognitive changes, both deriving from the coupling between the external model and the internal (mental) model of the mechanism as imagined by the modeler. One, similar to case 1, the modeler gains a *system-level view* (the way different elements change to create new configurations in time) of the pathway; two, again similar to case 1, the modeler gains a detailed understanding of the pathway's *dynamic behavior*, but this time with a limited visual representation (graphs). The building process allowed G10 to develop a system view of the lignin pathway—an allocentric perspective of the system, where each element is understood in relation to other elements and the system-level behavior, is gradually developed from the thousands of runs of simulations, using many parameter combinations, and the analysis of system dynamics for each simulation. These cumulative simulation runs during building also provided a detailed understanding of how the system behaves under different conditions. This detailed understanding of the dynamics, together with the system view, provided G10 with an *intuitive* sense of the biological mechanism—an imagined model of how the pathway structure could generate different types of experimental data. A similar case is reported by Ferguson (1992), where an engineer who extensively worked in wind tunnels developed a design principle for supersonic flight (the Area Rule), and his colleague comments that he is a "... a guy who just had a sense of intuition about these kind of aerodynamics problems. He sort of feels what the air wants to do." Similarly, G10 has an implicit sense of what the pathway can do, but this understanding is not available for articulation, as it is based on extensive interaction with the parameters and resulting system-level dynamics of the model, and the incorporation of this dynamics into imagination (see Chandrasekharan 2014 for a wider discussion of the cognitive processes underlying such implicit understanding).

This intuitive understanding of biological mechanism is based on the close coupling between the external model and his imagination, and it is what enabled him to extend the pathway structure in a highly constrained fashion, to account for experimental data that could not be accounted for by the current pathway structure. The modeler's allocentric perspective of the system as a whole and the detailed understanding of dynamics, however, are developed solely through the model-building process, and *not by gaining expertise in biology*. The model-building process thus creates abstractive capabilities as well as intuition about the pathway's dynamics (which the Lab G director often calls "a feel for the model"). This is an instance of what Hutchins calls creating new cognitive powers by creating new environments (Hutchins 1995; see discussion below). Bioengineering scientists create new cognitive powers through the building of modeling environments.

6 General discussion

Here we explore the main implications that follow from our analysis of the two case studies. The two cases present discoveries about physical systems that arise from building computational models with significant arbitrary elements. We have proposed three key points about the way discoveries are made and transferred by building computational models in the engineering sciences.

1. The process of building a computational model in this domain gradually creates an enactive system that is able to mimic the dynamic behavior of the engineered prototypes, both at the global and local levels, including counterfactual scenarios. (It is worth noting that the prototypes could also be engineered to mimic the model, a reverse-correspondence that is the engineering objective.) This enactive ability comes from the gradual tuning of the model's parameters, through the cumulative replication of many experimental results (considered) relevant to the target system under study. This enactive system is a generic model, as it incorporates results from a wide variety of experiments, including from other empirical domains. In effect, the model could be considered an *enactive literature review,* as it synthesizes and makes manifest the dynamics of the existing data on all related systems. However, the model is radically different from a usual literature review, as it has four novel elements: a reference element, where the model incorporates data from different real systems; an integrative element, where data from these real systems are synthesized; a performative element, where it acts out the behaviors related to the data and the systems as a whole; and a counterfactual element, where it acts out behaviors not seen in the wild or in the prototype, based on the modeler's manipulation of parameters. These elements and their enactive synthesis, where data are integrated to act out real and counterfactual behaviors, exist only in the model. The iterative processes of building the enactive system thus: a) brings the generic model (built from a variety of data) closer in behavior to the (built) target model, and b) helps create a new class of enactive systems, of which the real-world system and the built target model are instances.

