IAC-10.D1.5.4

HOW DO WE FIX SYSTEM ENGINEERING?

Michael D. Griffin
Eminent Scholar and Professor
Mechanical and Aerospace Engineering
University of Alabama in Huntsville
Huntsville, Alabama, USA
Michael.Griffin@uah.edu

61st International Astronautical Congress
Prague, Czech Republic
27 September – 1 October 2010

The now half-century-old multidisciplinary, interdisciplinary discipline that we call "system engineering" is the classic half-empty, half-full glass; optimists and pessimists can look at the same thing and draw opposing conclusions. Optimistically, the maturing of system engineering into a recognized discipline from its roots in large aerospace and defense programs has been, and will remain, an enabling factor in the ability of societies to deal with the macroscale problems facing us in energy, environment, and other key areas. Pessimistically, system engineers have some explaining to do. How is it that we continue to encounter failure of important and complex systems where everything thought to be necessary in the way of process control was done, and yet despite these efforts the system failed? Each time this occurs, we as an engineering community vow to redouble our efforts to control the engineering process, and yet such events continue to occur. The answer cannot lie in continuing to do more of the same thing while expecting a different outcome. We need to rise above process, to examine the technical, cultural, and political mix that is "system engineering", and to examine the education and training we are providing to those who would practice this discipline. This paper will discuss that training from a new perspective, the perspective of design elegance, how we identify it, and how we can design with elegance as a value.

I. INTRODUCTION

The period from the aftermath of World War II to the present day has been a period of technical innovation unmatched in human history. Engineers practice today in fields such as integrated circuits, computers, software engineering, information technology, materials science, nuclear engineering, and of course astronautics, that did not exist in 1950, while traditional engineering specialties such as civil, mechanical, electrical, chemical and optical engineering have been revolutionized by these newer disciplines. To say that, collectively, these things have transformed the practice of engineering is to trivialize the issue; they have transformed the world.

However, our world today is characterized, even dominated, not by the singular products which were themselves transformative when they were introduced – electrical power, telephony, automobiles, radio communications, airplanes, semiconductors, computers – but by large scale systems characterized by complex interactions between and among many separate products, elements, and technical disciplines. Advanced societies today are heavily dependent upon systems which cut across the broadest possible spectrum of disciplines.

Most prominently visible in the aerospace, defense, and energy sectors, such systems are characterized by the common trait that they "must work"; failures become societal events and national tragedies. Three-Mile Island, *Challenger*, *Columbia*, the power blackouts of 1965 and 2003 in the northeast United States, and the recent Gulf of Mexico oil spill offer just a few of many unfortunate examples. These sectors are likewise characterized by the necessity to analyze, understand, and predict the "in the large" consequences of decisions, actions, and designs having ostensibly limited scope, but which actually are of profound import, too often seen only in retrospect.

Thus, success in designing, building, and operating efficient large scale systems to serve crucial purposes in both public and private sectors is a defining requirement for societal advancement and economic prosperity in the world of today and in the future. In that world, the elegant integration of technologies is more important, more consequential, than the development of any particular technology or engineering discipline. Whether widely understood or not, these facts have given rise to the discipline of system engineering, another field that did not exist in

1950 and which, from a societal perspective, may ultimately be the most significant of those which have arisen since that time.

It is beyond the scope of the present discussion to recount the growth and formal development of the field of system engineering and its allied discipline, program management, from its origins in the development of complex aircraft, missile, and space systems in the years immediately following World War II. This subject is well treated by Johnson[1] in *The Secret of Apollo*, a "must read" by anyone seeking to understand the history of modern aerospace system development methodology. It will suffice to note here that after sixty years of development there exists today a substantial body of methods, processes, and tools which comprise the discipline of system engineering. Formal academic degree programs exist to train students in this methodology, and it is practiced in both government and industry, where ten percent or more of the cost of a large program may be devoted to the broad category of "system engineering".

II. WHAT IS SYSTEM ENGINEERING?

System engineering as it is taught and practiced is fundamentally concerned with identifying the separable elements or blocks of a proposed design, characterizing the intended relationships between and among those elements, and verifying that the actual configuration is fabricated and operated as intended in its environment. For large complex systems – a modern transport aircraft, a launch vehicle or spacecraft, a power plant, a submarine – this is no small feat, and the methods, processes, and tools which have developed over the last half-century to formalize and systematize it as an essential engineering discipline are not to be slighted. Yet failures continue to occur, often of the most glaring and consequential nature, commonly at the boundaries or interfaces between elements, often due to uncontrolled, unanticipated and unwanted interactions between elements, in many cases between elements thought to be entirely separate.

