Engineering, Science, and Design

Francis T. MARCHESE
Computer Science Department, Pace University
New York, NY 10038, USA

ABSTRACT

It is argued that the fields of science and engineering are at two ends of a spectrum of shared technological products (artifacts) and processes in which, at any given time and to some lesser or greater degree, scientists behave as engineers and engineers as scientists, creating technological artifacts and contributing to knowledge domains that drive and advance both fields. Employing an example from synthetic chemistry, it is shown that the concept of design governs the research process and production of its artifacts. And, just as it does in engineering, these artifacts exhibit the functional and quality characteristics of engineered products.

Keywords: Scientific practice, Design**,** Synthetic chemistry, Object-oriented software engineering.

1. INTRODUCTION

It is the purpose of this paper to put forth the notion that design is the fundamental principle that unifies science and engineering. This will be accomplished by drawing on an example from synthetic chemistry. The field of synthetic chemistry encompasses the discovery, design, and synthesis of new molecules and the understanding of the intricate mechanisms by which they react and interact. Specifically, chemical synthesis is the purposeful execution of chemical reactions to create a product by means of a process that is both reproducible and reliable.

Much of chemistry is synthesis. Chemical substances are designed and built for many reasons, but two of the most important are the creation of new and useful products such as pharmaceuticals, and the understanding of the underlying physical laws that govern chemical structure and process. By far, synthesis is the essence of chemistry. Schummer [1] has found that chemists had created 1.3 million new substances in 1996, with an annual growth rate of 5.5% per year. And 95% of these molecules were unknown, that is, they had neither been detected in Nature nor synthesized before. He found further that, through an analysis of 300 research papers dated from 1800 to 1995 from the general chemistry journal *Angewandte Chemie*, synthesis papers outnumbered theory by a margin of 5 to 1. Today, there are over 50 million known chemical compounds, with the 50 millionth cataloged by the Chemical Abstract Service in September 2009 [2]. Extracted from a two hundred page patent application, it was given as an example of a possible pharmaceutical candidate for reducing neuropathic pain, hence, a *designed drug*.

Synthetic chemistry embodies Callaos' [3] three components of engineering practice: *Scientia*, *Techné*, *praxis,* as well as meta-engineering. *Scientia* is propositional knowledge, or know-what. Synthetic chemists must be domain experts, mastering a bewildering array of chemical reactions, patterns of reactivity, and problem solving skills that are the foundation of the practice.

Technê is procedural knowledge or know-how. Synthetic chemists are designers and builders who not only demonstrate fluency with laboratory procedures, but also employ these procedures regularly and in a systematic way to fabricate chemical substances. One of the most important ingredients of synthetic chemical know-how is the design of a reaction pathway in which a sequence of chemical reactions must be selected from a myriad of possibilities to yield the requisite product. Although many synthetic routes may be possible to fulfill the design requirements, a truly great synthesis exhibits a simplicity, refinement, and overall beauty that are expressions of a designer's intelligence and ingenuity. Such syntheses may be considered to rival a great work of art or engineering [4].

Praxis is tacit/personal knowledge. It is exhibited as a combination of an individual's innate ability and on-the-job experience. Great synthetic chemists are said to have the *hands* or laboratory skills necessary to finesse a complex sequence of chemical reactions into a finished product. Not every synthetic chemist possesses these skills to the same degree. A typical gauge of an individual's synthetic prowess is the replication of a well known synthesis of a prescribed difficulty taken from the original literature. The quality of the output is an indicator of the chemist's practical laboratory skills or experience and to a significant degree reflects his or her attention to detail.

Meta-engineering, the improvement of engineering process, is a significant part of synthetic chemical practice. In an ideal world, a *single-pot* synthesis in which all reactions take place in one vessel under a simple set of experimental conditions producing 100% yield of a desired product would be the norm. But in reality this is seldom the case, because each reaction step requires a change in experimental conditions and results in a product yield that may vary significantly from 100%. Consequently, synthetic chemists continually experiment with developing syntheses that reduce time and costs by maximizing product yield (hence less waste), developing more economical starting materials, and shortening the length of synthetic pathways. In sum, process improvement and optimization are integral elements of synthetic chemical practice.

