Methodological Considerations of Heuristic Modeling of Biological Systems

Glen E. P. ROPELLA[1,2], C. Anthony HUNT[2], and Shahab SHEIKH-BAHAEI[1,2]

[1]Biosystems Group, University of California, [2]Tempus Dictum, Inc.
513 Parnassus Ave., San Francisco, CA 94143-0446
gepr@tempusdictum.com hunt@itsa.ucsf.edu shahab@tempusdictum.com

ABSTRACT

We present the basics of a modeling method, FURM (Functional Unit Representation Method) that attempts to address a fundamental difference in the practice of biological modeling: that of creating experimental, constructive analogs versus inductive, mathematical data fitting. Further, we briefly present two modeling frameworks built according to FURM as concrete examples of how the method can be used.

Keywords: synthetic method, heuristic models, systems biology.

1. INTRODUCTION

In this paper we make the case that, in order to make real progress in understanding how molecular and physiologyical components work together as robust systems, we need to engage in the hard work of constructing and experimenting with in silico models that can, through simulation, exhibit the system-level behaviors of interest. Before developing our case, however, we need to carefully revisit the concepts of modeling and simulation.

The etymology of the word “model” generally indicates that it is closely related to “measure.” Taken in that sense, a model can be thought of as a device, an engineered device, against which to measure an artifact or phenomenon. What, then, are we doing when we model some process, artifact, or system? What are we trying to achieve? How will the model be compared to the referent? What aspects of the referent are observable? Which of the observables are quantifiable and which are qualitative? The answers come directly from the contexts in which the engineered device will be used. The usage tells us what the model is good for, what it must do, whom it must serve, for what experiments it will be used, how long it must be useful, etc.

Closely examining the usage elicits the requirements and desirable characteristics of the device. Often, however, it is very difficult to delineate the salient aspects of biological systems, because they have a dense continuum of behaviors and situations in which they are relevant[1]. This forces modelers to choose, somewhat arbitrarily, a small subset of the potential characteristics their model might have. One consequence is that the modeler is now open to the rhetorical fallacy of *petitio principii*, which in modeling becomes the fallacy of “inscription error” [17].

Given that any modeling effort may be faced with a huge space of relevant characteristics, how does a modeler choose which are salient? The answer depends on the core context in which the model will be used. That context provides the impetus for model development, sets its expected lifetime, and defines the actors involved in its development and usage. A key property of the core context lies in the modeler's ability to collapse it into a detailed enough prescription for what the model must do. This reduction of the context is particular to the modeler, implying that the model is developed through subjective interpretation of the context. Hence, modeling is a subjective enterprise. Modeling is epistemological and cognitive. It is not ontological.

There are basically two categories of models as they are used in biology laid out in Fig 1. We place model organisms, such as specialized strains of mice, in the upper right corner of the sketch, and mathematical or statistical models in the lower left corner. The sketch illustrates a gap between the experimental models used to clarify or test hypotheses and computational models, which are designed to closely fit or precisely characterize their referents. This gap sits on a continuum between the two extremes of experiment and theory, where the former is more exploratory, aimed at discovering various distinct characteristics, and the latter is more exploitative, aimed at precisely honing specific characteristics.

To make the desired progress in understanding biological systems we need a method that can fill this gap and pro-

[1] In fact, there is a well-known metamathematics theorem that formalizes this practical difficulty, Tarski's Theorem, which states that “assuming that the class of all provable sentences of the metatheory is consistent, it is impossible to construct an adequate definition of truth in the sense of convention T on the basis of the metatheory” [19]. Convention T is simply a rigorous definition of truth. This theorem, with some additional consideration of the universality requirement for Turing machines, leads von Neumann to claim that, for particular types of systems (including biological systems), their description will be indefinitely long [22].

