

# A Top-Down Approach to a Complex Natural System: Protein Folding<sup>1</sup>

Alan Levin<sup>2</sup>

**Abstract** We develop a general method for applying functional models to natural systems and cite recent progress in protein modeling that demonstrates the power of this approach. Functional modeling constrains the range of acceptable structural models of a system, reduces the difficulty of finding them, and improves their fidelity. However, functional models are distinctly different from the structural models that are more commonly applied in science. In particular, structural and functional models ask different questions and provide different kinds of answers. As we clarify these differences and articulate how to use these models jointly, we extend our ability to do science and gain insight into the proper use of the terms *organization*, *order*, and *emergence* when describing systems in nature.

## 1. Introduction

*“Wholes are functionally prior to parts and parts are operationally prior to wholes. In absence of the other, each is an empty abstraction.”* (Wicken 1987 p. 166)

Organization is a startling and spontaneous macroscopic characteristic—we observe it in nature, and we participate in it socially, economically, culturally, and linguistically. Unfortunately, there appears to be no sound theoretical foundation for organization in nature. Organized behavior is often described as emergent in models from physics, chemistry, and biology. This is a statement that either the underlying structural models do not predict such behavior, or that the modelers did not expect such behavior from their models. A theory of organization, with appropriate models, would more effectively describe and predict the behavior of natural systems. In addition, we would be better able to develop technological applications using organizational motifs from nature.

A key insight (Wicken 1987 p. 40) is that organization is inherently functional. To the extent that a system is organized, it is organized to do something, regardless of how it is structured to do that thing. Order is structural. A system’s structure may be arranged in an orderly or disorderly way, even if it does nothing, e.g. it may be at equilibrium. While we can and must articulate the differences between functional and structural models, our goal is to relate them to one another and demonstrate how they can be used together in the scientific endeavor.

---

<sup>1</sup> Published in Axiomathes online 30 September 2009 DOI: 10.1007/s10516-009-9093-0  
The original publication is available at <http://www.springerlink.com/content/f7j2234104661r65/>

<sup>2</sup> email: ailevin@ix.netcom.com

---

Functional models take a behavioral analytic approach to modeling at the system level; we refer to this as a top-down approach. Structural models take a dynamical synthetic approach to modeling in terms of pieces; we refer to this as a bottom-up approach. Top-down functional models and bottom-up structural models are complementary descriptions of systems as we perceive and measure them in nature. Modern science shuns top-down approaches because of their teleological implications. On the other hand, when we observe a natural system and note that certain components play particular functional roles, this has nothing to do with purpose or design. Our observations are merely system level empirical data. A structural model is neither more real nor more physical than a functional model, and yet we routinely see the functional subordinated to the structural. It is a mistake to assume that everything can be explained from the bottom-up (Anderson 1972; Laughlin and Pines 2000; Rosen 1991; Sauer et al 2007).

In this brief paper, I have two objectives. First, I will develop a general method for applying functional models to natural systems. We will see how functional analysis can extend our ability to do science, that is, how it can extend our ability to model, understand, and explain complex systems in nature. Second, I will demonstrate that state-of-the-art understanding in protein modeling looks astonishingly functional and provides evidence that our extended approach to modeling is an idea whose time has come.

To better understand the differences between structural and functional models and to see how the two types of models complement one another, we must put them into a common context. There are three basic contextual questions that we must answer.

- Epistemic context, “What kinds of understanding do the models provide?”
- Practical context, “What kinds of modeling methods are available?”
- Application context, “How can both models be applied to the same problem?”

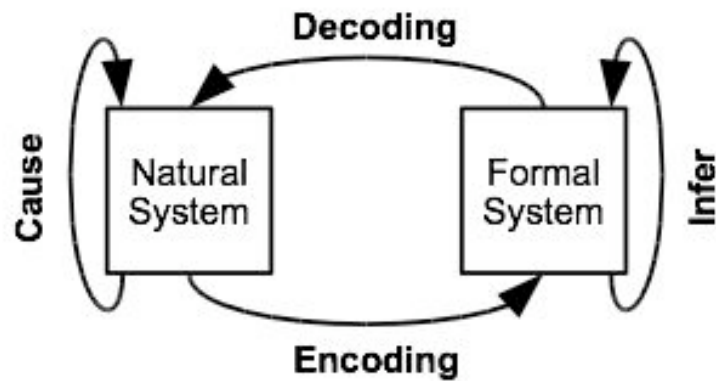
We will use Robert Rosen’s modeling relation (Rosen 1991) to provide the epistemic context. We will then indicate how structural modeling methods used in science and functional modeling methods used in system engineering (Defense Acquisition University 2005) can be used jointly. Finally, we will use protein folding as an application to explore the benefits of complementary functional and structural models.