What is of interest in many of the highly public failures which have occurred in large scale systems over the years are not those instances in which something known to have been needed was simply omitted, or those in which a piece-part simply fails. While significant, such cases are relatively easy to understand and correct. What is of interest are those cases, all too many, in which everything thought to be necessary to success was done and yet, in the end, the system did not perform as intended; in a word, it failed. It is these cases that should cause the system engineering community to ask whether something is missing, whether the discipline remains incomplete in ways that are substantive and meaningful rather than mere matters of detail.

Petroski has offered numerous thoughtful essays[2,3] on the role of failure in engineering. While drawn largely from examples provided by the civil engineering community, the broader principles are equally applicable to the system engineering domain. A key theme of Petroski's work is that a detailed understanding of the manner in which a given design fails in application allows iterative improvement of successor designs. What, then, are the lessons to be learned from failures of complex systems?

One observation is that failures of system engineering process have in the past typically resulted in the addition of more, and more detailed, process. In the world of 1950, when system engineering was largely non-existent – indeed, the term had not yet been coined – and the methods, processes, and tools which characterize the discipline today had yet to be invented, this would likely have been the right answer in any given case. But in the world of 2010, it is this author's view that the addition of more or new system engineering processes is likely not the right answer in response to any particular failure. It puts one in mind of the jocular definition of insanity: continuing to do the same things over and over, while expecting a different outcome. If system engineers are to achieve success through failure, the community must be prepared to do something different.

In the author's opinion, what is needed is a different view of the core system engineering function; specifically, that this core function is *not* primarily concerned with characterizing the interactions between elements and verifying that they are as intended. In a word, system engineering is not fundamentally about "process", except in cases where such process is clearly lacking, a state which characterizes very few large scale projects today. System engineering is about something more. Let us consider what this might be.

While at its core system engineering is concerned with the interfaces between and among separable system elements, it should be realized that the more important understanding concerns the dynamic behavior of the interactions between these elements, not the numbers in the associated Interface Control Document (ICD). As Gentry Lee has put it, "it's about the partials, not the values"[4]. Properly understood, system engineering is concerned with context over structure, with interactions over elements, with the whole over the sum of the parts.

When this is understood, it becomes immediately apparent that the system engineering *process* bears the same relationship to system engineering that financial *accounting* does to financial management. Careful financial accounting is an essential element of a good financial management plan; however, accurate accounting cannot distinguish between a good plan and a poor one, and it cannot make a bad plan better. Similarly, understanding and control of system interfaces, development of comprehensive test and verification plans, and proper allocation of requirements are among the things which are crucial to good system engineering; however, they do not help to distinguish a good design from a poor one, nor can they make a poor design better.

The concerns expressed here are not new. In a landmark 1969 paper for the system engineering community, former NASA Administrator Robert Frosch noted that[5],

> "I believe that the fundamental difficulty is that we have all become so entranced with technique that we think entirely in terms of procedures, systems, milestone charts, PERT diagrams, reliability systems, configuration management, maintainability groups and the other minor paper tools of the "systems engineer" and manager. We have forgotten that someone must be in control and must exercise personal management, knowledge and understanding to create a system. As a result, we have developments that follow all of the rules, but fail." …
>
> I can best describe the spirit of what I have in mind by thinking of a music student who writes a concerto by consulting a checklist of the characteristics of the concerto form, being careful to see that all of the canons of the form are observed, but having no flair for the subject, as opposed to someone who just knows roughly what a concerto is like, but has a real feeling for music. The results become obvious upon hearing them. The prescription of technique cannot be a substitute for talent and capability, but that is precisely how we have tried to use technique." …
>
> We have lost sight of the fact that engineering is an art, not a technique; technique is a tool. From time to time I am briefed on the results of a systems analysis or systems engineering job in a way that prompts me to ask the questions: 'That's fine, but is it a good system? Do you like it? Is it harmonious? Is it an elegant solution to a real problem?' For an answer I usually get a blank stare and a facial expression that suggests I have just said something really obscene."
>
> We must bring the sense of art and excitement back into engineering. Talent, competence, and enthusiasm are qualities of people who can use tools; the lack of these characteristics usually results in people who cannot even be helped by techniques and tools."

