An Introduction to Complex-System Engineering

Michael L. Kuras
mikelkuras@comcast.net

Introduction

This is an introduction to complex-system engineering (cSE). cSE is going to become the second branch of system engineering. It is still in its formative stages. All of cSE's presently known ideas are briefly discussed. A familiarity with the first branch of system engineering, so called traditional system engineering (TSE), is assumed.

Motivation and Overview

TSE is not applicable to every problem. Its applicability is limited by the assumptions on which it rests. TSE's assumptions are essentially the assumptions of **reductionism**. Reductionism, in its essence, is the belief that *any* portion of reality – including all of it – can be understood or comprehended by understanding the parts of that reality and composing a mental model of the greater reality exclusively from those parts. All of the parts of some portion of reality, and the relationships among those parts, can directly account for *everything* that can be known or conceptualized about the greater reality. The parts of such a whole can also be resolved into parts and understood in the same manner. This isn't true all of the time.

To reject the universality of reductionism is to accept that there are properties that can only be associated with the whole of something – but not with its parts. Such properties are frequently termed **emergent**. In addition, when the relationships or interactions among the parts of a whole are considered, it is not always possible to find a way to consider them sequentially that is equivalent to their consideration in parallel. Even if system engineers are willing today to accept these non reductionist propositions, they do not know how to transfer that acceptance into the practice of their discipline.

Much of the debate about reductionism versus its alternative, **holism**, is cast in terms of reality itself. This is a serious mistake. It overlooks the role that the human brain might play. It is the human brain that composes the mental models that are our only window on reality. The human brain is finite. The consequences of this simple fact are almost always overlooked.

Accounting for the consequences of a finite human brain leads to multi scale analysis (MSA). Multi scale analysis is one of the key ideas in complex-system engineering. Validating the need for multi scale analysis is exceedingly difficult. This is because any conceptualization of reality (a mental model of that reality) can never be directly compared with reality itself. Validation must be indirect.

The second major idea in complex-system engineering is evolution. Evolution is also inconsistent with reductionism. Most system engineers are unfamiliar with evolution. Most have heard of it – but they are not familiar with it in any depth, and not with the processes that account for it. And certainly system engineers do not know how to connect evolution to the engineering of systems.

Evolution is frequently associated exclusively with biology. This is too restrictive. Evolution is much more widespread than that. Evolution is also context dependent. The processes that drive evolution vary as that context changes. However, whatever the processes might be in any specific context, they always express in some fashion what are termed the predicates of evolution. These will be identified, as well as what can be considered an explanation for evolution in any context. Both of these topics are important in complex-system engineering.

Given multi scale analysis and evolution, there is one more key idea in complex-system engineering. Because cSE is a branch of system engineering, it must have a set of methods. Engineering, including system engineering, is always about using a set of methods to establish an equivalence between a real world problem and a real world solution. The methods in cSE are termed the Regimen of complex-system engineering (Kuras, 2004). They are just listed here for now. Their context sensitive formulation and application depend upon multi scale analysis and evolution.

THE REGIMEN OF COMPLEX-SYSTEM ENGINEERING

1. Define Outcome Spaces.	5. Specify rewards and penalties for autonomous agents.
2. Selectively perturb self-directed development.	6. Judge outcomes and allocate prizes, accounting for side effects.
3. Characterize Continuously.	7. Formulate and apply stimulants for synergistic self-directed development.
4. Perturb the environment.	8. Formulate and apply safety regulations.

MULTI SCALE ANALYSIS

In order to appreciate the limits of reductionism and how to get beyond them requires an examination of how engineers (in fact, all humans) think – about how they compose problems and solutions in their heads, or how they build mental models of the real world. This should quickly become a question about how the human brain works. This may seem a bit daunting to the average engineer. But it has to be addressed. How does the human brain form and manipulate conceptualizations of reality? And does this influence what an engineer thinks of as real problems and real solutions? Are there built-in characteristics of the human brain that prevent engineers from having a completely "transparent" window on reality? In other words, does the brain alter or transform or filter in some way what an engineer assumes to be reality as he or she builds and manipulates mental models of that reality?

The answer is that the way that the human brain works does influence what is comprehended as reality. The window on reality that the brain provides is not a completely transparent one. The human comprehension of reality is not distortion free. What makes this very difficult to deal with is that it is not possible to comprehend anything without that influence being present. Nothing can be understood without conceptualizing it. And it is not possible to directly make "before and after" comparisons of the brain's influence on conceptualizations of reality. Nonetheless, the influence of the brain on what is conceptualized should not be ignored. Multi scale analysis explicitly acknowledges this influence.