When we add functional analysis to the usual structural synthesis in scientific modeling, we more effectively model both structure and function in nature. Putting structure and function on a more even footing scientifically, we can more readily distinguish between organization and order. This has important implications in proteomics and beyond.

## **2. Epistemic Context: Rosen’s Modeling Relation**

*“Science, in fact, requires both; it requires an external, objective world of phenomena, and the internal, subjective world of the self, which perceives, organizes, acts, and understands.”* (Rosen 1991 p. 41)

Robert Rosen describes his epistemology and mathematical theory of modeling in *Life Itself* (Rosen 1991). In simple terms, science requires that there is some orderliness in nature and that we are able to recognize it. Rosen articulates these assertions with the modeling relation pictured in Figure 1.



**Figure 1 Rosen's Modeling Relation**

The natural system on the left hand side is the result of selecting a grouping of percepts that we call a system. The formal system on the right hand side is a mathematical model that we create to explain the orderliness that we perceive in nature. We relate natural and formal via encoding and decoding. Encoding is measurement; we assign a formal symbol to a phenomenon or event that we have perceived. Decoding is prediction; we take symbols in the formal system to represent what phenomena or events we expect to measure. We impute causes to the natural system on the left by decoding inferences in the formal system on the right. The formal system is a model of the natural system, and the natural system is a realization of the formal system. To the extent we successfully measure, infer, and predict, we are doing science.

We can now briefly explore structural and functional models as formal systems on the right hand side of the modeling relation, with a view to how they can be applied to the same natural system on the left hand side.

## 2.1 Structural Models

A structural model is motivated by the question, "What is this system made of and how does it work?" The question assumes that the system can be described in terms of smaller and simpler pieces, and that we can synthesize the system model from these pieces. Each piece has both fixed parameters and variables. The fixed or constitutive parameters essentially tell you what kind of piece it is. For instance, the charge, mass, and spin of an electron tell you that it is an electron. Each piece also has a set of variables, its configuration, that can change without altering the nature of what kind of piece it is. The state of each piece is given by the current values of its configuration. For a classical piece, the configuration might be its position and momentum while for a quantum piece the configuration might be a superposition of energy eigenfunctions in the position representation. Finally, each piece has dynamics modeled by differential equations or difference equations, which in general describe how the configuration of that piece changes with time in terms of the constituent parameters. For example this might be the classical equation of motion or the quantum Schrödinger equation.

---

A system level structural model has a series of interesting properties:

- The model has states because we can enumerate the configuration of every piece to articulate the state of the system. In quantum mechanics, wave functions or operator eigenfunctions are used to describe the configuration of the pieces, so the state of the system will be described by enumerating product wavefunctions or operator eigenfunctions.
- The system dynamics are modeled by differential equations or difference equations whose solutions (trajectories) describe how the system model moves from state to state. A trajectory corresponds to an inference arrow in Figure 1.
- There will be one trajectory through each state, so that given a current state, the dynamics prescribe its list of all past and all future states. The same is true quantum mechanically—once we fix the time independent wavefunctions or eigenfunctions, their future and past time evolution are determined.

We can see that the system model answers the initially posed structural question. The pieces are what the system is made of, and the dynamics of the system model explain how the system works in terms of the dynamics of the individual pieces and their interactions. The dynamics of many seemingly very different natural systems can be modeled with the same pieces; structural models are rather agnostic to what a system does. Thus structural models have the advantage of great quantitative predictive power. Everything to be said about the system is said from the bottom-up in terms of the pieces.

## **2.2 Functional Models**

A functional model is motivated by a different question, “What does this system do and how is it organized?” We first assume that what the system does can be described in terms of functional components and that we can analyze the system behavior in terms of these components. Each component in a functional model has constitutive or fixed behaviors that allow you to identify its distinct role within the system. At the same time, to have a functional role in the system, a component also has variable behaviors so that the behavior of the system influences the behavior of the component and vice versa. The mathematical model of a component is a mapping with specified domain and range. The mapping is the identifiable function of the component, while the domain represents the influence of the system behavior on the component and the range represents the influence of the component behavior on the system.