2. The new class of enactive systems allows new concepts and control structures generated for the generic model to transfer to the built prototypes, either in an applied fashion (case 1), or as a confirmation of a prediction (case 2).
3. The gradual process of building creates a close coupling between the model and the modeler's imagination (mental simulation), such that the computational model gradually develops into an external component of the imagination system. This coupling significantly extends and enhances the researcher's natural capacity for simulative model-based reasoning (see Chandrasekharan 2014, for cognitive and neural mechanisms that could support this extension.) Inferences leading to new concepts and control structures are made through, or by means of, the this coupled system.

There are two main implications of our proposals: (1) correspondence is not a fixed structural relation in the engineering sciences, but needs to be understood as an evolving and interactive process, and (2) understanding how the building process leads to novel inferences requires a distributed cognition approach to cognitive processes. We discuss these in turn.

6.1 The plasticity of correspondence

The *achievement* of correspondence is the key to how discoveries are made using models. To explain how correspondence is achieved between computational models and target prototype models, the building process needs to be incorporated into the account of correspondence. There is no one point where the correspondence between model and the real-world is final, as 'fit' constantly evolves as new data are added to the model. The correspondence is at the behavioral and systems-level, and not at the structural and elemental level. Also, it is acquired incrementally over many iterations of infusing and fitting data.

There are many arbitrary elements in the building processes, including: choice of modeling platform, initial selection of data, simplifying assumptions, choice of replications, and the modeler's intuition. Bits of theory are also opportunistically incorporated into the building of the model. This process is thus quite different from the situation that is customarily discussed, where models are in large part derived from theory, and are designed to apply or test it. In the kind of modeling we report, there is no theory of the domain. Rather, numerous theoretical elements are inserted into the models, such as kinetic orders, time lags that take into account the translation-transcription process, effects of cell boundaries, probability distributions, mechanical forces, and so forth. If the problem demands it, or if the approach seems promising, modelers will use any or all of these together, regardless of their origins and history. The bioengineering modelers we have been studying thus build their models like a bird builds its nest—it will use anything that suits the purpose of having a stable and comfortable place for eggs (plastic, twigs, cloth, string, wires, paper, cotton, leaves.).

One way to understand why this approach works is to consider the model's role as an enactive system, which reveals dynamic patterns when run. This role could be considered similar to the one played by stains and casts, which reveal static patterns. There are some general properties that the stains and the casts need to follow (such as they should not change/damage the cell/organ), but their structure can be arbitrary within these constraints. In the case of computational models in bioengineering sciences, the pattern they seek to reveal are configural patterns in time, which exist only in interaction—such as the way many limb elements interact together to create walking. Such patterns generated by the model are first compared with actual patterns (in the real systems and prototypes), first in a qualitative way, and then in a more quantitative way. As long as the comparisons are roughly equivalent, and the generated patterns provide insight for further building of the prototype, the 'stain/cast' elements of the model do not need to be justified.

In such modeling, theory is just handy and mathematically tractable elements (with some scientific justification in relation to their problem), which can be inserted into the model to make it fit the given data. The connections between the different theoretical elements are not clear, and the connections exist only when the model runs, and these connections can be different for different runs. The arbitrary elements are mitigated through the iterative fitting process that adds real-world data and complexity to the model. This process is stopped when the model fits all the available data well, which is considered as an achievement of correspondence. It then is a generic model (but not a general model, as the solutions are numerical, and not analytic) that has the ability to enact behavior common to the class comprising the real systems, the model, and the prototype. The discovery in the computational model can then be used to change the target system to meet some desirable criteria. The re-engineered system is then modeled and the cycle continues. This process is in direct contrast to the cases that start from a set of laws/axioms of a theory, and analytical solutions, which apply to the class.

A similar process account of biomedical modeling is provided by Carusi (2014). She argues that the process of establishing a reliable model ensemble may rely on an intensive process of parameter fitting and experimentation, which, according to Carusi, serves to establish "comparability between the variability in the population of models, and that in the experimental data set" (Carusi 2014, p. 33) through something like mutual exploration of the systems and the model. Validation grounds are not externally given, but internally established during the model-development process, as researchers learn how the system and model are related. As such, on her account, the means by which correspondence is established are variable and need to be established during the model-building process.