Human conceptualization is limited. This is because **the human brain is finite**. Its capabilities and capacity are both bounded. It is this limitation on capability and capacity that produces a less than transparent window on reality. Surprisingly, this characteristic has largely been ignored in the prior work on this issue. Maybe it is too obvious. But for now, here are some of the consequences. All of reality cannot be conceptualized at once. (We can't even conceptualize all that we think that we know at once.) And there is no way to ensure that combining partial conceptualizations of reality (either in their entireties or in some abstracted form) will yield a comprehensive conceptualization of the greater totality. Conceptualizations of reality, mental models, are *always* partial – and limited to one "scale" at a time. The human brain can, however, change its scale of conceptualization altering what is included and omitted in mental models of reality.

These are statements about how the human brain works and nothing more. For example, all of the properties and behaviors of all of the particulates of a cloud, and all of the properties and behaviors of the cloud as a whole cannot be conceptualized at the same time. It is not possible to conceptualize water and all of its properties while conceptualizing all of the individual atoms of Hydrogen and Oxygen that comprise the water. It is not possible to simultaneously conceptualize, say, the full meanings of individual letters or words and the meanings of whole sentences and paragraphs. It is not possible to simultaneously conceptualize all of the individual bees of a beehive and the totality of that beehive. The human brain doesn't work that way. It generates separate mental models for these aspects of the same reality; and it can only toggle between these mental models. Conceptualizations or mental models are based on the brain's internally developed frames of reference. And the brain can and does employ different frames of reference as it forms mental models of different aspects of the same reality.

Frames of reference do not refer just to the notions of space and time. They are the basis for *every* pattern that is a portion of every mental model of reality that the human brain generates: for example of color, of taste, of ownership, of authority, of matter or substance, of quality and quantity, and so on. And sometimes the frames of reference used in mental models of the same reality are not just different but incompatible as well. A change in the scale of conceptualization occurs when the brain shifts between incompatible frames of reference.

The entire content of every conceptualization (of every mental model that the brain might form) is exactly and only a finite set of patterns; memories are basically stored versions of earlier conceptualizations; the capacity for storing patterns is also finite or limited; every current conceptualization depends on earlier conceptualizations (i.e., draws on memory); and frames of reference are the deepest and most condensed forms of memory. These points elaborate what it means to say that the human brain is finite, and that it is limited or bounded in its capability to form and retain conceptualizations of reality. If these points were further elaborated, the consequences just outlined would follow.

What is crucial here, however, is that mental models of reality, based on incompatible frames of reference, simply do not combine as suggested by reductionism. And any analysis, based solely on such premises, is going to be partial and even flawed.

Multi scale analysis allows for, but does not rely exclusively on, reductionist techniques. It uses, in addition, **statistical analysis** to find and confirm relationships that span scales of conceptualization – that is, between mental models of the same reality that are based on incompatible frames of reference. Even at a single scale of conceptualization, multi scale analysis departs from an exclusively reductionist approach. That departure is **regime analysis**, which is based, at least, on Radial Category Theory and Reed network models. Regimes are non-disjoint and non-hierarchical partitionings. Radial Category Theory came out of Linguistics, and Linguistics is a branch of Cognitive Science – which focuses on how we think and communicate. Reed networks are discussed briefly below.

In summary, multi scale analysis explicitly acknowledges that many of our mental models of reality cannot be combined or related to one another in a purely reductionist fashion. At present the only way to find or to explore cross scale relationships is with statistical analysis. Moreover, at any one scale of conceptualization, reductionist analysis can be augmented with regime analysis. The presently known techniques applicable to regimes are based on Radial Category Theory and Reed networks.

EVOLUTION

Evolution is the next major idea. Evolution is a label for distinguishable differences in successive generations of a population.

Evolution is a label for *changes* that are seen as gradual or progressive or cumulative but *not* repetitive (like the changing seasons or a pendulum), and that are not viewed as arbitrary or random. Further, evolutionary changes are not the direct consequence of an explicitly identifiable outside agent's intervention (such as a constantly applied force) – and so are frequently termed *self-directed*. Such changes are generally understood to apply to the substance and structure of things as well as to their behavior. Since this form of change is not seen as arbitrary, it may be due to some process or processes. *Theories of evolution* are attempts to identify and characterize these processes. Theories of biological evolution are the ones most frequently and explicitly examined. In fact, most people still associate evolution exclusively with biology.

Evolution deals with self-directed changes in populations. What this means for a system engineer is that a system must be treated as a set of populations. Evolution unfolds at multiple scales of conceptualization. And it is always context dependent. Said another way, the evolution of a system – that is, self-directed changes in a set of populations – can never be wholly disconnected from the environment of those populations. Both the populations and their environment must be considered and analyzed if the system is to be engineered. It simply is not possible to disjointly partition an evolutionary system from its environment.