With these words, Frosch is holding the system engineering community to a high standard; i.e., the end result of its work is not to be the satisfaction of possibly complex but ultimately well defined requirements and processes. A larger, more holistic goal – an elegant design – is to be sought. It is this author's opinion that Frosch is entirely correct.

"Elegance" in engineering design is an ineluctable concept; it is immediately apparent when it exists, yet it is difficult to define, impossible to quantify and, so far, apparently incapable of being taught. Yet, no aeronautical engineer and no pilot need be taught that the DC-3 was an elegant design, while the Ford Tri-Motor was not. It is offered here that, properly understood, system engineering at its core is concerned with attaining elegant designs.

III. WHAT IS AN ELEGANT DESIGN?

If system engineering is to be fundamentally concerned with pursuing elegant designs and rejecting those that are not, then it is necessary to parse the term more carefully, to define the attributes of an elegant design, to learn how to quantify and rank these attributes, to understand what actually has failed when a complex system "fails", and how it might be improved in a later iteration. The present work constitutes such an attempt.

Among the possible attributes that might be ascribed to an elegant design, at least four are germane to the practice of system engineering.

The first of these is the most basic: does the design actually work? That is, will the system produce the anticipated behavior, the expected output, over the expected range of input conditions, control variations, etc.? This question, relatively simple to answer for a beam or an amplifier or an airplane wing, can be intractable for a complex system. Brute force testing – exhaustive input-output measurements across the entire operating range – is not the answer. Not only do modern systems with non-linear digital elements quite likely have more possible state configurations than can be examined in any reasonable time, it is usually the case that not all possible configurations can be reached in the test environment, or would be safe if they could be reached.

Thus, it must be asked: upon what grounds is it believed that a proffered system design will work as intended? Who understands, in the large, how it works, and why it is believed that it will work that way? If not the chief system engineer, then whom should it be? It would seem that in forty years we have not gotten beyond Frosh's observation that someone "must exercise personal management, knowledge, and understanding". It would seem that academic research into the question of how it is that we might know whether a complex system produces the anticipated output for a given input remains lacking.

If a given design is believed to "work", it must then be asked whether it is "robust", by which it is meant that the system should not produce radical departures from its expected behavior in response to small changes to its operating input, internal state, or external environment. It should degrade gradually and gracefully in response to component failures, changes in its operating environment, or when design loads are exceeded. Briefly, a robust system will not surprise us. When a tree branch falling on a power line can cause a blackout of the power grid in the northeastern United States, the system is not robust. When a piece of insulating foam can cause the loss of a Space Shuttle, the system is not robust. But again, while confidence in the robustness of certain systems can be gained through experience and intuition, and while fragile systems can be identified after the fact of failure, quantifiable measures of robustness do not exist.

Sociologist Charles Perrow addresses the topic of system robustness without specifically employing the term in his seminal work, *Normal Accidents*[6]. Perrow argues that complex, tightly coupled system designs – he makes much use of the Three Mile Island and Space Shuttle Challenger disasters – are pre-disposed to exhibit divergent behavior in response to relatively minor problems. The resulting catastrophe is, to Perrow, a "normal accident"; it is to be expected.

But to advocate that engineered systems should not be "tightly coupled" is too simplistic. In designing systems, it is a truism that we should strive for "simple interfaces", to make the system easier to understand and to limit the possibilities for unintended interactions. But equally important is the response of the system to unexpected variations in its input or environment; i.e., "faults". Simple interfaces by themselves do not address the propagation of faults across and among those interfaces. If a system is to be robust, what is needed is the equivalent of a "ripstop fabric" in the design.

At present, however, the system engineering community lacks a general theory to deal with these issues. It lacks quantitative measures for what constitutes "tight" coupling among system elements, it lacks knowledge of what an appropriate threshold might be if quantitative measures were to become available, and in any case it is frequently true that unintended, unanticipated, and possibly unanticipatible, interactions between system elements which were thought to be isolated are the root cause of a problem. It is hard to know how a system can be designed to be resilient in the face of "unknown unknowns". Finally, in systems where weight and volume are at a premium – aircraft, spacecraft, submarines – it is difficult to imagine how "tightly coupled" system designs can be avoided. Yet, it remains true that some system designs are more robust than others, and some system designers know how to enhance this property while others do not. Work is needed to understand how this can be done more purposefully.