Since both the model and target are built systems in the bioengineering domain, they focus on phenomena that are amenable to control, and the objective of the computational model is to identify variations and parameter combinations that help develop a control structure that optimizes some desirable feature in the built prototype. In other words, in the bioengineering sciences, and engineering sciences in general, the transfer of a new concept or new understanding derived from a built computational model usually involves developing a control structure that optimizes performance of the physical prototype.

This engineering notion of transfer involves reconfiguring existing variations in the real-world prototype, to develop behavior that was not possible previously. This is easier than showing that a new behavior or element identified by the computational model exists in the natural system, though this latter type of transfer also occurs, as illustrated by our case 2, as well as the case outlined by Lenhard (2004). This type of transfer to the larger natural system is more difficult to explain, since only a selected part of that system has been implemented by the prototype. Our enactive system account provides insight into, but does not fully address, how this transfer from model to the natural system occurs. We hope to address this problem in future work.

As we noted at the outset, the recent philosophical literature on computational modeling acknowledges that models which are based on established theory necessarily introduce arbitrary elements in the model-building process, leading to the conclusion that even when "simulation is the offspring of theory… it is a mongrel offspring" (Winsberg 2003, p. 108). In the kind of engineering sciences research we are investigating, models incorporate theoretical elements, but they are by no means the "offspring of theory." Fox Keller (2003, 2009), distinguished the strategy of "modeling from above" (characterizes how cellular automata models are built in the absence of theory to be "imitative" of phenomena) from "modeling from below," which relies on information about causal structure and dynamics of the systems' compositional elements. We contend the imitative goal comprises the full range of modeling approaches—physical to computational—used in building models to simulate phenomena. Lenhard has also argued that even in theory-based modeling the discrete model is not simply an "aid for the numerical computation of solutions to the theoretical model" (2007, 177), but has an imitative goal. He, too, maintains that the imitative goal of Keller's "modeling from above" is a "more general characteristic of computer simulations" (2007, 192). His analysis of model building in climate research in the 1960s details iterative aspects of the fitting process that introduce arbitrary elements (such as, "Arakawa's trick") in order to make the model imitate the known phenomena. Additionally, MacLeod and Nersessian (2013) examine how integrative systems biologists, lacking a systematic theoretical starting point, iteratively build models imitative "from the ground up" by collecting dynamical and structural information from a variety of sources, including empirical data, bits of theory, and their own simulations.

Our cognitive ethnographic approach provides a complimentary view, investigating model building practices, providing rich data for detailing and further specifying how modeling builds correspondences with target phenomena. Unlike climate models and other similar cases, where the target system is considered fixed, the cases we report involve building computational models of engineered (and engineerable) prototypes that are themselves being evolved to meet some desirable criteria. Here correspondence between the model and the real system is highly fluid and dynamic, and this shifting correspondence can develop from both directions (model →prototype, prototype →model), with changes made in both systems, and integrated, to generate a discovery. The model is thus a space for exploring the manipulation possibilities of the target system, which, from the outset, is considered amenable to redesign, and hence 'plastic'. In this view, the parameter values, and even components, in the natural system are contingent. When the modeled systems are assumed to be plastic, and reality is considered just one contingent instantiation of the parameter space, the ontology itself is plastic, and it may thus not be feasible to have a robust notion of truth in such engineerable domains.

