On The Necessity of a Deep Paradigm Shift in Systems Engineering

Introduction

Kent Palmer Ph.D.
kent@palmer.name
http://kdp.me
714-633-9508
Copyright 2014 KD Palmer
All Rights Reserved. Not for Distribution.
OnTheTheNecessityOfAParadigmShiftInSE_01_20140816kdp02a
Started 2014.3.23, edited 2014.8.16 Draft Version 02
http://orcid.org/0000-0002-5298-4422

Key Words: Systems Engineering, Systems Science, Software Engineering, Design Science, Schemas Theory, Paradigm Change, Synergistic Engineering, SE 2.0., Philosophy of Science, Philosophy of Engineering, Systematicity,

Abstract: Systems Engineering is poised for a deep Paradigm Shift in the discipline. This paper will explore the necessity of that Paradigm Shift and some alternatives regarding what the nature of that Shift may be.

Systems Engineering is a new academic discipline with many new Masters and Ph.D. programs having been established recently in various academic institutions. However, this shift from being a merely Industrial classification of workers to an Academic discipline has brought to the fore the crisis in foundations within the Systems Engineering enterprise itself. This is because to be an Academic discipline in good standing there must be valid knowledge for such a discipline to teach and this brings up the crucial question as to what the teachable knowledge foundations that Systems Engineering might claim for itself within the Academic arena where it is in competition with other specialist disciplines with established knowledge foundations. This leads us to the fact that Systems Engineering has recently published its Body of Knowledge, which is an attempt to write down the accumulated wisdom that has proved useful in practice of the Systems Engineering role in Aerospace contexts as determined mostly by standards imposed by the Department of Defense, where conformity to imposed practices was the measure of Successful execution on Defense contracts. In other words the nature of Systems Engineering as captured in the Body of Knowledge was not the result of free competition of ideas but conformity to the expectations of a single customer base who represented the sovereign, i.e. the one entity that one could not stop working for until one was released from the obligation to continue. What is good about the

Body of Knowledge is that it captures the received wisdom promulgated by the largest corporations who have responded to Government RFPs on how Systems Engineering should be done by the people involved. But in that sense it only represents the baseline from which the discipline must begin to evolve which is currently the entrenched basis for practice within corporations under contact with the DOD. But if we consider the Discipline itself outside its relation to its major customer and within the context of Academia as more than merely a source of professional training for career enhancement. If instead we consider it as an Academic discipline in its own right what we notice is that it is without a basis of practice based on bonified validated knowledge foundations. In the Body of Knowledge it is acknowledged for the first time that Systems Science may be that missing foundation. Systems Science as represented by organizations such as the ISSS.org has a 60 year history of the study of systems to offer Systems Engineering. But there are almost no Systems Engineering programs that teach Systems Science at this time. Hopefully this will change in the near future since it is now a part of the Body of Knowledge. And so one of the challenges that the Systems Science Working Group under the leadership of James Martin has undertaken is to attempt to teach Systems Engineers about Systems Science by developing courses and by adding to the Body of Knowledge in this area. Thus there is a recognition that just as Systems Science is a meta-discipline above the Special Disciplines of Science so to Systems Engineering is a meta-discipline above the Special Engineering Disciplines and there is much that Systems Science can teach Systems Engineers about the nature of Systems that for the most part they are ignorant of due to lack of knowledge transfer in this regard. The agreement between INCOSE.org and ISSS.org to exchange member representatives and promulgate Systems Science information to Systems Engineers as best we can given the disarray of the Systems Science body of Knowledge is historic, and probably one of the most fundamental steps toward the acquisition of a Systems Science basis for Systems Engineering that has been undertaken to date. Also of note is the work of Len Troncale, a past president of the ISSS.org within the INCOSE working group on Systems Science to bring the accumulated knowledge of how systems work in nature to make it available to Systems Engineers through the concept of the Systems Isomorphies (Processes and Patterns) and the Linkage Propositions between these Isomorphies. What we can learn from the way nature puts together Systems that can be applied though Systems Mimicry in Systems Engineering can only enhance our knowledge base within Systems Engineering.