A system level functional model has its own unique set of properties:

- A functional architecture describes the system organization by articulating all the domain/range relationships between components (mappings) and the constituent behaviors of each component.
- The domain and range of mappings are sets, and these can be sets of mappings. Thus, mappings can be in the range and domain of mappings, and entire components themselves can be inputs to and outputs of components. This leads to very rich architectural possibilities.
- The inference arrow in Figure 1 corresponds to traversing a path in the functional architecture (composition of mappings representing components or subcomponents). Because the architecture can potentially be very rich with many

---

different types of mapping relationships, the number and types of inferences are also potentially very rich.

While, the functional question is rather different from the structural question, the functional model in fact answers it. The paths through the functional architecture explain what the system does, and the way that the components in the functional architecture cooperate to give rise to these system behaviors demonstrates how the system is organized. Because functional models are agnostic to what pieces the system is made of structurally, they can describe behavioral and organizational similarities between very different natural systems. Thus a single model describes what a class of systems has in common behaviorally and organizationally. Functional models have the advantage of great qualitative descriptive and analytical power based on a rich mathematical formalism. The implicit assumption is that everything to be said about a system is said from the top-down.

### **3. Practical Context: Science and System Engineering**

Science traditionally begins with observations, both quantitative and qualitative, and then moves directly to the synthesis of structural models. Structural models are built from the bottom-up and generally focus on quantitative prediction. As we have seen, the structural model of the system has a state space and dynamics based on the parameters, states, and dynamics of the pieces. At this point, there is no functional model so we have not even asked whether the system is organized. If we solve or simulate the structural model dynamics, decode or interpret predictions about our natural system, and observe the behavior of these predicted dynamics, we are in essence creating an informal functional model based on what we decoded from the structural model. Here, in a very explicit sense, structure determines function, because the function we observe is based on, and constrained by the dynamics generated from the structural model. So while quantitative prediction is the primary measure of a structural model's success, we can interpret a structural model's dynamics as behavior and then analyze it functionally to answer questions about organization. This is the traditional approach for the study of emergent systems.

System engineering (see Appendix A) traditionally begins by translating observed behaviors into functional, performance, and interface requirements (Defense Acquisition University 2005). The choice of components and their organization are determined by the system level behaviors as expressed through the system requirements. The functional architecture is developed using functional analysis, and requirements are allocated to components of the functional architecture. Performance and interface requirements are also allocated to the components as the architecture is refined. The functional model is built top-down and generally focuses on behavioral fidelity. At this point there are no pieces chosen for implementation—no stuff—in other words, there is no structural model. However, we can decode any portion of the functional architecture by examining specific structural implementations of a component. We may observe that some functions that are better implemented by one collection of pieces, while others are better implemented by another collection of pieces. These implementation considerations, or trade studies, can be simulated, analyzed, or prototyped. In each trade study, we are

---

decoding a portion of our functional architecture, asking structural modeling questions, and synthesizing a structural model. This time, function determines structure because the structural model we are considering is constrained by the functional architecture and performance parameters; not any choice of pieces, or any structural model can be used to implement the functional architecture.

To formalize the observations of the previous two paragraphs, we refer back to Figure 1, Rosen's Modeling Relation. If you decode a model, you are predicting phenomena in nature. We can decode particular structural dynamics and see if there is something to model functionally, or we can decode portions of a functional architecture and see if there is something to model structurally. As long as we decode and encode in between, the two models can be applied sequentially. We ask both structural and functional questions of the same natural system in a disciplined and rigorous way. With this method, we can, as stated in the introduction, "...extend our ability to model, understand, and explain complex systems in nature."

The order in which we apply the models depends on the question(s) we are asking, and it depends on the system we are studying. Recall that a structural model asks what a system is made of and assumes that everything that there is to say about a system can be said via synthesis from the bottom-up. Explanation in a structural model flows from the pieces to the whole system. A functional model asks how a system is organized and assumes that explanation in the model flows via analysis from the top-down. Functional components inherit their behavior from the system. If we are more interested in how a system is organized, in how it carries out its functions, in its qualitative aspects, and we believe that the system is more than the sum of the pieces, then we apply the functional model first. If we are more interested in what a system is made of, if we want to understand its quantitative dynamics, and we believe that the system is nothing but the sum of its pieces, then we apply the structural model first.

The most interesting applications for this extended approach are organized systems where we observe both system level functions and components that cooperate to perform these system level functions. Biological systems are a prime example. On the other hand, we prefer an initial application that is sufficiently constrained and where there is already a significant body of work in structural modeling, so that we can evaluate the utility of the extended approach. For these reasons, and because of the importance of the problem, we will briefly examine a key problem in proteomics, determining the three-dimensional folded conformation of a protein.