6.2 The built imagination

Our data provide insight into the cognitive dimension of modeling: how the model-building processes support inferences that lead to conceptual innovation and novel hypotheses about the phenomena (how to control prototypes, missing pathway elements). To understand how the building process leads to new knowledge, we need to move away from cognitive models that focus exclusively on individual cognitive processes, to distributed cognition models that study the close coupling between individual cognitive processes and external processes with representational features. Even within distributed cognition, we need to move beyond the standard analyses which focus on the *use* of external representations, to the *building* of such representations, and how this building process changes the agent's cognitive system and its relation to the external model (Chandrasekharan and Nersessian 2015; Chandrasekharan 2009). As we have argued, the building process leads to a gradual development of twin changes (enaction ability, broader imagination space), and this gradual process creates a close coupling between the modeler's imagination (mental simulations of target phenomena) and the variations made possible by the computational model, such that the model acquires the role of an external imagination system, supporting the generation of novel and fine-grained scenarios that would be impossible to create in the mind. This externalized imagination has the ability to generate more detailed scenarios than the internal imagination, and its gradual building creates a systematic relation with the internal imagination, such that the behaviors generated in the external model are immediately incorporated into the internally imagined models of possible mechanisms underlying the behavior (for detailed accounts, see Chandrasekharan 2009, 2016; Chandrasekharan and Nersessian 2015; Rahaman et al. in press).

Humphreys (2004), too, has argued that we "extend ourselves" in developing new computational methods and simulation possibilities. However, there is a significant difference between the *tool* view of extension he posits, and our own *coupled system* view. On the tool view, scientists extend their sensory and cognitive capacities by creating instruments (telescopes, microscopes) and new analytical tools (models, mathematics). The tool view emphasizes the new operations that a *final* model (tool) allows the modeler to perform, such as fast numerical solutions to complex equations. The process of building the tool is not considered part of the extension. On the coupled system view, scientists create cognitive powers gradually, both by building new environments (Hutchins 1995), and *also through the building process* (Chandrasekharan and Nersessian 2015; Chandrasekharan 2009; Nersessian 2012a). The new cognitive power we have been calling "external imagination" is the power to simulate complex dynamical phenomena, which extends the limited human cognitive capacities for mental simulation (Nersessian 2009b; Chandrasekharan 2014, 2016). But we argue that this extension is a gradual, systematic and iterative process that develops in step with the process of building the model, eventually leading to the 'incorporation' (Chandrasekharan and Nersessian 2015; Chandrasekharan 2014) of the model into the modeler's imagination system. This view is not incompatible with the tool view—particularly the notion of models as epistemic tools developed by Knuuttila (2005)—but focuses attention closely on the process of building and incorporating the tool, instead of the final product.

Nersessian (1992, 2002, 2008) has argued that the simulation capacity underlies the practices of thought experimenting and making dynamical inferences from static diagrammatic representations (see also Gooding 2004). Inferences, such as those leading to new concepts, are made through the coupled system of mental and external representations. But mental simulations coupling with diagrams (using pen and paper tools) are much more limited than couplings where the external representation has enaction capabilities (Chandrasekharan et al. 2012; Chandrasekharan 2009, 2016). The enaction ability (computational simulation) brings with it a range of possibilities not previously available to scientists. The following is a sample set:

- It allows running many more simulations, with many variables at gradients not perceivable and manipulatable by the mind (say .0025 of metabolites a and b). These can then be compared and contrasted, which would be difficult to do in the mind.
- It allows testing what-if scenarios that are difficult to do in the mind. Such as, what would happen if I change variable 1 and 2 downwards, switch of 6 and 21, and raise 7 and 11 with a time lag between 16 and 19?
- It allows stopping the simulation in between, and checking its innards, or tracking its innards at every time point, and if something desirable is seen, tweaking the variables to get that more often and consistently. This 'reverse simulation' is impossible to do in the mind.
- It allows taking different parts of the system as modules, simulating them, and putting them together in different combinations.
- It allows changing the time at which some in-between process kicks in (say, making it start earlier or later), again a process very difficult to do in the mind.

It is worth noting that these manipulations are also not possible in experiments. Building the model thus changes the manipulation landscape, as well as the control landscape. A distributed cognition approach is necessary to understand how these changes reshape the cognitive capacities of the scientist. The (gradually acquired) manipulation and control abilities change what is available to the mind and what it can do with it. Our significant finding in case 2, for instance, is that one of the things the mind does with the new configurations and manipulative abilities is to ask the question: what happens if the very representational mapping of the model is changed? This question can be asked only after the modeler has created lots and lots of variation, and every possible change other than this one fails to provide a good fit. It is a very bold move, especially for a novice in the domain, and the building process (and the variations seen during it) provides the novice researcher with warrant (and confidence) to make this bold proposal.