However, these positive steps are not enough to face the multiple crises that our Country and indeed the world faces of a Systemic nature. Systems are becoming more complex, and have broader scope in their influence on the lives of citizen's everyday, and the reliability of those infrastructure systems is becoming paramount to our maintaining our lifestyle and superiority within the world economy during globalization. Thus there is a deep crisis in Systems Engineering as a discipline exacerbated by the impending retirement of a large part of the Systems Engineering practitioners as the Baby Boomers age. So what is necessary is a deeper look at foundations and it is always the case that when we look deeper into the foundations

of a discipline the chances of discovering something that causes a paradigm shift grow more likely. Since the multiple problems that confront Systems Engineering as a discipline and our civilization as a whole are not going away. In order to respond to these challenges our discipline will have to change in fundamental and probably unexpected ways. It behooves us to look out for possible paradigm shift candidates and to attempt to nourish those candidates so that we can more effectively evolve our discipline the way other disciplines evolve. For instance, our sister discipline from which many of the Systems Engineering approaches are drawn which is Software Engineering has undergone major paradigm shifts throughout its existence. It started off using imperative functional languages like C and then developed these into object oriented languages like C++ and problems with objects has given rise to Aspect Oriented programming, and more recently to truly Functional programing languages like Haskell. This has been accompanied by many changes in Software like the Agile and Lean approaches to project execution that have been recently introduced. No comparable fundamental change has occurred in Systems Engineering as a discipline. And this is basically because the nature of Systems Engineering is dictated by standards and the practitioners and not free to experiment and define their own fundamental approaches to Systems Engineering the way Software practitioners and Computer Scientists are able to do. Another problem is that Systems Engineering has no corresponding Scientific Discipline that has its own department, unless we count Industrial Engineering which is attempting to absorb Systems Engineering the way that Computer Science absorbed initiatives to have independent Software Engineering masters degrees. There are Systems Engineering Masters and Ph.D. degrees but many times these are spinoffs from Industrial Engineering departments which have traditionally addressed the Systems of Systems Problems in Industry, just as Systems Engineering is attempting to address them in Aerospace contexts now. So from the Systems of Systems side of Systems Engineering there is a congruity between Industrial Engineering and Systems Engineering. But Industrial Engineering is a fundamentally analytic discipline like other Specialist Engineering disciplines and thus is not founded on the idea of the System as an ideal basis of approach to the problem, and thus Systems Engineering curricula sponsored by Industrial Engineering is a hybrid of traditional approaches and new holistic or synthetic approaches championed by Systems Engineering.

So it could be that Systems Engineering will just be absorbed into established Industrial Engineering departments and lose its integrity in Academia the way that Software Engineering was absorbed into Computer Science. But a key difference is that while Computer Science was a free discipline to develop whatever methods worked for it and thus has gone through multiple paradigm chances, Systems Engineering if it finds its foundations with regard to Systems in Systems Science does not have the same sort of Academic counterpart, because Systems Science is interdisciplinary and has been carried on for the most part in institutes rather than departments within Universities. And for this reason the knowledge accumulated by Systems Science is spread out in the work of the major figures in Systems Science over the past 60 years or so and is not codified into textbooks or unified in the way

that the knowledge of specialist disciplines normally are codified in a teachable and learnable form. This produces severe problems when we want to teach this information to Systems Engineers. Systems Science needs to be organized and codified into its own Body of Knowledge that is readily transferable to others who want to use that knowledge as a basis for practice. So in a sense Systems Science has its own crisis as a discipline as success has led to dispersal of energy in many different Systems Societies such that there is a Federation of Systems Societies. And many of the newest versions of Systems Science have established their own societies and conferences to promulgate knowledge, for instance, of Complex Systems. So though ISSS.org is the oldest of these Societies, it does not necessarily represent the cutting edge of developments in the Systems Science realm. Thus there is a separate Complexity Science working group within INCOSE to address the newer versions of Systems Science separately from the Systems Science working group. Both of these working groups provide a forum for practitioners to catch up with developments in other disciplines that bear on the understanding of large scale complex systems and also the theoretical basis of systems theory itself.