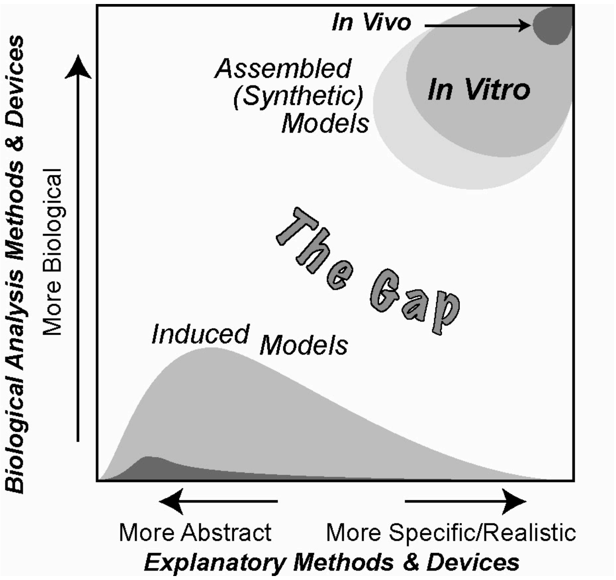
In Vivo
Assembled (Synthetic) Models
In Vitro
The Gap
Induced Models
Biological Analysis Methods & Devices
More Biological
More Abstract
More Specific/Realistic
Explanatory Methods & Devices

Figure 1. Relative properties of the two major classes of models used in biomedical research. Model types are arranged according to abstraction level versus biological character. The experimental models used for laboratory research are confined to the upper shaded area; computational and mathematical models are confined to the lower shaded area. Assigned to the upper region are model organisms (darker shading); in vitro cell, tissue, and organ cultures; and standardized cell forms. Assigned to the lower region are statistical (darker shading) and mathematical models

duce new classes of models. The methodological considerations below attempt to directly address the issues faced when trying to work within this gap, making computational models more experimental and vice versa, making experimental models more exploitative.

2. CONSTRUCTION METHODS FOR EXPLORATION AND EXPLOITATION

The usage contexts for models or model types along the exploration-exploitation continuum are distinct, but not mutually exclusive. In fact, they are vague and open to interpretation. The distinction relevant for this discussion is the degree to which the modeling device is separate from the particularities of the modeler and, hence, the particular intent of the model when it is created.

In the case where a model is **induced** by abstracting characteristics from a collection of objects, data sets in this context, that model amounts to a membership function that defines a class of objects. That membership function can be applied to any particular object to determine whether it is a member of that class. This is the inductive method [18]. It ties the model, inextricably, to the semantics of membership functions. Objects that are presented are interpreted in accordance with that function. In contrast, when a modeling device is **assembled** out of pre-existing components the resulting device is not necessarily a coherent membership function but a post-hoc encoding of the various functions of its constituent components. This is the synthetic method[2] [18].

It is instructive to contrast the two methods. In most cases, experimental models are synthetic and computational models are inductive. The essential difference is that the inductive method explicitly uses the observed phenomena (the data) as its input whereas the synthetic method starts with proposed building blocks and their interrelations. The inductive method takes the ontology defined implicitly in the data as an inherent assumption whereas the synthetic method constructs an ontology (and an analog that "lives" in that ontology) that can realize the data.

Another useful way of thinking of the distinction is to consider a mapping from the space of generators or mechanisms to the space of phenomena. The inductive method starts with the phenomena and works backward to generators. The synthetic method, in contrast, works forward from generator to phenomenon. When viewed from this perspective, it is clear that the methods are not mutually exclusive, but are two sides of the same coin.

Contrasting these two methods helps determine how a model can best be put to use. Induced models hone certain characteristics to a precise predicate against which any new situation can be evaluated. Synthetic models precisely specify their constituents and leave the conesquences resulting from constituent interactions to be discovered through experiments. This implies that induced models are ideally suited for exploiting discovered characteristics; and synthetic models are ideally suited for exploring the consequences of assembled components. With this in mind, we return to the original questions: What are we doing when we model something? How will the model be used? How do we purposefully choose a modeling method that achieves our ends?