7 Rethinking correspondence

The two cases we have outlined provide an early view of an emerging method in engineering sciences, one based on the following spiral: (1) building prototypes and computational models, (2) gaining insights by building and exploring many numerical solutions and scenarios, and (3) instantiating these insights in further prototypes. An understanding of the natural phenomena (that inspired the prototype) emerges from many iterations of this spiral. But this understanding is a half-way house, and provisional, as the existing natural system is viewed as just one possible prototype, and the objective is to engineer it in desirable ways. In this spiral approach, the correspondence between model and the world is plastic (dynamic and always evolving), with no final mapping or limit, as every engineering feat leads to further engineering possibilities. Our analysis in this paper provides a starting point to develop an account of the critical role played by the model construction process in this emerging scientific method, and the philosophical implications that follow from this understanding of scientific cognition—as developing in relation to *the process of building* models.

In our account, discoveries arise from the coupling between the modeler and the model, and this coupling gradually develops through the building process. The model's ability to enact a range of real-world behavior, and also counterfactual scenarios, leads to the gradual coupling between the modeler's imagination and the model, and discoveries emerge from probing this coupled system. In this view, the model acquires the nature of a probe through its ability to generate coherence between different experimental results, and new discoveries emerge from exploring and widening this coherence. The modeler's cognitive capacities are thus extended through coherence. This view provides a central place for coherence in the context of discovery, apart from the role coherence plays in theories of justification. This dual role makes coherence more attractive as a theoretical criterion, compared to correspondence.

The cases we present illustrate that the computational model generates knowledge not by accurately describing the target system, but by enacting, and thus revealing, its system-level behaviors through the building process. When knowledge emerges this way, from an enactive process (the model) mimicking another enactive process (the target system), through a third enactive process (the modeler's cognitive system) tracking the coherence between the two and making changes to both, correspondence—a static, linguistic, and possibly typographic (see Ong 1982; Chandrasekharan 2016) notion —does not seem the right metaphor to capture the connection between the model, the target system and the modeler. Resonance (or entrainment) would be a more process-oriented, realist, and integrative metaphor (Chandrasekharan 2014). The implications of shifting to such a process view of the relationship between models, modeling and the world, particularly the built world, are worth exploring, particularly given the current shift towards enactive and process theories in cognition and the philosophy of mind.

Acknowledgements The theoretical account presented here owes a great deal to questions from participants in the Advanced Topics in Cognition course offered at the Homi Bhabha Centre for Science education, TIFR. This research was supported by US National Science Foundation Grants (DRL0106773, DRL0411825, DRL097394084) to Nancy J. Nersessian (PI) and Wendy Newstetter (co-PI), and a grant to Sanjay Chandrasekharan (PI) from the Cognitive Science Initiative, Department of Science and Technology, Government of India. We thank the members of the Cognition and Learning in Interdisciplinary Cultures research group who worked on this extended research project, especially Lisa Osbeck, Christopher Patton, and Miles MacLeod. We thank also the researchers in the labs for being generous with their time in letting us observe and interview them about their research.

References

Ankeny, R. A., & Leonelli, S. (2011). What's so special about model organisms? *Studies in History and Philosophy of Science Part A*, *42*, 313–323.

Carusi, A. (2014). Validation and variability: Dual challenges on the path from systems biology to systems medicine. *Studies in History and Philosophy of Science Part C: Studies in History and Philosophy of Biological and Biomedical Sciences.*, *48*, 28–37.

Chandrasekharan, S. (2009). Building to discover: A common coding model. *Cognitive Science*, *33*(6), 1059–1086.

Chandrasekharan, S. (2014). Becoming knowledge: Cognitive and neural mechanisms that support scientific intuition. In L. Osbeck & B. Held (Eds.), *Rational intuition: Philosophical roots, scientific investigations*. New York: Cambridge University Press.