However, this grounding in Systems Science and Complexity Science is still not enough and we must consider various possible paradigm shift candidates that will transform our discipline of Systems Engineering in an even deeper way. Part of that is the recognition of the close ties between Software Engineering and Systems Engineering and the fact that we should probably talk of a hybrid discipline which I call Software Systems Engineering. And this is because most of the innovations in processes and methods used by Systems Engineering come ultimately from the fertile ground of the paradigm changes in Software Engineering and the consolidated research programs of Computer Science. On the other hand Systems Engineering needs closer ties with the Social Sciences because all the systems that are built by Systems Engineering are for the support of and interact with humans and Human Systems Integration is just not enough, there needs to be real understanding of not just human factors but of human psychology, political science, economics, sociology, anthropology, cognitive science and other social science disciplines. Engineers who become Systems Engineers have traditionally prided themselves on the fact that they avoided the Humanities and the Social Sciences during their educational careers. But in their actual careers in industry they end up being intensively involved in social situations with psychological and political ramifications. Thus more needs to be done to bridge between Systems Engineering and the Social Sciences as well as with Software and Systems Science. Thus basically we see that Systems Engineering needs to reach out and become acquainted intimately with these other disciplines that can help it with its mission of taking responsibility for the whole system, but which increasingly is made up of Software, and has a Social component that is more and more important on the basis of knowing more about the nature of Systems as discovered by Systems Science as exemplified in the work of Len Troncale and George Klir and other Systems Scientists.

But these directions toward other disciplines does not help us to understand what is

unique about Systems Engineering which will provide its foundations, because interestingly enough Systems Science by itself is not enough. And so this is where we introduce a candidate paradigm change for Systems Engineering called Schemas Theory. This is just one candidate and we need to foster many candidates to that the real paradigm change will be recognized against the crisis in the discipline of Systems Engineering that is mostly unconscious at this time for most of the practitioners and leaders in the field. The more candidate paradigm shifts that can be explored the more likely that a deep paradigm shift will occur within the discipline and the more likely it will become a real discipline like Software Engineering with a real history of multiple paradigm shifts. As it stands Systems Engineering is quite virginal in the sense that it has not even undergone one paradigm shift in its existence, but is instead been reined in by contractual arrangements in which the customer has dictated the form of the practice by standards. This is slowly changing with the new 5002 Capstone Systems Engineering document that attempts to change how Systems Engineering is viewed in the acquisition process. However, paradigm changes cannot be willed by even as big a customer as the DOD because the crisis of Systems Engineering is intrinsic to its nature and that nature cannot be dictated by the Sovereign no matter how much it tries because it flows from fundamental contradictions within the discipline itself and its lack of a clear knowledge basis that can make it a respected Academic Discipline in its own right. Now that Systems Engineering is within Academia new pressures will be exerted on it which for instance lead to the Graduate Model Curriculum as an adjunct to the Systems Engineering Body of Knowledge. What we shall teach is determined by this consensus but that does not mean the consensus represents real knowledge in the traditional sense produced by other disciplines in the course of their development and refinement. There is validation for the information contained in the Systems Engineering Body of Knowledge which merely elevates the received wisdom of the practice to a body of knowledge to be passed down in an academic setting which merely amounts to career training at this point for most programs that have been recently set up.

So here we will briefly outline one example potential basis for a paradigm shift within the discipline. This exemplar is mostly meant to show what a paradigm shift might be like as opposed to the various efforts for refining the discipline already mentioned that do not qualify as a paradigm shift as first defined by Thomas Kuhn and refined by Martin Friedman in Dynamics of Reason. In a separate paper I have compared the Schemas Theory change in perspective to the criteria of Friedman and there we see that Schemas Theory has the characteristics that Friedman outlines for paradigm shifts. Here we merely mention the broad outlines of the nature of the proposed paradigm shift. And this revolves around the idea that whereas systems are used within the various disciplines in different ways Systems Science is a meta-discipline which is interdisciplinary and transdisciplinary between and above Sciences which is a similar position that Systems Engineering has with regard to the various Engineering disciplines. So there is a natural logical affinity between these two meta-level disciplines one in Science and the other in Engineering, and both of them decidedly outliers with regard to comparison with normal special disciplines