Finally, invention, a process normally coupled with engineering practice, is an essential attribute of synthetic chemistry. As we noted above, 95% of the 50 million known chemical substances were either invented outright or constructed through modification of preexisting substances. And many of the synthetic pathways necessary for their formulation may have been invented as well. Yet, synthetic chemistry is not chemical engineering. What separates these two fields is the issue of scale. Chemical laboratory syntheses generate products whose

quantities are measured in grams or fractions of liters. It is left to the chemical engineers to scale up these bench processes for mass manufacturing. And because many chemical reactions are complex in nature, they and the experimental conditions under which they are run, may not scale explicitly. As such, the character of these chemical reactions may need to be researched further by chemical engineers in order to ascertain the experimental conditions necessary to support syntheses on a practicable scale.

2. AGENDA

We believe that the above discussion has raised sufficient parallels between synthetic chemistry and engineering to warrant further exploration into the common processes shared by scientists and engineers. Indeed, it is often said that engineers learn so they may build, while scientists build so they may learn. This synergy between science and engineering goes to the core of our thesis, that is, design governs the research process and production of its artifacts. Since design and artifact construction are normally considered to be an engineering activity, we argue by analogy that the artifacts of the scientific research process, from instrumentation, to synthetic chemical pathways, to complex software, are designed and exhibit FURPS (Functionality, Usability, Reliability, and Performance) characteristics of engineered products. As such, we should be able to employ engineering methods in our analysis of scientific practice.

We will employ the concepts that comprise object oriented software engineering for the analysis of synthetic chemistry. The software engineering process encompasses modeling, problem solving, and knowledge acquisition [5]. In modeling, software engineers deal with complexity by focusing on only the relevant component details and ignoring everything else. Abstraction is used as part of the software engineering's analysis-design-build process to create analysis models that represent the problem domain and design models that represent the solution domain. Problem solving is an activity in which models are used to search for an acceptable software solution. The process is driven by experimentation, the use of design patterns to build models, the incremental evolution of the system towards a converged solution, and the adaptation and revision of models in response to change. As part of knowledge acquisition, software engineers collect data, organize it into information, and formalize it into knowledge. And at software engineering's core is the concept of design – understanding the design of the real-world system so it may be mapped onto a software design.

Our analysis begins with a definition of design. An example of synthetic chemical practice as a problem in reverse engineering follows. The paper ends with a discussion and conclusions.

3. DESIGN

Design is an underlying scheme that governs the structure, function and development of a system. From the architecture of the ribosome to integrated components of a cell phone, the consequences of design are ubiquitous. Buildings, bridges, automobiles, computers, coffee makers, highways, networks, and software are all designed. The shape of a finch's beak, the systematic structure of the elements, and the machinery behind DNA's coding and expression all exhibit attributes of design that governs structure and process. Indeed, what makes Nature's laws experimentally accessible and amenable to theoretical analysis is that they reveal a coherent organization of structural, functional, dynamical and developmental relationships, hence, a design. Design is the central step in an engineering process of analysis – design – build, where the goals are to create useful artifacts (products). Design is a mediated activity. Designers use tools from pencil and paper to computer-guided milling machines to create representations of their creations. Software engineers employ theories, modeling languages (e.g. UML), design patterns, editors, compilers, and alike to design software. Design tools, in-and-of-themselves artifacts of other design processes, allow designers to meet the conceptual requirements for a product's functionality, given constraints such as size, shape, materials, technology, and cost. The scientific process is replete with artifacts such as beakers, balances, spectrophotometers, centrifuges, electronic laboratory notebooks, mice, models, experiments, and even theories - all of which are designed.

A theory is an artifact or product of a design process, the goal of which is to create a useful representation of physical system that can be employed to explain the results of experiment and predict a system's attributes when experiment is unavailable. Scientific theories are designed, beginning with a collection of propositional concepts (e.g. insights, guesses, intuition, hypotheses) related to the phenomena to be understood. These are then transformed into formal, typically mathematical representations utilizing mediating tools such as laws, theories, models, and languages, under design constraints dictated not only by experimental data but also the theoretical and mathematical mediums within which the theory will be devised. Just as the material selected for the shell of cell phone (e.g. metal vs. plastic) specifies its ruggedness, or the speed of its processor limits its ability to send and receive video, the selection of mathematical form (e.g. Euclidean vs. non-Euclidean geometry) limits a theory's generality. Thus, whether assembled or derived, theories are arrangements of component parts integrated for a purpose under a set of constraints.