3. INTERPOLATION VS. EXTRAPOLATION OF CONTEXT

Most models in the lower half of Fig. 1 are inductively defined mathematical compressions of the phenomena they are intended to model. This restricts their predictive capabilities to situations that are members of that inductively defined set, i.e. variations of the original experiment that preserve the induced parameterization and input/output characteristics. The induced model can only make predictions about other members of that class of phenomena or experiments. This is a kind of context interpolation. In contrast, synthetic models present a complex or plectic[3] [5] set of characteristics. The sys-

[2] The words "synthetic" and "constructive" are synonymous. We somewhat arbitrarily restrict "synthetic" to indicate "assembled from pre-existing parts" just for consistency.

[3] Plectics refers to the study of systems where the degree of entanglement of their constituents is, itself, an object of study.

temic characteristics presented might even be incoherent or highly nonlinear in some regions of the model behavior space. Because the context in which the synthetic model is used is not likely to comply with all the components' requirements and because the conditions of its usage are compounded by stacking component requirements atop one another, this results in a kind of context extrapolation.

4. THE COMPLETENESS OF A REPRESENTATION

Robustness is the primary failure of biological models we expect to address through synthetic modeling. Biological models are fragile to the context that drives their development. For example, an ODE model formulated to replicate data from specific experiments is rarely applicable to another set of experiments without reformulation. That reformulation may be minor (adjustment of parameter values) or major (addition of new terms or expansion to a system of equations); but, it is reformulation. The old model is considered inadequate because it is a lossy compression of the data from the first set of experiments. It did not capture the generative mechanism of the biological system well enough to be reused to account for the new data. All such models are "broken" or **incomplete** in some deep sense. They obviously do not model their referents very well.

The fact that current biological models are impoverished in this way is well-known but often overlooked. The point was raised by Dr. Sydney Brenner at the NIH BISTI 2003 Symposium [2]: "The man in the street doesn't believe in evolution, because he says we're trying to tell him you can take a black and white television set [and] make random mutations in it that will turn it into a colored television set. He knows if you tamper with a television set, the most likely thing is you'll break it. So how do biological organisms not get broken all the time? There must be an architecture within the way they are constructed logically and actually which makes them resistant to this and allows evolution to proceed. And that will be a feature of what we're looking at. It will be there; it will be underlying everything that we do."

Discovering and understanding this robust architecture is the goal of heuristic biological models. Synthetic modeling is one technique for building such models. In that same keynote address, Dr. Brenner also makes the connection between the synthetic method and discovering this robust architecture by pointing to von Neumann's constructive proof of self-reproducing automata. Constructive proofs are inherently heuristic. And like self-reproduction, this robustness to context changes or contextual extrapolation seems to be an inherent component of biological systems.

5. THE CASE FOR A NEW METHOD

Of course, the problem with constructive proof and, by extension, the synthetic method as a whole, is coming up with devices that successfully generate the behavior sought. What is the best means for doing that? The way most followed today is to seek methods that help build better models. Is it better models that we need or is it methods for facilitating the building of models, a wider variety of models made easier? We suggest that the field may benefit more from the latter. When we target the building of models, the focus shifts from the model as an endpoint to the process of model building. The more models that we can build and evaluate, and the easier that process becomes, the more likely we are to find plausible mechanisms underlying the robust architecture of biological systems.

We have attempted to lay the foundations for an argument for a new modeling method. Carrying forward from the premises that:

- effective models are developed to stay close to their context;
- models are subjective, psychological artifacts;
- heuristic models are constructive analogs of their referents;

we can infer that modeling efforts (especially of complex biological referents) should be "thought experiments" where models are constructed from building blocks and evaluated experimentally against some criteria defined by the usage context.

We are building a modeling method, tools that support the method, clinically relevant models, and biomimetic devices that adhere to this method. These devices can fill an important gap in how models are used to study biological systems. The gap is one of **exploratory modeling** in the tradition of von Neumann's automata [22], Ulam's description of "experiments in theory" [20], monte carlo methods [9], extensions to more modern efforts like the "opaque thought experiments" of Di Paolo [4], Mark Tilden's BEAM robots and Danny Hillis' Tinker Toy computer [3], Gerald Edelman's Darwin automaton [8], and even extensions to some methods like that proposed by Steven Bankes [1]. These efforts are united in the emphasis they place on the synthesis of and experimentation with models.