Attempts have been made along these lines. While Perrow does not offer much in the way of praise for "safe modes" or "fail safe" design approaches, such concepts – along with circuit breakers, dampers, and limiters – can and do work; they have saved many complex spacecraft. As the theory of adaptive control systems continues to mature, it should enable system stability across a wider range of operating conditions, thereby reducing accidents. With these and other techniques, it may be possible to attain optimal, tightly-coupled designs that are also robust.

A third property of an elegant design is that it is efficient; it produces the desired result for what is thought to be a lesser expenditure of resources than competing alternatives. But how is this to be evaluated? Design studies conducted for the purpose of selecting a preferred system concept from among a set of alternatives are notoriously poor in regard to their ability to predict cost, schedule, and performance for a finished system. The system engineering community's ability to differentiate between and among competing designs, based upon such cost, schedule, and performance predictions, is even poorer. As-built prototype designs can of course be compared, but competing designs which were not selected are generally not built. "Fly offs" are rare in aeronautics and essentially non-existent in most other arenas where complex systems are needed. Yet without direct comparison, the question of relative efficiency is difficult to answer. An alternative concept not being built will almost always appear preferable to a system which is actually being built, and is in the throes of typical engineering development problems.

Adm. Hyman Rickover noted this aspect of complex system development programs in his famous discussion of the difference between paper reactors and real reactors[7]. When confronted with a situation in which a variety of alternative concepts were being advocated in place of the pressurized-water reactor design he favored for the nuclear navy, Rickover noted that there were two kinds of reactors, "paper reactors"; i.e., new reactor concepts, and "real reactors". A paper reactor has the following characteristics:

- It is simple.
- It is small.
- It is cheap.
- It is lightweight.
- It can be built very quickly.
- Very little development is required; it can use off-the-shelf components.
- It is in the study phase; it is not being built now.

In contrast, a real reactor has the following characteristics:

- It is complicated.
- It is large.
- It is heavy.
- It is behind schedule.
- It requires an immense amount of development on apparently trivial items.
- It takes a long time to build because of its engineering development problems.
- It is being built now.

All experienced practitioners of engineering development are familiar with this phenomenon, which is as much social and cultural as it is technical: "the grass is always greener on the other side of the fence".

With that noted, it remains true that the system engineering community requires greatly improved means to compare alternative design concepts prior to selection for development. As Norm Augustine has famously noted[8], "ninety percent of the time things will turn out worse than you expected", reflecting the fact that only ten percent of development programs are completed for the advertised cost. Of this performance, it can only be said that mediocrity would be an improvement.

A fourth attribute of an elegant system design is that it accomplishes its intended purposes while minimizing unintended actions, side effects, and consequences. Among these might be wasted energy in the form of heat, noise, or vibration, electromagnetic interference or other undesired interactions with other systems and subsystems, or pollution. Control of these effects is a more subtle facet of system engineering, less noted but of considerable significance. It will often be observed that nearly all of a designer's attention is focused upon producing a design which accomplishes its intended purpose. However, upon further thought it may be recognized that, once a concept has been selected, most systems do in fact accomplish the basic purposes for which they were designed. A key differentiator between better designs and poorer designs is to be found in the unwanted features and effects produced by poorer designs, and the actions which must be taken to compensate for them. Even designs which "work", which are robust in the face of varying input and environmental conditions, and which are reasonably efficient in comparison with competing designs, may be found to produce a variety of unintended and unwanted side effects.

As noted above in connection with the system engineering community's inability to rank the efficiency of competing designs prior to making a commitment to a particular concept, so too the community lacks the tools to evaluate the unintended consequences produced by a given design. While the properties of workability, robustness, and efficiency can be examined by means of simulation or the fabrication of prototype systems, this is not the case where the evaluation of unintended consequences is concerned. Experimental or prototype systems are generally not intended to be faithful to the final system design in regard to precisely the ancillary characteristics that may be troublesome for a system in actual use. Often the troublesome effects are noticed only in the long term, or are statistically rare but are found, in retrospect, to be of considerable import when they occur. Further, it is a rare modeling and simulation tool which provides any analysis of potential side effects in addition to the primary purposes for which it is developed, unless that analysis is specifically requested by the system engineer. Often such questions are not even asked.