Design may be expressed in natural and artificial systems alike through the notions of abstraction and modularity. An abstraction is a representation that does not include all of the system's properties, leaving out features that the system has in its concrete form. The abstraction process formulates general concepts by selecting common properties of instances that are essential to defining the system while ignoring inessential details. Designers speak of "levels of abstraction," meaning that as they move to "higher" levels of abstraction, they shift their attention to the essential attributes of the system; while moving to "lower" levels of abstraction signifies focusing on implementation details, such as how the artifact is constructed. One way to envision this layering of abstractions is to look at the organization of process and structure for chemical systems. At the highest level of abstraction are the conceptual designs that specify the universal processes or dynamical patterns that can be considered independently of any structure or material form. For a chemical system, reaction patterns of oxidation-reduction, functional group transfer, hydrolysis, and isomerization, realized through specification and modification of a molecule's structure are examples of the highest level of abstraction. At the molecular architecture level, design specifies the guiding models behind the construction of all molecules, such as the structural chemical rules or theories - pure formal ideas or specifications without physical manifestation. At the lowest level of abstraction, a design procedurally defines the

creation of molecule's physical manifestation and measurable properties.

Design organizes the conceptual abstractions that constitute a knowledge domain and provides blueprints for the creation of objects that are instances of these abstractions. In object oriented software design, domain knowledge is organized into a design model consisting of a set of classes that define an object's attributes and behaviors. In general, a class is a template or pattern for the creation of objects that are instances of a class. It is the foremost goal of the design process to determine the high-level conceptual abstractions that become the referents for instantiations in any medium whether a mathematical theory, computer model, or physical entity. For a scientist, the possession of an understanding of these conceptual classes allows the design and construction of not only more faithful or useful representations of the real world but also alternative instantiations.

4. REVERSE ENGINEERING

This abstraction process is equivalent to a reverse engineering process in which a scientist works backward from physical instantiations (i.e. real-world objects and systems) to abstract out the conceptual representations that underlie the physical systems. Reverse engineering is the method of analyzing a system to identify its component parts and the interrelationships among those parts in order to either create representations of the system in another form or representations at a higher level of abstraction [6]. Reverse engineering generally involves extracting design artifacts to synthesize more general abstractions. It does not involve changing the system under study or creating a new system based on the reverse engineered system, although these are two characteristic goals of the process. As Chikofsky and Cross have stated [6] "reverse engineering is a process of examination, not a process of change or replication." As an example of how a scientific investigation may be viewed in this way, we consider the design of a mimetic enzyme, a molecule that functions like an enzyme but has a different structure.

Enzymes are molecular machines fashioned by living organisms that catalyze chemical reactions without being consumed in the process. Enzymes come in different shapes, sizes, physical characteristics and perform different tasks such as nitrogen metabolism (e.g. glutamine synthetase) and digestion (e.g. chymotrypsin). Superoxide dismutases (SODs) are a family of enzymes that neutralize radical anions of oxygen molecules called superoxides, which are toxic to cells, thereby reducing inflammation and preventing tissue injury caused by these radicals [7]. Elevated levels of superoxides can overwhelm the body's natural production of SOD, resulting in a continuing cycle of cell injury and inflammation. Treatment of inflamed cells with additional SOD does not solve this problem because SOD administered from external sources cannot penetrate a cell's membrane and thus cannot reach the superoxide within. The design problem is – how to build a molecule that replicates SOD's activity and can pass through the cell wall. The design process then requires extracting and mapping out the conceptual classes that govern enzyme structure and reactivity, by determining the kinetic processes involved in the neutralization of superoxide radicals and how the structure of the enzyme's active site controls its activity. This knowledge makes it possible to instantiate structural and kinetic theories, design and perform experiments on idealized systems, and design alternative enzyme structures that mimic SOD's reactivity. At the same time it must meet design constraints such as solubility in water, ability to pass through a cell's wall, not react adversely with other molecules, and decompose gracefully *in vivo* into nontoxic compounds.