6. FURM

Biologists tend to see biological systems as comprised of functional units at some selected level of resolution. To match this tendency, we center our method around the development of functional unit analogs. We call it the Functional Unit Representation Method (FURM).

As Dr. Gell-Mann explains, it is preferable to "complex" and "simple" because those words carry unwarranted implications about the systems being studied. Using "plectic", here, indicates that the degree to which a synthetic system is simple or complex is indeterminate when it is created.

The underlying assumption of unity allows us to effecttively use Object-Oriented (OO) design as the basis for construction. However, OO design does not generalize well into biology because functional units do not have hard boundaries and often exhibit autotelic properties not explicitly addressed by the general OO principles of encapsulation, polymorphism, and inheritance. Specifically, the concept of a biological functional unit entails a degree of purposeful behavior, self-motivation, and local agendas, whereas OO objects more closely fit a pure stimulus-response paradigm. So, we violate some of the strict OO guidelines and adopt Agent-Based Modeling (ABM) techniques where appropriate. We suggest four fundamental guidelines:

R1: standardize interfaces across the models;

R2: use discrete interactions;

R3: design for an extended life cycle; and

R4: define observables to be consistent across models.

These basic requirements flow down into nine more specific requirements that are detailed in Ropella et al 2005 [14].

In addition to facilitating the creation of models, FURM is intended to encourage disciplined experimentation with digital computers, including keeping track of source code as if it were an important experimental material or critical reagent. It is also intended to bring enough rigor to our modeling process to allow us to automate the generation and evaluation of models. Evidence of success in achieving these goals will depend, fundamentally, on whether the models we generate through FURM are successful in their respective domains. For example, the two frameworks described below present us with facilities for evaluating heuristic or predictive hypotheses about pharmacokinetics, drug disposition, and metabolism.

The whole method can be described as follows. Knowledge is acquired directly from experts and domain literature to infer a collection of building blocks and a syntax for plugging those building blocks together. A prototypical experiment or family of experiments is chosen from the literature. Building blocks are assembled to create biomimetic devices that are controlled and measured similarly to the original experiment. Several computational experiments are run with alterations made to the building blocks and their connections. The results of the experiments are compared to the original experimental data and the degree to which the model results fail to match is taken to indicate the implausibility of that model structure.

The behaviors of surviving models can then be explored by experimentation and the results analyzed to test existing hypotheses, provide researchers with new hypotheses, and facilitate the design of novel experiments on the referent systems. We expect the system and method to be a virtual laboratory in which a computational researcher can discover innovative explanations and propositions for how the real biology might give rise to the phenomena embodied in the similarity measures [12].

7. FURM ARCHITECTURE

Requirements 1-4 above provide the skeleton of the architecture. R1 requires all the models, regardless of their internal organization (e.g. equation-based, object-oriented, etc.) or domain (e.g. scale or type of knowledge represented), to adhere to an identical interface. To satisfy R1, we specify that multiple model variants, preferably using varying modeling paradigms, be run simultaneously in a co-simulation framework [21]. As a minimum, we suggest three model types (Fig. 2): the *DatModel*, representing data taken from the model referent, the *RefModel*, a preexisting model that has demonstrated value, and the *ArtModel*, an **articulated** model developed synthetically. To help ensure R1, all models are executed by an experiment agent (*ExperAgent*), which executes the models and takes measurements off them as they run. The observations taken from each model should either be the same or sufficiently similar to be automatically processed by an observer agent to produce the derived measures by which the models are compared.