#### **4. Application Context: Protein Folding, Genomics, and Proteomics**

*"How does the amino acid sequence of a protein specify its three-dimensional structure? And how does an unfolded polypeptide chain acquire the form of the native protein? These closely related questions are the core of the protein folding problem, one of the most challenging and important areas of inquiry in biochemistry."* (Stryer 1995 p. 417)

The problem of protein folding (see Appendix B) is no less challenging and even more important today. Proteins play critical functional roles in all living organisms. These

---

roles include enzymatic catalysis, transport, coordination of motion, mechanical support, and regulation of growth and differentiation. The goal of structural genomics (Burley et al 1999; Chandonia and Brenner 2006) is to determine the three-dimensional conformations of an entire proteome. By analogy with the term *genome*, the proteome is the complete set of proteins for an organism.

With advances in NMR spectroscopy and X-ray crystallographic techniques, the number of known protein structures has increased rapidly, and in fact the number of structures solved per year has increased steadily. In 1990, 144 new structures were solved, but over 2,600 were solved in 2000 and over 7,000 were solved in 2008. By the end of 2008 there were over 55,000 solved three-dimensional structures in the Protein Data Bank (PDB) (RCSB Protein Data Bank 2009).

We would like to believe, and there are good reasons to believe, that when we isolate a protein and crystallize it, that protein maintains a conformation closely related to its native conformation. However, in proteomics we seek the conformation not of the isolated crystallized protein, but rather of the protein *in situ*, in its functional context. The distinction is important because *in situ*, the protein is operating in a non-equilibrium open system as one of several components organized functionally. In fact, it is common for a protein to ‘contain’ several functional components in the sense that there are multiple functions associated with that protein. The protein in a crystal is not operating at all because it is in a closed system at equilibrium. The crystallized protein is part of an ordered system, not part of an organization; there is no function at equilibrium, and there is nothing to model functionally. We can immediately see that seeking the conformation of a crystallized protein more or less excludes functional modeling.

#### 4.1 Structural Protein Modeling

*Ab initio* protein modeling begins with the chemical and physical description of the pieces of a protein and proceeds to predict the conformation and dynamics of the protein system using a structural model synthesized from those pieces in a particular sequence. To put this approach into our structural modeling terminology, the pieces are the amino acid (peptide) subunits of the protein polymer. The constituent parameters of each piece include the parameters that define the amino acid’s side chain and the parameters that define the peptide backbone (constituent atoms, bond lengths, steric properties, chemical properties). The variable configuration of the peptide subunit is described by bond rotation angles at the alpha carbon and depending on the fidelity of the model, rotation angles within the side chains. Again, depending on the fidelity of the model, there may be state variables associated with each side chain’s chemistry, or these properties may be fixed by constituent parameters. Finally, for each piece, forces or potential energies in the environment of that piece determine the bond angle dynamics. These forces can be steric (two atoms cannot occupy the same space) or bonding (disulfide links, hydrogen bonding, van der Waals interaction) or can be related to creation/disruption of solvent structure (hydrophilic or hydrophobic influences). In all the above cases, the forces depend on the constituent parameters and the variable configuration.

The *ab initio* protein model inherits the constituent parameters of all the pieces, and it also inherits a state space. The conformation of the protein is completely described by specifying the sequence, specifying all of the alpha carbon angles, and again depending

---

of the fidelity of the model, specifying the various side chain variables. The dynamics of the system model are also inherited from the pieces, because the total energy of a conformation is just the sum of all the energies of interaction between the pieces. The difficulty is that the environment of each piece depends on the alpha carbon bond angles of many other pieces, so the protein state space is large and the protein dynamics are complicated. Further, the difference in energy between different folded and unfolded protein states is relatively small. This is particularly true if you consider the energy difference per amino acid. Last, but not least, van der Waals bonding, hydrogen bonding and hydrophobic/hydrophilic interactions are dominant factors, and water, the solvent, plays a key role in mediating these forces.