Dennis Riley and coworkers [8] approached this drug-design problem not with the intent of building a replica of the active site of SOD in a small molecule but rather to understand the mechanism of catalysis, and to duplicate the mechanism in a small molecular weight, drug-like molecule. As part of their computer aided design process they developed and synthesized a family of molecules, characterized their three-dimensional structure, measured their kinetic activity *in vitro* and *in vivo*, and analyzed them utilizing molecular modeling methods in order to develop a theory of the details of SOD catalysis that could correctly predict the effects molecular structure exerted on catalytic rate. In the end, they produced a molecule called M40403 [9]. This molecule possessed kinetic activity comparable to the natural enzyme; was thermodynamically stabile, so it would not readily decompose when introduced into the human body; could be easily synthesized; and was highly active in a variety of pharmacologically relevant models of human disease. This drug is currently being marketed as an SOD mimetic.

There are two important points to be emphasized here with respect to their research process and the research product. First, this research process was one of reverse engineering carried out by scientists to extract the conceptual classes governing SOD catalysis. In order to construct a mimetic SOD, these researchers needed to perform basic research to understand the underlying physiochemical processes governing the SOD-superoxide reaction. The only difference between science and engineering was the goal of creating a useful artifact, in addition to adding to the knowledge domain. The second point is that the M40403 structure differs markedly from its natural biological counterpart, yet it performs identically in the catalyzation reaction. As such, we can say that M40403 is a SOD mimetic because it replicates SOD's activity. We could say as well that any other molecule, regardless of its structure, that performs SOD's tasks in the same way will also be a SOD mimetic. Yet, we could make a more wide-ranging statement, specifically, that both SODs and mimetic SODs are instantiations of the conceptual classes that represent the mechanisms of neutralization of superoxide radicals. This means that neither the enzyme's physical structure nor its method of manufacture is important as long as its attributes and functions validate the conceptual class's definition. Physical design and coding of conceptual classes consequently becomes a problem in engineering that is constrained by the materials and manufacturing processes available. Hence, a mimetic SOD object designed by employing computer models and synthesized in a laboratory utilizing available starting materials and requisite chemical reactions, and a SOD molecule created through evolutionary design and built from amino acids in the cell's protein manufacturing facility (the ribosome) are both valid instantiations of the highest order of abstraction stated above (i.e. the universal processes or dynamical patterns that can be considered independently of any structure or material form).

To generalize, the underlying conceptual abstractions that govern physical systems are the limiting referents, not their physical embodiments. If the latter were true, then the chemists

who created a SOD mimetic would have had to construct a molecule that was isomorphic with SOD; instead they built one that captured the essence of its catalytic activity. It then follows that any instantiation, which satisfies the conceptual models of a relevant system, should produce an object that realistically embodies those conceptual designs.

This conceptual design discovery process is composed into two parts: creation of an analysis model of the problem domain containing the conceptual classes and their inter-relationships, followed by implementation of the model's conceptual classes as an object design model expressed in some medium (e.g. mathematics, physical materials, programming language). In the field of software engineering, analysis models are created from observations of work practices and interviews with individuals such as managers and employees with the express goal of creating an abstract model of a system. These abstractions are purely informational, allowing the designers to understand how the static components of the system relate to its dynamical processes. It is crucial for the designer to ensure that the conceptual classes of the analysis model match the organizational hierarchy and dynamical patterns of the system [10]. Omissions or deviations in the analysis model typically are symptomatic of an incomplete understanding of the problem domain, and are propagated through the software implementation. By analogy, scientific theories that begin with incorrect conceptual foundations will lead to inappropriate representations. The problem faced by scientists is that Nature cannot be interviewed. Since Nature's conceptual abstractions are hidden behind the structures and processes of the physical universe, scientists must postulate conceptual classes and explore them by performing experiments, building models, and deriving mathematical theories from these concepts. As with many software engineering lifecycles, this is an evolutionary process of repeating the analyze-design-build-test sequence until it converges to expose a set of invariant conceptual classes. This has been the case with molecular structure and bonding. Chemists' concepts of molecular structure have evolved and converged over the past 100 years, for the most part without the aid of a formal rigorous theory such as quantum mechanics. Instead, experimental evidence, especially the average atomic positions supplied from diffraction experiments on crystals, has fueled the accumulation and characterization of chemical bonding that ultimately has led to the creation of rules for predicting molecular structure from a chemical formula [11]. These rules reside at the middle level of our design abstraction hierarchy (i.e. the guiding models behind the construction of all molecules regardless of physical manifestation). Chemists then use these rules with tables of standard bond lengths and angles to instantiate molecular objects in a variety of media such as scale models or molecular design and visualization programs such as those used by Riley and co-workers [12].