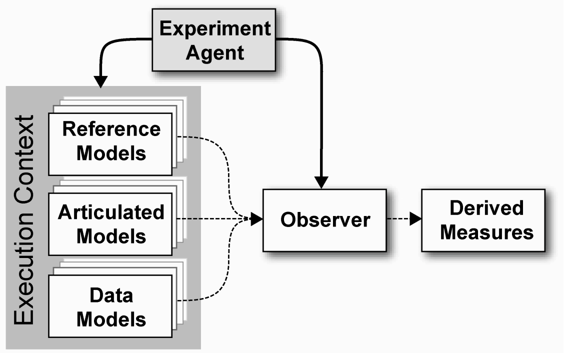

Figure 2. Basic architecture for a FURM-developed framework. The Experiment Agent manages the multiple models, which are measured by an observer, the results of which are used to calculate derived measures.

Co-simulation and model comparison places an indirect constraint on the models. The biggest obstacle in comparing sibling models is discretization. To side-step this issue for practical reasons we include R2 and force all the models to interact with the *ExperAgent* and each other through discrete interactions. This has no implications about the continuity of model internals, however. A model can be internally continuous with the discretization occurring at the model's encapsulation interface. Using discrete interactions allows us to ground the interaction of the various models in a minimal formalism, that of partially ordered sets (posets) of events, without forcing an internal structure on the models. For example, if one model is a discretized continuous system of differential equations and another model is a recurrent artificial neural network, the two models are only required to publish their outputs and sample their inputs in a coherent way in order to interact. The input sampling and output publishing are packaged as discrete events that may or may not impact the evolution of the other models in co-simulation. R2 ensures that all models do this and can interact via discrete interactions *at least* at the level of the *ExperAgent*. Models composed hierarchically are not required to interact with other models via discrete events unless their behavior impacts the *ExperAgent*, which embodies the usage context of the co-simulation framework and the experimental apparatus.

R3 is intended to encourage the modelers to think beyond their immediate problem context. The co-simulation requirement, descending from R1, should encourage the addition and subtraction of models at will, where one model, or a component thereof, can be replaced by another at some later point. R3 is a stronger statement about using "defensive" design deeper within the models and in designing the *ExperAgent* so that it can tolerate new conditions when the experimental context changes. The lifecycle of the models that survive being exercised in the context of the co-simulation framework is expected to grow as they withstand the scrutiny of model comparison. Models that are invalidated, which cannot be made commensurate with their siblings, are thrown out and replaced by newly assembled models. This effectively breaks apart the development of the infrastructure from the development of a particular model and forces modelers to consider the contextual and systemic issues involved in developing a model for a specific experimental framework.

R4 provides the core for measurement of the models and ultimately verification, validation, and accreditation (VV &A) for any model in the framework. The requirement has two basic elements, one focusing on the quantification of model observables and the other focusing on the cross-model consistency of model observables. VV&A are the processes by which a user or community gains trust in a model. R4 establishes the concept of a similarity measure as the source for the development of this trust.

8. EXAMPLE FRAMEWORKS

The two example frameworks, or biomimetic in silico devices (BISDs) [7], demonstrate that the method is practical and can be applied directly to domain-specific research. The In Silico Rat Liver (ISL), was the first to be developed using FURM and stresses the issues raised by the method. The In Silico Hepatic Intrinsic Clearance (IS-HIC) framework demonstrates that FURM generates reusable components, that the functional unit can vary across frameworks, and that experiment protocol can be decoupled from the construction of the models.

In Silico Rat Liver

The In Silico Rat Liver is a BISD built to include physiologically relevant aspects of a liver in the context of the *in situ* liver perfusion protocol detailed in [11]. The experimental data are outflow profiles of fractions of an administered compound. A solution containing the compound is injected into the liver where some is extracted. The ratio of solute flowing out to total solute injected provides the outflow profile. During the process some of the compound may be metabolized. Data for several compounds are available [6]. The liver interacts differently with each compound. In some cases the compounds are extensively metabolized. This data is provided in the ISL *DatModel*, including the capability for interpolating observations when the simulation frequency is higher than the original data set.