and special engineering approaches that are well established. However, to affect our paradigm Shift it is necessary to go to the next higher level of emergent abstraction or logical hierarchical type which is above Systems Theory. Unfortunately this direction of the expansion from Systems Theory has not been explored except tenuously previously for instance in the work of George Klir in Architecture of Systems Problem Solving (ASPS) where he creates a formal structural system model combining schemas. Schemas Theory is just a possibility unless we have a hypothesis that renders it concrete. Therefore I developed the S-prime hypothesis that says that there are ten specific schemas at different scopes (facet, monad, pattern, form, system, meta-system [openscape], domain, world, kosmos, and pluriverse). These are the only schemas that exist by the S-prime hypothesis and they are nested with each other with no gaps in our intrinsic understanding of spacetime from a human perspective. Each one is a different organizational template that is emergent and captures a certain way of looking a phenomena. But the key to the S-prime hypothesis is that it links these ten schemas with a rule that says that there are two schemas per dimension and two dimensions per schema. Thus the schemas range from -1 to the 9th dimension and each dimension is related to two schemas within the hierarchy of scopes of the schemas within our experience as projected as a priori syntheses in the manner imagined by Kant in his equation between Transcendental Realism and Transcendental Idealism that eventually gave rise to the Phenomenology of Husserl. The history of the Naïve Realism of Analytical Philosophy in contrast to the Anti-realism of Continental Philosophy is rehearsed by Lee Braver in his book A Thing of This World. In that book he goes through the evolution of the Kantian Perspective in the Continental Philosophical tradition which really has no counterpart of Analytical Philosophy which remains essentially naively realistic and epistemological in its tenor.

Schemas Theory offers a paradigm change because it widens the field of the schemas to be considered by Systems Engineers beyond the ubiquitously referred to Systems Schema. Many practitioners roll their eyes when someone asks “What is a System?”, and that is because the word System has lost its meaning because it is applied to everything. But when we get to the level of the Schema then there we find different schemas besides the Systems Schema that we can contrast it to and use for other purposes which go beyond what the Systems Schema is capable of understanding because in Schemas Theory there is a whole tool box of Schemas available to us which has been developed through the history of Western Civilization. The most important of these is the Form which is seen in Software as Objects with encapsulated data and methods. Prior to the 20th century the Form Schema was dominant and many other schemas were merely implicit and not very well developed. But during the 20th century there were two other schemas that become well defined which was the Pattern schema which was known under the name of Structuralism and the System schema. Of these the Structuralism was the more successful within the disciplines because it lends itself to reductionism, linearity, and analysis which is rampant in the special disciplines. But of course the holism of Systems Science was less well accepted by the disciplines and that is why it was only developed in Institutes rather than given a department status within

academia. Of these two disciplines in the 21st century Systems Science is now finally coming into its own with the advent of interest in the phenomena of emergence and complexity theory within the disciplines as more and more complex phenomena are considered that are non-linear and depart from the safe haven of tractable linearity that yields to analysis via reductionism. Ways of talking about emergent phenomena and complex systems we do not understand in detail but whose behavior is different in aggregate than it is in its instances have been developed and that has basically meant that the Systems Science approach has been widely accepted with disciplines like Systems Biology and Earth Systems being developed within the disciplines with conferences on Complex Systems being held that cut across disciplinary boundaries. The older Systems Science community was thus left behind in a way by these new developments, but yet it preserves the history by which the breakthroughs into Complex Adaptable Systems Theory like that described by John Holland became possible. And Systems Engineering needs to tread that path to bring together the different Theoretical strands of Systems Science including its history of the great minds in this area of endeavor who sought to define what a System was, and the new developments in Complexity Theory. But in order to unify this material a higher vantage point is necessary like that provided by Schemas Theory. Schemas Theory is a means of understanding what is unified about Systems Science and then it also gives a means of evaluating that contribution by contrasting it with other lesser known schemas that are latent in our tradition that have not been so extensively developed but with which most of us are familiar if our attention is drawn to them like Form, Pattern, Domain, etc. We use these terms, but we do not actively use these templates of understanding alongside that of the System to augment our understanding of phenomena. Rather we look at all phenomena as if it were determined by the System Schema. Thus as a discipline it is as if we have blinders on only seeing Systems when in reality we are dealing with multiple schemas every day as part of our practices. The fact that Schemas Theory opens up a vista by which we appreciate the contribution of other schemas to give meaning to the system schema and augment it, shows that it has one of the essential features of a paradigm change which is that it goes beyond yet encompasses by backward compatibility the current paradigm which is to use only one schema to describe everything, while the new paradigm is to recognize the breadth of the other schemas as the ground for understanding and augmenting the systems schema. It is like realizing that we have many other tools in our tool box other than a hammer, and we don't have to use the hammer to do everything. Rather there are also pliers, an awl, a saw, etc. i.e. different tools for different purposes that can be used together to build what we need to build.