5. DISCUSSION AND CONCLUSIONS

We have argued that the fields of science and engineering are related by shared technological products (artifacts) and processes in which, at any given time and to some lesser or greater degree, scientists behave as engineers and engineers as scientists creating technological artifacts and contributing to knowledge domains that drive and advance both fields. By placing scientists and engineers on equal footing we have eliminated the privileged relationship that purportedly exists between scientists, scientific knowledge and reality. In turn, science inherits the concepts of design and artifact, typically attributed to engineering. From the position of design we have contended that Nature is composed of a collection of design abstractions organized into conceptual classes that are instantiated into real world objects, and it is the goal of scientists to discover these classes and their relationships. As part of these discovery processes scientists design and synthesize artifacts that embody aspects of our scientific knowledge and act as mediating tools in an evolutionary process of successive analysis-design-build-test cycles that converge over time to expose the set of invariant conceptual classes underlying Nature's designs. Although we have relegated mathematical theories and models to artifacts, we have done this with the recognition that engineers instantiate their designs in a variety of media. In particular, software engineers speak in general about the systems they build as being composed of three parts: the hardware system, the software system, and the people system. These components are orthogonal yet complementary, and their integration is essential to the system's design and success. So too in science, the medium in which the attributes of a class are expressed determines not only which attributes of an object are seen but also how well they are seen. Clearly this is the case in chemistry where a wide variety of complementary representations are used collaboratively to best express molecular structure and process. Within this context, mathematical theories no longer become the ultimate referent, but are just one of many possible useful representations.

6. REFERENCES

[1] J. Schummer, "Challenging Standard Distinctions between Science and Technology: the Case of Preparative Chemistry," **Hyle - International Journal for the Philosophy of Chemsitry**, Vol. 3, 1997, pp. 81-94.

[2] J. Raloff, "50 Million Chemicals and Counting," **Science News (Web Edition)**, September 8, 2009, http://www.sciencenews.org/view/generic/id/47036/title/Science_%2B_the_Public__50_million_chemicals_and_counting.

[3] N. Callaos, "The Essence of Engineering and Meta-Engineering: A Work in Progress," working paper accessed January 2, 2010, http://www.iiis.org/Nagib-Callaos/Engineering-and-Meta-Engineering/.

[4] R. Hoffmann, "In Praise of Synthesis," **American Scientist**, Vol. 79, Jan-Feb, 1991, pp. 11-14.

[5] B. Bruegge and A. Dutoit, **Object-Oriented Software Engineering**. New Jersey: Prentice Hall, 2000.

[6] E.J. Chikofsky and J.H. Cross, "Reverse Engineering and Design Recovery: a Taxonomy," **IEEE Software**, Vol. 7, 1, 1990, pp. 13-17.

[7] J.M. McCord and I. Fridovich, "Superoxide Dismutase. An Enzymic Function for Erythrocuprein (Hemocuprein)," **Journal of Biological Chemistry**, Vol. 244, 1969, pp. 6049-6055.

[8] D.P Riley, S.L. Henke, P.J. Lennon, and K. Aston, "Computer-aided Design (CAD) of Synzymes: Use of Molecular Mechanics (MM) for the Rational Design of Superoxide Dismutase Mimics," **Inorganic Chemistry,** Vol. 38, 1999, pp. 1908-1917.

[9] D. Salvemini, Z. Wang, J.L. Zweier, A. Samouilov, H. Macarthur, T.P. Misko, M.G. Currie, S. Cuzzocrea, J.A. Sikorski, and D.P. Riley, "A Nonpeptidyl Mimic of Superoxide Dismutase with Therapeutic Activity in Rats," **Science**, Vol. 286, 1999, pp. 304–306.

[10] C. Larman, **Applying UML and Patterns: An Introduction to Object-oriented Analysis and Design**

and the Unified Process, 3rd Edition. New Jersey: Prentice-Hall, 2005.

[11] R.J. Gillespie, and P.L.A. Popelier, **Chemical Bonding and Molecular Geometry: From Lewis Theory to Electron Densities,** New York: Oxford University Press, 2001.

[12] J.P. Doucet and J. Weber, **Computer-aided Molecular Design: Theory and Applications.** San Diego, CA: Academic Press, 1996.