The *RefModel* is an accepted reference mathematical model that consists of an ODE derived from idealized vascular and cellular components that calculates drug clearance over time (for details see [6]). This ODE is integrated based on intervals set by the *ArtModel*.

The *ArtModel* is our functional unit model. It is an agent-based model (ABM) whose components reflect physiological components in the liver as determined by liver physiology literature. Space does not permit providing model details (See [7]). However, the primary functional unit being modeled is a liver lobule. A lobule is a collection of sinusoids, which are tubular structures that guide the blood flowing in through the liver tissue responsible for drug extraction. Inside this tissue, the objects responsible for metabolization of the drugs are the liver cells, called hepatocytes. Hepatocytes are only one of the many constituents. The systemic behavior that results in the outflow profile is a function of the collective of these constituents all operating together.

This framework and the *ArtModel* are experimentally interesting because of the high degree of heterogeneity that can be expressed in relation to current state-of-the-art models of hepatic drug clearance. However, since this paper targets methodology, the interesting element, here, is that the phenomena it attempts to represent show some subtleties that are difficult to generate from a purely constructivist method. The outflow profiles can be precisely represented with the above ODE and the liver is a reasonably homogenous organ when healthy. This means that an ODE, which is ideal for large populations that generate aggregate phenomena, is well-suited to the task. However, the ODE doesn't facilitate reduction of the high-level aggregate phenomena down to lower-level particulate phenomena. The liver is an interesting organ to approach with multi-scale methods because although a healthy liver is very homogenous, a diseased liver begins to show some canalizing effects of low-level phenomena.

I.e. the particulars of certain regions or components can become critical to generating the high-level phenomena. Hence the co-simulation of a precisely fit ODE with a very flexible, fine-grained articulated model is a viable method for studying the mapping from generators to phenomena. Results from the simulation along with additional details can be seen in [13] and [7].

In Silico Hepatic Intrinsic Clearance

The functional unit of IS-HIC is the *hepatocyte*, which is one component of the ISL and bears primary responsibility for metabolizing foreign compounds found in the blood. The IS-HIC framework measures the extraction or clearance (*CL*) of compounds (in this case drugs) from the "injected solution" by in silico hepatocytes. Where the ISL models hepatocytes in the context of their spatial and functional roles as a part of a lobule, the IS-HIC arranges the hepatocytes on a 2 dimensional grid as if they were in suspension culture. There is no blood flow and the "hepatocytes" simply sit in the "solution" and act upon "drug molecules" as they come into contact with them.

The *DatModel* represents data obtained from *in vitro* experiments [10], measuring the time course for nine unchanged compounds in cell culture media containing freshly isolated rat hepatocytes. As in the ISL, the *DatModel* can interpolate the data to adhere to the interactions with the *ExperAgent.*

The *RefModel* is an accepted mathematical model that describes *in vitro* hepatic clearance under specific conditions. It is also an ODE and is detailed in [15] based on work presented in [10] and [16].

In the *ArtModel*, hepatocytes are placed randomly in a 2D grid as fixed objects. Drug (and other) mobile objects are then randomly placed within the space (external to hepatocytes) based on the initial concentrations of those compounds. The mobile objects move around pseudo-randomly, while the hepatocytes have the opportunity to "take up" and "metabolize" them. As the simulation progresses, the number of solute objects decreases as a consequence of metabolism. At intervals this number is counted, normalized and scaled to represent the concentration of the compound.

Hepatocytes are very complex cells whose behavior can vary widely depending on their context. We began the IS-HIC partly to help develop these complex components separately from the ISL and partly as a mechanism to build a model of hepatocyte behavior *in vitro*. The aspect of this framework of methodological interest lies primarily in the fact that it is a simpler FURM construction and allows us to demonstrate elements and consequences of FURM that become more difficult with the more complex ISL. It also provides a test-bed for more advanced additions and developments to FURM like automatic selection and optimization of successful models and variations in similarity measures. However, the experimental framework of the IS-HIC is scientifically relevant to the study of hepatocytes and should help posit generative mechanisms at an even smaller scale. Results from IS-HIC experiments run with 3 drugs, Diltiazem, FK1050, and Acetaminophen are shown in [15].