The point of Schemas is that there are a set of them that is limited, and they are a priori projections of organization to spacetime tied to dimensionality and thus they suggest that conceptually we deal with higher dimensions and that our short term memory is organized to hold 7 plus or minus 2 INDEPENDENT things, so that we are always thinking in higher dimensionality and that is aided by the schemas as templates for our representations at various levels of complexity. Semantics occurs in this higher dimensional realm much of which is lost when we reduce those higher

dimensional thought processes to two and three dimensional representations of some kind. But also key is that it is Schemas on which we base our designs of artifacts that we produce in the artificial built world, and that nature has other schemas by which it builds things within the environment because there are many different species many of whom have other schemas than those we project. But also Language is organized by the schemas and so the fundamental way in which we understand things through expression in natural language has the same organizational pattern based on the schemas as that we see in things we encounter in experience that later we realize were a priori synthetic projections discovered after the fact to have informed our views of phenomena we encounter. The history of Schemas Theory was captured by Umberto Eco in Kant and the Platypus and within that book what we are discussing here under the rubric of schemas is what he calls the 'Mathematical and Geometric Schemas' which is merely one but the most basic forms of the schemas originally proposed by Kant as a way to relate the Categories that described objects in space to time. Now of course we know that spacetime is a continuum and therefore Schemas all have both spatial and temporal intervals that are fused with each other into a continuum.

So Schemas Theory is a candidate for a Paradigm Shift in Systems Engineering because it says that the Western tradition is much richer than we supposed and that there were more in heaven and earth than was comprehended in our Systems Theory, and that we need theories related to the other Schemas as well like Pattern Theory, Domain Theory, EcoSystem (Meta-system) Theory, etc. in order to fully cover all the various schemas that are used by Science to understand nature, and used by Engineering to build artificial systems, and used in natural language to organize the logos from which springs logic and reason. If the spilt between Logos and Phusis is the fundamental duality in our tradition, and Nomos (Order in the form of Mathematics) stands between them, then we can understand Logic as the Phusus of the Logos inducing the fundamental limits of sense, while the Schemas are the Logos of the Phusis, in other words they are the medium though which the phenomena speak to us via the anomalies which defy the templates of understanding that we naturally project as a species. Science is precisely the process of looking beyond our synthetic a priori projections to see what is there beyond our projections. We have developed ways to get around the fact that we project order on experience that may actually not exemplify the underlying order of the things we experience. But in engineering design we naturally produce designs and constructions of artifacts based on these templates of understanding as we create a posteriori syntheses of products. So Schemas are crucial to our engineering design efforts, and we delve into all the schemas in our production processes during development even if we only talk about systems. And we understand the things we build trough natural language both spoken and written whose structure is determined by the schemas as well, thus something within us is structured the same way as we see the things as structured in our experience producing a fundamental congruence between what we see in experience and what we can talk about and understand. But beyond that the schemas have a fundamental connection to mathematics via dimensionality and that might be the biggest benefit of the

Schemas for Systems Engineering, which I call Synergistic Engineering (following the lead of B. Fuller's insights) when it is informed by knowledge of all the schemas based on an understanding of Schemas Theory.