IV. HOW CAN WE ATTAIN ELEGANT DESIGNS MORE SYSTEMATICALLY?

Additional attributes of design elegance may well exist; however, the present author contends that any design which is believed to possess that elusive property will be thought to possess at least these four attributes. A key theme of the present work is that few in the system engineering community recognize that the responsibility for ensuring that a system design possesses these qualities lies with system engineers. It is advocated here that this should be their core concern.

A second theme of this work is that, lacking quantitative means and effective analytical methods to deal with the various attributes of design elegance, the development of successful complex systems is today largely dependent upon the intuitive skills of good system engineers. System engineering today, with regard to the key attributes of design elegance identified here, is in a position analogous to that of civil engineering prior to the development of the theory of strength of materials. Structures were designed and built according to “rules of thumb” which were derived from prior successes and failures, and passed down from master to apprentice. Good structures were those that did not fall down.

System engineers today are in this same position; good systems are those which do not fail. But today, we have a sophisticated theory of the strength of materials, and of structural design generally; we know how to design optimal structures for a wide range of user-specified optimality criteria. In system engineering, we do not have a corresponding theory for the design of optimal systems. We do not even have an accepted definition of what is meant by an “optimal system”. Such things are, by and large, matters of intuition and judgment.

While it is this author’s view that the role of human intuition, of “engineering judgment”, will not soon be replaced by algorithms and analytical methods, it would be well for the system engineering community to develop better and more appropriate theories, tools and methods to augment that judgment. To do this, advances will be required in at least four broad areas.

The first of these necessary advances must occur in the academic community. Most of the research required to go beyond rule-of-thumb approaches to the design of elegant complex systems will be performed within the academic community. But at the same time, it will be essential to involve the academic community in a new and different way.

The academic community is organized to, and provides strong incentives for, discipline-oriented research. Faculty are hired and rewarded for their propensity to obtain and conduct sponsored research in, generally, quite narrow areas of specialization, which usually are narrowed further as the success and prestige of a researcher’s career are advanced. Funding is allocated by most research sponsors along discipline lines. As a consequence, multi- and cross-disciplinary studies are generally discouraged, especially for the creative, young, tenure-seeking faculty which would normally be expected to contribute the most to the development of a new arena. “Academic silos”, and the rewards for remaining within them, are alive and well in the preponderance of engineering institutions.

But if academic institutions and traditions, and academics themselves, are to remain relevant, they must address the real problems the world is facing. We in academia must be prepared to devote our efforts to those problems, rather than others which might be more traditional or comfortable, and we must develop and train the students who can solve them. If we are to make critical contributions to the critical challenges which will shape our future, we must understand the interaction between and among different technologies, organizational management, contract management, public policy, national politics, funding strategies, and the role of these interactions in the behavior of large scale systems.

If we are to remain relevant, if we are to continue to influence policy and process, we must learn to treat the interactions between disciplines with the rigor we have, over centuries, learned to apply to the disciplines themselves. We must learn to treat context itself as a discipline. We are not yet doing that, and we certainly are not teaching it. Today, we are training system engineers to be the engineering equivalent of financial accountants. We are not teaching them to be Chief Financial Officers.

The second area where improvement must be realized lies within the community of engineering practice, which also bears responsibility for the present state of system engineering. In the real world of engineering development, projects are managed to fruition through a variety of organizational, legal, and contractual tools, many of which are antithetical to the fielding of an elegant design.

Experienced practitioners are deeply familiar with requirements-based design, the bedrock of engineering and management of large systems. For example, in traditional requirements-based design, it would be customary to specify both the maximum allowable cost and mass of a given subsystem. Such requirements are hard constraints; in theory they cannot be exceeded, although in practical systems it is often found to be otherwise. Nevertheless, it is in this manner that the "value" which is placed on these quantities by the customer is provided, through the system engineer, to the design team. The team accepts these constraints, and generally strives to produce designs which fall within them.

However, it is possible to describe the "value" associated with various system properties – mass and cost in this case – by other means. In classical optimization theory, these properties would be associated with appropriate "weights", selected according to the relative importance assigned to those attributes. In turn, the weighted attributes are combined analytically to yield an "objective function"; the optimal design is that which produces the minimum value of the objective function. Note that there need be no "hard constraints" in the objective function, though these can be included if necessary. More typically, however, the attribute weights and the form of the objective function are chosen to penalize more heavily those designs which utilize too much of one resource or another, yet without imposing hard limits on such use.