Chandrasekharan, S. (2016). Beyond telling: Where new computational media is taking model-based reasoning. In L. Magnani & C. Casadio (Eds.), *Model-based reasoning in science and technology: Logical, epistemological, and cognitive issues*, Volume 27 of the series Studies in applied philosophy, epistemology and rational ethics (pp. 471–487). Heidelberg: Springer.

Chandrasekharan, S., & Nersessian, N. J. (2015). Building cognition: The construction of computational representations for scientific discovery. *Cognitive Science*, *39*, 1727–1763.

Chandrasekharan, S., Nersessian, N. J., & Subramanian, V. (2012). Computational modeling: Is this the end of thought experiments in science? In J. Brown, M. Frappier, & L. Meynell (Eds.), *Thought experiments in philosophy, science and the arts* (pp. 239–260). London: Routledge.

Chandrasekharan, S., & Stewart, T. C. (2007). The origin of epistemic structures and proto-representations. *Adaptive Behavior*, *15*(3), 329–353.

Dennett, D. (1992). The self as a center of narrative gravity. In F. Kessel, P. Cole, & D. Johnson (Eds.), *Self and consciousness: Multiple perspectives*. Hillsdale, NJ: Erlbaum.

Ferguson, E. S. (1992). *Engineering and the mind's eye*. Cambridge, MA: MIT Press.

Fox Keller, E. (2003). Models, simulation, and computer experiments. In Hans Radder (Ed.), *The philosophy of scientific experimentation* (pp. 198–215). Pittsburgh: University of Pittsburgh Press.

Fox Keller, E. (2009). Organisms, machines, and thunderstorms: A history of self-organization, part two. complexity, emergence, and stable attractors. *Historical Studies in the Natural Sciences*, *39*(1), 1–31.

Gelfert, A. (2009). Rigorous results, cross-model justification, and the transfer of empirical warrant: The case of many-body models in physics. *Synthese*, *169*(3), 497–519.

Gooding, D. C. (2004). Cognition, construction and culture: Visual theories in the sciences. *Journal of Cognition and Culture*, *4*(3), 551–593.

Gramelsberger, G. (2013). The simulation approach in synthetic biology. *Studies in History and Philosophy of Science Part C: Studies in History and Philosophy of Biological and Biomedical Sciences*, *44*(2), 150–157.

Green, S. (2013). When one model is not enough: Combining epistemic tools in systems biology. *Studies in History and Philosophy of Science Part C: Studies in History and Philosophy of Biological and Biomedical Sciences*, *44*(2), 170–180.

Humphreys, P. (2004). *Extending ourselves: Computational science, empiricism, and scientific method*. Oxford: Oxford University Press.

Hutchins, E. (1995). *Cognition in the wild*. Cambridge, MA: MIT Press.

Ihde, D. (2000). Epistemology engines. *Nature*, *406*(6791), 21–21.

Knuuttila, T. (2005). *Models as epistemic artefacts: Toward a non-representationalist account of scientific representation*. PhD. Dissertation. Philosophical Studies from the University of Helsinki 8. Vantaa: Edita Prima.

Knuuttila, T., & Loettgers, A. (2013). Basic science through engineering? Synthetic modeling and the idea of biology-inspired engineering. *Studies in History and Philosophy of Science Part C: Studies in History and Philosophy of Biological and Biomedical Sciences*, *44*(2), 158–169.

Lenhard, J. (2004). Surprised by a nanowire: Simulation, control, and understanding. *Philosophy of Science*, *73*, 605–616.

Lenhard, J. (2007). Computer Simulation: The Cooperation between Experimenting and Modeling. *Philosophy of Science*, *74*(2), 176–194.

Loettgers, A. (2009). Synthetic biology and the emergence of a dual meaning of noise. *Biological Theory*, *4*(4), 340–356.

MacKenzie, D. (2008). *An engine, not a camera: How financial models shape markets*. Cambridge, MA: MIT Press.