## **4.2 Functional Protein Modeling**

For concreteness, consider a protein that is a soluble enzyme. Using a system engineering approach, we begin by stating requirements that describe this enzyme in a system context. This might include the reaction(s) that the enzyme catalyzes, the broader biochemical pathway(s) that supply inputs or utilize outputs of those reactions, regulatory interactions with other reactions, ranges of catalytic rates and constants, inhibitors, activators, ranges of chemical environment in which the enzyme does or does not operate, and physical characteristics like solubility, and so forth. Next, we model the behaviors described by the requirements as a functional architecture. In this process, the architecture is refined, and the requirements are assigned to the functional components in the architecture. Notice that at this point there are no amino acids, no primary sequences, i.e. nothing structural, since our model is looking at the relationship between the functions in the system rather than looking at the stuff that is implementing those functions. In the abstract, we could consider that the catalyst (enzyme) need not even be a protein if it could perform the required functions. The example of catalytic RNAs or ribozymes comes to mind (Cech 2004). However, let's assume that we have sufficient data to tell us that it is a protein.

With this functional model, we have a series of new opportunities. First, we can look for functional homology between different metabolic pathways either within an organism or across species to identify proteins that perform related functions (Mcdermott and Samudrala 2004). We identify similarities in regulatory relationships, binding, activity, which imply tertiary conformational motif or fold similarity in enzymes with similar functional characteristics. We can then look for similar structural realizations in proteins that share functional components. Using functional homology, we are working from function to a tertiary structure hypothesis in a top-down system sense.

## **4.3 Current Practice in Protein Modeling**

In addition to the protein sequences with known PDB tertiary structures, a much larger number of protein primary sequences can be 'read' directly from sequenced genomes (Chandonia and Brenner 2006). So in addition to the direct solution of protein tertiary structure, modeling activity has increased dramatically. Every two years there is an assessment of the state-of-the-art in protein structure modeling called Current Assessment of Techniques in Structural Modeling of Proteins or CASP (Moult 2005). The goal of this modeling effort is to predict the conformation of a protein directly from its primary

---

sequence. The evolution of modeling as viewed from the perspective of these regular CASP conferences is quite interesting (Moult 2005; Zhang 2008).

While *ab initio* models have been refined and developed, they are not the predominant approach to protein conformation modeling today. The most successful protein folding models use various forms of template matching. Simple sequence homology compares primary sequence runs in the target protein, the protein whose conformation is being modeled, with sequences of template proteins whose structure is already known (Zhang 2007). If there is sufficient homology between a sub-sequence in the target protein and a template protein sequence, then the known secondary structure is assigned to the target sub-sequence, and that rigid unit is treated as a new piece. Note that many degrees of freedom in the structural model are constrained by this assignment.

Higher-level templates are used to match tertiary structural motifs or protein fold patterns. This more powerful technique, protein threading, matches the target sequence against templates searching for tertiary structural homology. Threading can be quite successful even when there is very little sequence homology. When a high level template is found, the structural modeling degrees of freedom are even more constrained than in the secondary template matching. Today's pure *ab initio* calculations are most useful in understanding and refining the forces and dynamics at the piece level of the model. *Ab initio* insights are used in heuristics to improve threading performance and to optimize models developed using sequence homology and threading.

Results of CASP are both impressive and disappointing (Zhang 2008). Template matching gives conformations that very closely match the proteins that are used as templates. To the extent that the target protein has a structure similar to the template, the conformational modeling is very good. For the proteins that only weakly match the chosen template, the results are considerably worse, and in the case where no reasonable template is found, the *ab initio* approach must be used. In the study cited above (Zhang 2007), using *ab initio* calculations, the general topology of protein folding was approximately correct in only seven of nineteen cases (average length 155 amino acids).

Over the years, a classification system has been developed, Structural Classification of Proteins (SCOP) (Murzin et al 1995). This has been done by hand based on similarity of secondary structural domains or *folds* in a given family of proteins and also based on known evolutionary similarity of protein sequence. The SCOP categories have remained relatively stable over the past 10 years (Andreeva et al 2008), and this is an indication that the PDB spans the space of structural conformations for soluble proteins (Zhang and Skolnick 2005). Unfortunately, our ability to find the correct template is actually rather poor (Zhang 2008). Our ability to automatically assign newly solved protein structures into SCOP is also rather poor (Andreeva et al 2008).

#### **4.4 Functional /Structural View of Current Modeling Practice**

With the automation of gene sequencing and more rapid determination of protein tertiary conformation, protein folding has largely been transformed from a structural modeling problem in physical chemistry to a bioinformatics problem of finding a good template in the PDB for a target protein. However, if we examine the template approach closely, we can see heuristics based on complementary functional and structural models. Further, it

---

is the homology between functional models that significantly constrains the problem and allows us to determine the conformation more quickly and effectively using templates.