9. SUMMARY

When the physicist Richard Feynman closed the door of his Caltech office for the last time in 1988, he left a striking epigram scrawled on the blackboard: "What I cannot create I do not understand." To better understand how living systems generate their endless variety of robust and fragile behaviors we need to create—synthesize—models that can exhibit some of those behaviors. Synthesizing a system that behaves, in some limited way, like a naturally occurring one stimulates ideas for how that natural system might actually work. We can and will continue to tinker with living systems to explore the consequences of those changes, but we are far from being able to build functioning analogs of whole systems from inert parts. A viable option is to build deep analog models in silico using the synthetic approach described here. FURM provides a method for doing that within a framework that also makes use of the traditional, inductive mathematical models.

The two example products demonstrate that the method is practical enough to produce scientifically relevant models while retaining a pragmatic stance toward modeling formalisms and styles. The ISL demonstrates some of the classical difficulties with modeling and simulation, including over-parameterization, validation, and discretization, and incorporates some of the newer capabilities of agent-based modeling and complex systems. The IS-HIC presents a simpler and, therefore, more tractable problem showing that the method also applies at the smaller scale and facilitates component reuse and validation across models. In addition, the tractability of the IS-HIC presents the opportunity to test more sophisticated methods like automatic model evaluation and optimization.

FURM brings rigor and clarity to the modeling and simulation of biological systems and helps computational scientists think about their modeling devices, referents, tools, and techniques at a higher level. The method is not complete, however. It is only a heuristic intended to guide the computational scientist. It will continue to evolve as we deal with existing difficulties. One such difficulty lies in continuing to bring rigor to in silico experiments. Although FURM facilitates more systematic code and model changes than less structured modeling methods, it is still difficult to separate software engineering from experimentation. Inscription error remains the foremost risk; but, FURM keeps that risk at the surface of every experimental context, which helps to mitigate it.

10. ACKNOWLEDGMENTS

We are grateful for the funding (R21-CDH-00104) provided by the CDH Research Foundation. We thank Michael Roberts not only for the perfused rat liver data, but also for open and useful discussions. We thank the members of the Biosystems Group—the Hunt lab—for assistance, support, and helpful discussions, with specific thanks going to Dev Nag, Li Yan, and Amina Ann Qutub.

11. REFERENCES

[1] S.C. Bankes, J. Gillogly, "Exploratory Modeling: Search through Spaces of Computational Experiments", in **The Proceedings of the Third Annual Conference on Evolutionary Programming,** A.V. Sebald and L.J. Fogel (eds.), World Scientific, 1994.

[2] S. Brenner, Keynote Address, **Biomedical Information Science and Technology Initiative (BISTI) 2003 Symposium** (Bethesda, MD, November 5-7), 2003.

[3] A.K. Dewdney, "Computer Recreations: A Tinkertoy Computer That Plays Tic-Tac-Toe", **Scientific American,** October 1989, pp. 120-123.

[4] E.A. Di Paolo, J. Noble, S. Bullock, "Simulation Models as Opaque Thought Experiments", **Proceedings of the Seventh International Conference on Artificial Life** (Portland, OR), Cambridge, MA: MIT Press, 2000, pp. 497-506.

[5] M. Gell-Mann, "Let's Call It Plectics", **Complexity,** Vol. 1, No. 5, Wiley Periodicals, Inc., 1996.

[6] D.Y. Hung, P. Chang, M. Weiss, M.S. Roberts, "Structure-hepatic Disposition Relationships for Cationic Drugs in Isolated Perfused Rat Livers: Transmembrane Exchange and Cytoplasmic Binding Process", **Journal of Pharmacology and Experimental Therapeutics,** Vol. 297, No. 2, 2001, pp. 780-89.