MacLeod, M., & Nersessian, N. J. (2013). Building simulations from the ground-up: Modeling and theory in systems biology. *Philosophy of Science*, *80*, 533–556.

Nersessian, N. J. (1992). In the theoretician's laboratory: Thought experimenting as mental modeling. *PSA Proceedings of the Biennial Meeting of the Philosophy of Science Association*, *2*, 291–301.

Nersessian, N. (2002). The cognitive basis of model-based reasoning in science. In P. Carruthers, S. Stich, & M. Siegal (Eds.), *The cognitive basis of science* (pp. 133–153). Cambridge, MA: Cambridge University Press.

Nersessian, N. J. (2006). The cognitive-cultural systems of the research laboratory. *Organization Studies*, *27*, 125–145.

Nersessian, N. J. (2008). *Creating scientific concepts*. Cambridge, MA: MIT Press.

Nersessian, N. J. (2009a). How do engineering scientists think? Model-based simulation in biomedical engineering research laboratories. *Topics in Cognitive Science*, *1*, 730–757.

Nersessian, N. J. (2009b). How do engineering scientists think? Model-based simulation in biomedical engineering laboratories. *Topics in Cognitive Science*, *1*, 730–757.

Nersessian, N. J. (2012a). Engineering concepts: The interplay between concept formation and modeling practices in bioengineering sciences. *Mind, Culture, and Activity*, *19*, 222–239.

Nersessian, N. J. (2012b). Modeling practices in conceptual innovation: An ethnographic study of a neural engineering research laboratory. In U. Feest & F. Steinle (Eds.), *Scientific Concepts and Investigative Practice* (pp. 245–269). Berlin: DeGruyter.

Nersessian, N. J., & Chandrasekharan, S. (2009). Hybrid analogies in conceptual innovation in science. *Cognitive Systems Research Journal: Special Issue: Integrating Cognitive Abilities*, *10*, 178–188.

Nersessian, N. J., Kurz-Milcke, E., Newstetter, W., & Davies, J. (2003). Research laboratories as evolving distributed cognitive systems. In D. Alterman & D. Kirsch (Eds.), *Proceedings of the Cognitive Science Society* (Vol. 25, pp. 857–862). Hillsdale, NJ: Lawrence Erlbaum Associates.

Nersessian, N. J., & Patton, C. (2009). Model-based reasoning in interdisciplinary engineering. In A. Meijers (Ed.), *Handbook of the philosophy of technology and engineering sciences* (pp. 727–757). North Holland: Elsevier.

Nordmann, A. (2015). Synthetic biology at the limits of science. In B. Giese, C. Pade, H. Wigger & A. von Gliech (Eds.), *Synthetic biology: Character and impact* (pp. 31–58) Springer.

O'Malley, M. A., & Soyer, O. S. (2012). The roles of integration in molecular systems biology. *Studies in History and Philosophy of Science Part C: Studies in History and Philosophy of Biological and Biomedical Sciences*, *43*(1), 58–68.

Ong, W. J. (1982). *Orality and literacy: The technologizing of the word*. London: Methuen.

Parker, W. S. (2013). Ensemble modeling, uncertainty and robust predictions. *Wiley Interdisciplinary Reviews: Climate Change*, *4*(3), 213–223.

Rahaman, J., Agrawal, H., Srivastava, N., & Chandrasekharan, S. (in press). Recombinant enaction: Manipulatives generate new procedures in the imagination, by extending and recombining action spaces. *Cognitive Science*. doi:10.1111/cogs.12518.

Voit, E. O., Qui, Z., & Kikuchi, S. (2012). Mesoscopic models of neurotransmission as intermediates between disease simulators and tools for discovering design principles. *Pharmacopsychiatry*, *45*(1), 22–30.

Winsberg, E. (2003). Simulated experiments: Methodology for a virtual world. *Philosophy of Science*, *70*, 105–125.

Winsberg, E. (2010). *Science in the age of computer simulation*. Chicago: University of Chicago Press.