The most powerful template based technique, sequence threading, begins with a template of known tertiary conformation. Enzymes with similar functions have highly conserved conformation across species and conformation is much more highly conserved than sequence (Shin et al 2007; Sims et al 2009). In other words, when we choose a template, we are implicitly, hypothesizing a functional model for the target protein. In fact, threading is nothing more than a heuristic means of applying the constraints of the template protein's functional model to determine if the target sequence is structurally capable of those functions by seeing if it can fold into that conformation. We are literally working from function to structure with the threading approach. Once we have a decent template match, we can use heuristics based on our *ab initio* experience to explore the immediate vicinity of the template conformation using the target sequence and its side chains. So in threading, we first apply the functional model to constrain the structural possibilities and then we apply structural modeling to optimize the folded conformation.

It is interesting to reflect on why the threading approach is so powerful. First, the conformational space for an *ab initio* protein model is extremely large, while the fold space or number of unique tertiary structures for soluble proteins in the PDB is surprisingly small (Zhang and Skolnick 2005). When we apply the tertiary constraints first, we greatly reduce the dimensionality of the problem as progress in CASP has borne out. To press this further, why is the fold space is so small? The fold space is small because of evolutionary constraint, and it is function that is being evolutionarily constrained. Variation in the primary sequence of a protein is evolutionarily constrained to give rise to tertiary conformations that are sufficiently close in function to the wild type. The conformations of proteins that occur naturally are highly constrained by their evolutionary history and therefore by their biological function.

The fact that function and conformation are much more highly conserved than sequence is empirical evidence that functional requirements constrain, but do not determine the primary sequence of a protein. So we should not expect to predict protein folding with a purely functional model. And yet the *ab initio* experience has shown us that we cannot get very far with a purely structural model either. It is clear that functional conservation through evolution has given rise to a large number of system level functional constraints in protein folding. When we apply these constraints before we model structurally, we get a superior result.

Though the heuristics used are not usually described in these terms, we can see that the current state-of-the art in predicting protein tertiary conformation is in fact critically dependent on functional modeling. We also see the interplay of structural and functional constraints and improved predictions when an extended modeling approach is used.

## **5. Conclusion**

We have developed a general method using system engineering techniques to create functional models of natural systems, and recent progress in protein modeling demonstrates the power of this approach. Specifically, functional modeling constrains the acceptable structural models of protein conformation so that we greatly reduce the difficulty of finding structural models, and improve their fidelity. Protein modeling has

---

evolved from pure *ab initio* calculations to template-based heuristics precisely because functional modeling provides these improvements. Also, functional models better represent many of the questions that we ask in proteomics. We seek the conformations of proteins to better understand how proteins are related functionally in a pathway, an organelle, or an organism, i.e. to understand how an organism is organized. We are asking a functional question, “What are the functional components and how do they cooperate to achieve the behaviors of the organism?” Our wealth of structural data provides tremendous insight, but the question of organization is functional, and we should not expect to answer it in strictly structural terms.

Our approach to soluble proteins naturally extends to other biochemical systems. As we articulate broader functional contexts and relate them to the continuing flood of structural data, we will gain ever-greater problem solving leverage at the system level. The size of the structural model state space grows exponentially with the number of pieces, and yet analysis of functional model constraints can make complicated problems significantly more tractable. System engineering is well suited to such information intensive analysis task in biology and beyond.

We can combine functional/structural modeling, or more broadly qualitative/quantitative modeling, to studies in other fields. In each field, we choose a system of interest, put qualitative/quantitative models for this system into a common context using the modeling relation, and then examine how the models constrain one another. Clearly, the utility of the result depends on the nature of the models, so this is not a prescription for choosing specific quantitative or qualitative models. However, our approach provides a disciplined method for applying both top-down and bottom-up mathematical models to a broad range of applications.

Finally, this combined approach sheds light on a theory of organization in nature. When we put functional and structural models in a common theoretical context, we clarify the difference between organization and order. Organization is inherently functional; it is a description of the relationships between functional components that cooperate in producing system behavior. Order is inherently structural; it is a description of the arrangement of structural pieces that interact in producing system dynamics. Thus, functional models do not model order and structural models do not model organization. However, order and organization are coupled in a subtle and important way. As we have seen, when a specific system in nature is a realization of both a functional and a structural model, the models constrain one another. A more organized functional model constrains the minimum order necessary in the structural model, and a less ordered structural model constrains the maximum organization possible in the functional model. This brings us to the term *emergence*. *Emergence* is commonly used in science to comment about organization with respect to a structural model. When we add a functional model and describe organization properly, emergence need not be so mysterious.