This connection to mathematics is desperately needed in Systems Engineering which as a discipline is weak in its basis in mathematics. Scientific Theories need to connect to phenomena based on the use of mathematics. Special Disciplines use mathematics of various levels of sophistication to study and teach their knowledge in their discipline. Systems Engineering if it is to be a Scientifically based discipline also needs theories and those theories (called Design Science perhaps) need to be based on Mathematics, but a higher form of mathematics than that used by the individual disciplines. And for me the prime candidate for that is Category Theory which has more and more success in organizing our understanding of Mathematics and is more and more being used by fields such as Software Engineering to organize its own understanding especially due to the discovery that the dual of Algebra is Co-algebra which is basically a series of states which can be seen in terms of Programming and Lambda Calculus, a representation of computability. So it turns out that there is a higher dimensional analog of Category Theory recently developed by John Baez and others called N-Category Theory and this is a good basis for attempting to connect to the higher dimensions within which we think according to schemas theory. But this is not all. If Einstein had not had Reimann geometry he could not have expressed concretely the general theory of Relativity. It takes advances in mathematics to support paradigm chances in disciplines like physics and the same is true of Synergetic Engineering. In our case the breakthrough in Mathematics was made by Valdimir Velovoski in his Homotopy Type Theory which is a corollary to N-Category Theory. It turns out that Category Theory is intimately related to what is called Groupoids, and Directed Graphs, and that there is a corresponding N-Grouped theory that is associated with N-Category Theory. It is this joint relation between N-Category Theory and N-Groupoid Theory which is expressed in terms of Directed Graphs that is the proposed basis for the understanding of Schemas that come out of the connection of Schemas to higher dimensions in mathematics. Thus Schemas Theory has its own basis in a recently discovered theory of math that establishes the basis for Types which is not just a subject of programming language construction but is in fact a fundamental basis for understanding the synthetic qualities of Systems and other Schemas. Thus Schemas Theory based on N-Category, N-Groupoid and N-Directed Graph Theories is a fundament and deep sea change in how we understand systems that we need to seize and utilize to address the challenges that lie before us in Synergistic and more narrowly Systems Engineering informed by Systems Science and the Social Sciences as well. It turns out that Social Sciences have always drawn on Systems Sciences for guidance in studying their phenomena of interest related to people. So the Social Sciences has a less arduous learning curve in confronting the imperative of using Systems Science as an inspiration and guideline for studying phenomena and thus the Systems Engineers have a lot to learn about the application of Systems Science from Social Scientists.

Here we only mention this possibility to inspire others to create candidate paradigm sifts for our fledgling academic discipline that needs to shake off the constraints of the dictates of the customer in order to explore alternatives on its own for understanding its own subject matter beyond just following the lead of Software Engineering. Systems Engineering needs to discover what is its ownmost characteristics beyond Software and the other special engineering disciplines. And it is suggested here that what is its ownmost is Schemas Theory by which the information of Systems Science can be unified and digested, and by which Complexity Theory can be given a context because not just systems, but patterns, and forms and other schemas can be either complicated, complex or emergent. And what is amazing is that because Schemas Theory has not been developed previously by other disciplines this is something that it can bring to the table which will affect eventually the other disciplines themselves as they realize that they too are organized by the schemas which they use to study phenomena or design artifacts in engineering. Thus Schemas Theory is potentially not just a revolutionary approach to Systems Engineering but other disciplines as well especially Systems Science. The two meta-disciplines will draw support from having the meta^2-discipline of Schemas Theory defined. Speaking in terms of Category Theory we can say that the mappings of the schemas that occur in disciplines are basically morphisms represented by arrows. But at the meta-discipline level where we along with Len Troncale identify Isomorphies and Linkage Propositions are basically studying the functorial morphisms between categories defined by the disciplines. But Schemas Theory operates at the third meta-level of morphism with natural transformations which in fact differentiate functors. And that is why schemas theory discovers other schemas besides that of the system even though it is the next emergent level of generalization beyond the System Science tradition and the Systems Engineering practice community. Schemas Theory has the potential for transforming the Systems Engineering Body of Knowledge by bringing in what is known about the other schemas into that body of knowledge and expanding the ways we have of looking at various phenomena. We already use these other schemas in our daily and working life as engaged Systems Engineers and it is odd that we do not talk about them in the knowledge that we store about out our dealings with systems that contain monads, patterns, forms, ecosystems (meta-systems, openscapes), domains and worlds. There are four other schemas but they lie beyond the realm of experience as scaffolding. These six are the central schemas directly experienced in our daily life and in our working capacity which can be verified easily phenomenologically. Because they are the a priori synthetic templates by which we organize all phenomena in spacetime that we experience they are ubiquitous. And because we can phenomenologically verify them directly in our experience they provide a foundation for our practice that can easily be validated and serve as a framework for storing knowledge. And because they connect to mathematics directly they give us a royal road for bringing higher order mathematics to bear on Systems Engineering problems at the level of the system itself, as well as the other emergent levels of the schemas at their different scopes. We are constantly thinking conceptually higher dimensionally and it is in these higher dimensionalities of thought that the semantics of what we are doing together as teams are stored, in conversational