Of interest here is that Collopy has shown[9] that strategies designed to impose firm requirements which are allocated to individual system elements yields, on average, results inferior to those obtained using optimization techniques. This result runs counter to both intuition and established practice. Firm requirements and a logically consistent flow-down of those requirements, with carefully established parent-child relationships, are at the core of present-day system engineering practice. Yet theory suggests that such practices produce results inferior to those which would be obtained using alternative methods.

Proponents of requirements-based system design and engineering will note that firm, documented requirements are essential to the management of contracts. This may well be so; however, if we are to improve the practice of system engineering, it may be that legal contracts as presently structured do not constitute the best tool for the management of engineering development projects. Present methods of system management tend to focus attention on what is required, rather than what is important; i.e., "doing things right" vs. "doing the right things". The community of practice must develop management methods that favor the development of elegant system designs, rather than just legally defensible designs.

A third area where advances in system engineering are required concerns the structure and operation of the engineering design team itself. Whether or not the paradigm of the "lone inventor" is or ever was reflective of reality, that paradigm is the antithesis of modern system engineering, which is intrinsically a team effort. Accordingly, any comprehensive theory of system engineering must include the study of the human interactions which lead to the final result of that effort. Even a cursory study of such interactions is sufficient to show that the organization and operation of the engineering team itself has a profound influence on the finished product of that team.

One interesting example follows from Arrow's Theorem[10, 11] and the theory of social choice. Arrow's Theorem shows that, under very general conditions, decisions involving three or more options which are made by means of pairwise comparisons can lead to paradoxical outcomes. Common examples may be found in democratic elections where there is a significant "third party" candidate. In the U.S., recent examples include Bush-Clinton-Perot in 1992, or Bush-Gore-Nader in 2000; in each case, the actual winner of the election was rejected by a majority of those who voted.

Such paradoxes in the selection of alternatives were first studied in the 18th Century by French Academician J. C. Borda[12], who also proposed a decision rule which avoids them. In Borda counting, voting is not pairwise; it is not "either-or". If there are, for example, three alternatives, each voter expresses a preference for all outcomes, with (2, 1, 0) points assigned respectively to the voter's (1st, 2nd, 3rd) choice. The decision rule is then to select the alternative which receives the highest point total. In this scheme, in contrast to that of pairwise voting, no information is rejected; the ranking of all alternatives by all participants is included in the decision rule. However, to the best of this author's knowledge, Borda counting is not used as a decision tool anywhere in engineering. Indeed, it may be observed in practice that most engineering tradeoff decisions are made through sequences of pairwise comparisons -- exactly what Arrow's Theorem shows cannot work.

This result has significant implications for how we as system engineers structure our design teams and management plans. However, until and unless we begin to delve into the social and cognitive aspects of how engineers work together and how system engineering is performed, these and other possible contributions from fields far apart from engineering will continue to go unrecognized.

Once this is understood, other questions arise; e.g., what is the role of personality type? It is readily observed that many technical professionals excel at, and take great satisfaction from, the acquisition of great expertise in particular disciplines. The value of such expertise in designing and developing any complex system does not require further discussion. Similarly, it will be observed that some whose early work is established in a given discipline will later pursue technical and career breadth rather than ever-increasing depth. This sort of expertise is rarely recognized, particularly in academic circules, yet it is from this latter population that system engineers are generally drawn.

It may be observed that, today, it is easier to identify good system engineers than it is to define good system engineering. The question then naturally arises: can system engineering, with the more holistic view required to pursue "design elegance" as considered here, actually be taught? Can system engineers be trained? Or must they be identified? Is there a useful middle ground?

If system engineering is at its core a holistic discipline, if it is to be fundamentally concerned with design elegance, and if breadth of viewpoint is important to attaining that goal, then what is the role and value of social, cultural, age, and gender diversity on the large design engineering teams which are required for the successful development of complex systems? Quite apart from the overtly political way in which the importance of "diversity" is generally couched, is there real, quantifiable, engineering value to be had by purposefully "designing the design team" with the inclusion of diverse views as a goal?