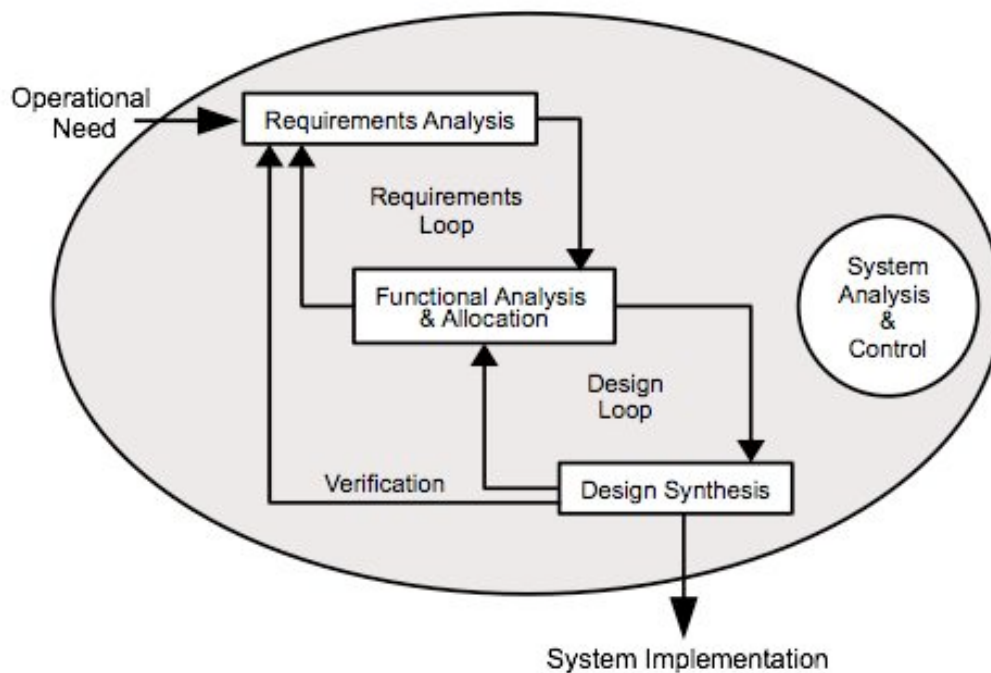
### **Acknowledgements**

I am grateful to Robert Kuberek for valuable discussions and critical insights throughout this endeavor. An anonymous reviewer provided comments that helped me correct and improve the manuscript.

---

## Appendix A: System Engineering

System engineering (Defense Acquisition University 2005) is a technical management methodology that has been developed over the past 50 years. While originally used to develop large defense systems for embedded control and information management, it has become the basis for many engineering methodologies, including modern software development and information technology systems development. System engineering may be defined as a logical sequence of activities and decisions that transforms an operational need into a description of system performance parameters and a preferred system implementation. In this context, a system is an integrated composite of people, products, services and processes that provide a capability to satisfy a stated need or objective.



**Figure A1 The System Engineering Process**

Figure A1 describes the flow of basic system engineering activities: requirements analysis, functional analysis and allocation, and design synthesis. These activities are managed using a set of tools called system analysis and control. The first activity, requirements analysis, translates the operational need, a top-level description of the system's behavioral repertoire, into top-level functional and performance requirements. During requirements analysis, these requirements are analyzed to identify implied requirements, to determine that requirements are verifiable, and finally to ensure that the set of requirements is a consistent, coherent, and complete description of the operational need.

The second activity, functional analysis and allocation, provides a description of system functions that can be used to guide the design synthesis that follows. This activity includes arranging functions in logical sequences and decomposing higher-level

---

functions into lower-level functions using functional flow block diagrams and timelines. The value of this activity is that it will give us a means for tracing requirements, via functions, to implementation descriptions that are potential outcomes of the design synthesis activity without unduly limiting the range of design choices. The output products of this activity include a functional architecture that describes the system in terms of component functions and a set of functional performance parameters. This activity does not include a physical description although there may be physical and environment parameters associated with either the system level or particular functions.

The third activity, design synthesis, develops implementation designs using the products of functional analysis and allocation. This activity eventually develops a physical architecture that includes a set of products, hardware, software, and processes capable of performing the required functions within the limits of the performance parameters prescribed. Since there may be many physical architectures capable of meeting the functional and performance requirements, this activity sets the stage for trade studies to select the best among candidate implementation architectures. The objective is to combine and restructure products, hardware, software, and processes in such a way as to achieve a design solution capable of satisfying the operational need.