memory but also in representations that we create. All the representations we build as well as plans, pictures, models, simulations, testbeds, etc are lower dimensional constructs that we have to use to attempt to approximate what we know directly at higher dimensions of conceptual thought. It turns out that natural language in teams that share conversational memory is the most effective and efficient way to store information about the system and that is why the move toward agile and lean approaches are gaining momentum. But still we need models of the systems we build that are externalized, objectified and then re-internalized in the development process. And it would be best of those models were semantically rich rather than semantically poor as are UML and SysML due to only representing point to point relationships. And that is why I have done research on Domain Specific Domain Languages that are semantically rich because a single statement in human readable form represents a single fact about a design within a controlled vocabulary and a controlled simplified grammar. And it turns out after years of research looking for a way to make Design Domain Specific Languages viable as an alternative approach that it is Homotopic Type Theory that achieves that goal when applied to the Domain Specific Design Languages such as the Integral Systems Engineering Methodology (ISEM) so that we do not have to express ourselves as Systems Engineers using mathematical formula and category theory diagrams but can instead use controlled languages that are human readable but also semantically rich. Because Schemas structure language as well as experience of phenomena out in spacetime beyond our bodies by controlling the language vocabulary and structure we can gain the benefits of the basis in Category Theory, Homotopy Theory of Groupoids, and Directed Graph Theory in a way which is easy to understand both by ourselves and by others who are not Systems Engineering specialists. Extensible Domain Specific Design Languages give a universal way of expressing more restricted modelling alternatives such as UML or SysML and other Systems Engineering Tool internal languages in a way that like Category Theory allows us to connect these various representations and also allows us to render them human readable, and also in a form that can be stored like Code in text files rather than in other specialized tools that have other visual representations but no connection to the code base. We can perform a kind of literate programming by including design level domain specific language statements in the comments of the code, which is free of the delocalization and decoherence which occurs when we translate design theories into Turing Complete source code representations.. Thus there is an effective way available for applying the insights gained from Category Theory, Groupoid Homotopy Type Theory, and Directed Graphs that not only gives a basis for translation into and out of UML and SysML but also gives a more easily accessible way to extensibly add to the language of the design minimal methods as well as represent design level objects that can be used as configuration files for code generation, or at least design detail representations. Thus we suggest ISEM as a tool for communication between and about the other modelling languages that expresses the internal coherences within the systems, and other schemas that we are dealing with in a way that takes advantage of the new deeper mathematically informed vantage point offered by Category Theory, Groupoid Homotopy Type Theory and Directed Graph Theory, of which SysML and UML are weak sematic

cousins to ISEM. We need strong semantics, a mathematical basis, a way of representing concepts at their own level of complexity by an controlled approximation to natural language and ISEM is a candidate to perform that bridging and implementation of the mathematical underpinnings in a way that is humanly readable and not mathematically challenging in the way that most mathematical notation is daunting. We need many possible candidates for paradigm change and Schemas Theory gives us a straw man to judge other candidates by. We are looking for a Candidate stronger than Schemas Theory ultimately and that may appear two or three paradigm shifts down the road of the evolution of the discipline of Systems Engineering as a purportedly Academic discipline. But in the mean time Schemas Theory can stand as an exemplar that points to the horizons beyond Systems Engineering as it is taught and practiced today that can bring great benefit if understood by practitioners who need a full toolbox of ways of understanding phenomena if they are going to confront the challenges that face us in the near future. Seeing the world through a single lens of the System Schema is no longer acceptable once we know that the other schemas exist and are just as useful as the Systems Schema for understanding phenomena and building technological artifacts that are our products.