This brings us to the fourth area where advances are required if the understanding and practice of system engineering is to be improved. The questions raised here are questions of social science and cognitive psychology as well as of engineering, and the answers will matter to society at large as well as to engineers. To obtain these answers, it will be necessary to reexamine our existing research establishments and paradigms.

In the decades since World War II, scientific and technological advancement in the United States and, more broadly, the western democracies has followed a paradigm in which government sponsorship of academic research leads to powerful industrial applications. This paradigm has served us very well indeed. In the author's opinion, it has worked as well as it has in part because the relevant questions in most individual engineering disciplines are understood, and understandable, by all parties. Academic researchers in a given discipline understand what their counterparts in government and industry do in that discipline.

The same is not true of system engineers and system engineering. Academic researchers and research teams are rarely, if ever, exposed to the actual practice of system engineering as it occurs on a major development program. Similarly, few if any successful practicing system engineers have either the inclination or, frankly, the classically required academic credentials necessary to make a transition to academia. Finally, it is rare or non-existent to find graduate students – the lifeblood of academic research – involved in a significant way in the development of large, complex systems. Because these necessary conditions have not been present in the development of system engineering as they have in other engineering disciplines, the rate of advancement has been correspondingly less.

If, as claimed here, our success as a society is increasingly dependent upon the mastery of complex system design, and if we believe there is progress to be made in system engineering through the use of the same government-academia-industry coupling that has served us so well in other areas, then it will be necessary to find ways to expose academicians and their students to the real problems confronting those who actually practice system engineering, and to find ways to expose practitioners to the useful results which will arise from properly focused academic research.

V. CONCLUSION

Properly understood, the core purpose of the discipline of system engineering, and the primary responsibility of the system engineer, is the fielding of an elegant design. As discussed here, an elegant design is one which produces the intended result, is both robust and efficient, and generates a minimum of unintended consequences. Considerable work remains ahead to define and quantify these attributes in ways which are meaningful to the engineering community, and to develop methods and tools by which these goals may be attained.

To accomplish these results, it will be necessary to enlarge the view of academia to include the interdisciplinary and multidisciplinary studies required for both research and education in system engineering. It will likewise be necessary for the community of practice to develop methods to incorporate the results of improved theories of system engineering in the management of actual projects and programs. The study of human interactions, cognitive psychology, social choice theory, and other disciplines must be included in the development of effective theories of system engineering. Finally, there must be closer collaboration between those who practice system engineering as it occurs in the real world, and those who will teach and study it.

If we can do these things, we may ultimately be able to produce designs which meet Frosch's standard of "elegance" in a more systematic manner.

REFERENCES

1. Johnson, Stephen, *The Secret of Apollo*, Johns Hopkins University Press, Baltimore, Maryland, 2002.

2. Petroski, Henry, *To Engineer is Human: The Role of Failure in Successful Design*, Vintage Books, New York, 1992.

3. Petroski, Henry, *Success Through Failure: The Paradox of Design*, Princeton University Press, Princeton, New Jersey, 2006.

4. Lee, Gentry, Personal Communication, Jet Propulsion Laboratory, Pasadena, CA.

5. Frosch, Robert A., "A Classic Look at Systems Engineering", Speech to IEEE Group on Aerospace and Electronic Systems, IEEE International Convention, New York, 26 March 1969. Reprinted in *Readings in System Engineering*, NASA SP-6012.

6. Perrow, Charles, *Normal Accidents: Living with High-Risk Technologies*, Princeton University Press, Princeton, New Jersey, 1999.

7. Rockwell, Theodore, *The Rickover Effect*, John Wiley & Sons, New York, 1995.

8. Augustine, Norman, *Augustine's Laws*, 6th ed., American Institute of Aeronautics and Astronautics, 1997.

9. Collopy, Paul, "Adverse Impact of Extensive Attribute Requirements on the Design of Complex Systems", AIAA 2007-7820, AIAA Aviation Technology, Integration, and Operations Conference, Belfast, Northern Ireland, 18-20 Sep 2007.

10. Arrow, Kenneth J., *Social Choice and Individual Values*, New York: John Wiley and Sons, 1953, 1961.

11. Saari, Donald G, *Disposing Dictators, Demystifying Voting Paradoxes: Social Choice Analysis*, Cambridge University Press, New York, 2008.

12. Borda, J. C., "Memoire sur les elections au scrutiny", Histoire de l'Academie Royale des Sciences, Paris, 1781.