### **Appendix B: The Protein Folding Problem**

Proteins are polymers called polypeptides, because they are made from amino acids by forming peptide bonds. A polypeptide has a regular repeating part called the peptide backbone and a series of variable parts called side chains. The 20 different side chains, corresponding to the 20 different amino acids, have a variety of sizes, shapes, and a rich repertoire of chemical properties. Each specific protein has a unique amino acid (side chain) sequence, and a typical protein has several hundred peptide bonds, though there are biologically significant polypeptides that are quite short and proteins with sequences that number several thousand.

A folded three-dimensional structure for a protein is a conformation. The native or active conformation is the folded three-dimensional structure of the protein when it is performing its functional role. The conformation is usually described in terms of specific levels of structure. The sequence of amino acids is the primary structure. The secondary structure is the local structure of the protein based on interactions between amino acids that are close to one another in the primary sequence. Examples of secondary structure are the alpha helix, and the beta pleated sheet. These secondary structures tend to be quite rigid as units. Tertiary structure refers to folding that is based on interaction between distant amino acids in the primary sequence. Most soluble proteins (many enzymes) have a compact globular tertiary structure in which portions of the protein with relatively rigid secondary structure are connected by shorter sequences of amino acids that are less constrained. This allows large regions of the protein to fold back upon one another to create the globular structure. Quaternary structure is the arrangement of multiple folded polypeptides into a multi-subunit conformation.

---

## References

- Anderson PW (1972) More Is Different. *Science* 177(4047):393-396
- Andreeva A, Howorth D, Chandonia JM, Brenner SE, Hubbard TJ, Chothia C et al (2008) Data growth and its impact on the SCOP database: new developments. *Nucleic acids research* 36(Database issue):419-25
- Burley SK, Almo SC, Bonanno JB, Capel M, Chance MR, Gaasterland T et al (1999) Structural genomics: beyond the Human Genome Project. *Nature Genetics* 23(2):151-158
- Cech TR (2004) Exploring the New RNA World.  
[http://nobelprize.org/nobel\\_prizes/chemistry/articles/cech/index.html](http://nobelprize.org/nobel_prizes/chemistry/articles/cech/index.html). Cited 19 Feb 2009
- Chandonia J, Brenner SE (2006) The Impact of Structural Genomics: Expectations and Outcomes. *Science* 311(5759):347
- Defense Acquisition University (2005) *Systems Engineering Fundamentals* January 2001. Fort Belvoir, VA
- Laughlin RB, Pines D (2000) The Theory of Everything. *PNAS USA* 97:28-31
- Mcdermott, J, Samudrala R (2004) Enhanced functional information from predicted protein networks. *Trends in biotechnology* 22(2):60-62
- Moult J (2005) A decade of CASP: progress, bottlenecks and prognosis in protein structure prediction. *Current opinion in structural biology* 15(3):285-289
- Murzin AG, Brenner SE, Hubbard T, Chothia C (1995) SCOP: a structural classification of proteins database for the investigation of sequences and structures. *Journal of molecular biology* 247(4):536-40
- RCSB Protein Data Bank (2009)  
<http://www.rcsb.org/pdb/statistics/contentGrowthChart.do?content=total&seqid=100>. Cited 19 Feb 2009
- Rosen Robert (1991) *Life Itself*. Columbia University Press, New York
- Sauer U, Heinemann M, Zamboni N (2007) Genetics Getting closer to the whole picture. *Science* 316(5824):550-551
- Shin DH, Hou J, Chandonia JM, Das D, Choi IG, Kim R et al (2007) Structure-based inference of molecular functions of proteins of unknown function from Berkeley Structural Genomics Center. *Journal Of Structural And Functioning Genomics* 8(2-3):99-105
- Sims GE, Jun S, Wu GA, Kim S (2009) Alignment-free genome comparison with feature frequency profiles (FFP) and optimal resolutions. *PNAS USA* 106(8):2677-2682
- Stryer Lubert (1995) *Biochemistry*. WH Freeman & Company, New York
- Wicken J S (1987) *Evolution, Thermodynamics, and Information*. Oxford University Press, New York

---

Zhang Y (2008) Progress and challenges in protein structure prediction. *Current Opinion in Structural Biology* 18(3):342-348

Zhang Y (2007) Template-based modeling and free modeling by I-TASSER in CASP7. *Proteins* 69:108-117

Zhang Y, Skolnick J (2005) The protein structure prediction problem could be solved using the current PDB library. *PNAS USA* 102(4):1029-1034