[7] C.A. Hunt, G.E. Ropella, M.S. Roberts, L.Yan, "Biomimetic In Silico Devices", in V. Danos and V. Schächter (eds.), **Computational Methods in Systems Biology 2004, Lecture Notes in Bioinformatics,** Vol. 3082**,** Springer, 2005, pp. 35-43.

[8] J.L. Krichmar, G.M. Edelman, "Machine Psychology: Autonomous Behavior, Perceptual Categorization and Conditioning in a Brain-based Device", **Cerebral Cortex**, August 12, 2002, pp. 818-830.

[9] N.C. Metropolis, S.M. Ulam, "The Monte-Carlo Method", **Journal of the American Statistical Association,** Vol. 44**,** 1949, pp. 335-341.

[10] Y. Naritomi, S. Terashita, A. Kagayama, Y. Sugiyama, "Utility of Hepatocytes in Predicting Drug Metabolism: Comparison of Hepatic Intrinsic Clearance in Rats and Humans In Vivo and In Vitro", **Drug Metabolism and Disposition**, Vol. 31, No. 5, 2003, pp. 580-588.

[11] M.S. Roberts, Y.G. Anissimov, "Modeling of Hepatic Elimination and Organ Distribution Kinetics with the Extended Convection-dispersion Model", **Journal of Pharmacokinetics and Biopharmaceutics**, Vol. 27, No. 4, 1999, pp. 343-382.

[12] G. Ropella, D. Nag, C.A. Hunt, "Similarity Measures for Automated Comparison of In Silico and In Vitro Experimental Results", **Proceedings of the 25th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC03)** (Cancun, Mexico, September 17-21), Piscataway, NJ: IEEE, 2003, pp. 2933-2936.

[13] G.E.P. Ropella, C.A. Hunt, "Prerequisites for Effective Experimentation in Computational Biology", **Proceedings of the 25th Annual Conference of the IEEE Engineering in Medicine and Biology Society** (Cancun, Mexico, September 17-21), Piscataway, NJ: IEEE, 2003, pp. 1272-1275.

[14] G.E. Ropella, C.A. Hunt, D.A. Nag, "Using Heuristic Models to Bridge the Gap Between Analytic and Experimental Models in Biology", **Spring Simulation Multiconference (SpringSim'05), The Society for Modeling and Simulation International,** 2005.

[15] S. Sheikh-Bahaei, G.E.P. Ropella, C.A. Hunt, "Agent-Based Simulation of In Vitro Hepatic Drug Metabolism: In Silico Hepatic Intrinsic Clearance", **Spring Simulation Multiconference (SpringSim'05), The Society for Modeling and Simulation International,** 2005.

[16] Y. Shibata, H. Takahashi, Y. Ishii, "A Convenient In Vitro Screening Method for Predicting In Vivo Drug Metabolic Clearance Using Isolated Hepatocytes Suspended in Serum", **Drug Metabolism and Disposition**, Vol. 28, No. 12, 2000, pp. 1518-1523.

[17] B.C. Smith, **On the Origin of Objects**, Cambridge, MA: MIT Press, 1996.

[18] L. Steels, R. Brooks (eds.), **The Artificial Life Route to Artificial Intelligence,** Hillsdale, NJ: Lawrence Earlbaum Associates, Inc., 1995.

[19] A. Tarski, "The Concept of Truth in Formalized Languages", in **Logic, Semantics, Metamathematics,** Oxford: Clarendon Press, 1956.

[20] S.M. Ulam, "Computing Machines as a Heuristic Aid", in **A Collection of Mathematical Problems,** New York: Interscience Publishers, 1960.

[21] H. Vangheluwe, J. de Lara, P. Mosterman, "An Introduction to Multi-Paradigm Modelling and Simulation", **Proceedings of AI, Simulation and Planning, AIS'2002,** Lisbon, 2002, pp. 9-20.

[22] J. von Neumann, **Theory of Self-Reproducing Automata**, edited and completed by Arthur W. Burks, Urbana, IL: Univ. of IL, 1966.