Evolution is driven by a set of evolutionary predicates. Typically, five or fewer are identified. But this can be misleading. These predicates will always be expressed as context specific processes. These processes are continually generating outcomes (as modifications or changes in the populations of the system). The Regimen of cSE does not directly seek to produce such outcomes itself. Instead, the Regimen's methods operate on these processes.

These are the five commonly identified predicates of evolution.

1. **Member Addition**: The size of a population increases (through reproduction, through recruiting, etc.).
2. **Similarity** (Heredity): New members are similar to existing members. (In the specific case of reproductive increase, the characteristics of new members are directly derived from the characteristics of existing members.)
3. **Variation**: The characteristics of new members are not uniform.
4. **Adaptation**: Members of a population do not behave uniformly and behavior is not independent of other members and the environment.
5. **Selection**: Members of a population are subject to attrition as a consequence of their characteristics and behavior.

Generally, the rate of new member addition will be greater than the rate of member attrition. This is usually referred to, in biological contexts, as super fecundity. However, there can be temporary exceptions.

There are actually more predicates, although they are frequently overlooked. Not all populations evolve. And not all ***finite*** **populations** evolve. The necessary and sufficient condition for this is identified below. The five predicates listed above are still necessary, however; but they are not sufficient by themselves. For example, if all members behave uniformly the population will not evolve. At the same time, if any of the other four predicates is not expressed in some way, a population will not evolve *even if* behavior is not uniform. But there is more.

The persistence of a whole population must be much greater than that of any of its members. This is almost always overlooked as well. Roughly speaking, there are **no immortal members** in an evolving population. This is why it is foolish to take too literally the often repeated phrase that "survival of the fittest" is the essence of evolution. It's not. No member of an evolving population, no matter how fit it might be – even the fittest in a population – is going to survive indefinitely. If that were to happen, then the population itself would cease to evolve. But so called fitness is a useful measure in understanding evolution. It is a way to score the persistence of members and populations. And it can be quantified. In general, fitness is always less than 1. An informal definition of fitness is provided below.

Members must also exhibit **life cycles**. A life cycle is the period of existence from inception to elimination through attrition. New members must **mature** before they are full members of an evolving population. The period of maturation may be extremely brief – and even confined to the moments of inception – but it is a necessary facet of evolution. Only mature members can participate in (contribute to) the addition of new members. The processes of maturation are usually distinct from those of evolution, even given specific populations. Maturation recapitulates prior evolution; but recapitulation should not be interpreted as simply repeating prior evolution. The processes of maturation and of evolution do, however, share the same predicates.

There is a necessary and sufficient condition for a system of finite populations to evolve. It is the accumulation of complexity by the system. If this isn't happening, then evolution isn't happening. Roughly speaking, if one system is more evolved than another, then one has accumulated more complexity than the other. This makes an understanding of complexity important in complex-system engineering. Informal definitions of complexity and fitness are provided below – and then used to make an important distinction.

Price's equation is currently the best summary of evolution at a given scale of conceptualization. It is a useful statistical relationship. But it is still a very partial summary of evolution. Much work remains to be done in understanding evolution. That task is by no means complete.

$$\overline{w}\Delta\overline{z} = COV(w_i, z_i) + E(w_i\Delta z_i)$$

Price's equation is applicable to a single scale of conceptualization of an evolving system, but it is not expressed directly in terms of complexity. Instead, it is expressed in terms of individual characteristics or behaviors and their connection to fitness. In Price's equation, w represents fitness, and z a specific characteristic or behavior.

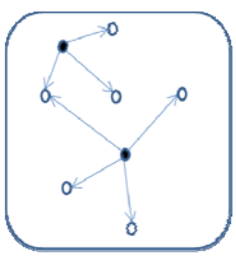
Regardless, the complexity of a system can accumulate across multiple scales of human conceptualization. To understand how that might unfold, and in the absence of a better version of Price's equation, network models can be used.

Finite populations can evolve. When they do, their capacity for accumulating complexity is not unbounded at any one scale of conceptualization (since they are, after all, finite). Nonetheless, the capacity of a system to accumulate complexity is seemingly unbounded. Network models can help to understand this phenomenon.

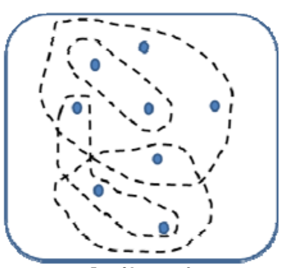
This can be done by associating members with **nodes** and the organizational and behavioral relationships among members with **edges** or links. The complexity of a population grows as membership increases and the relationships among members increase. There are three types of network models available to capture this. They differ primarily in terms of the characteristics of their edges. Basically edges can be one directional, two directional or without direction. Reed networks have edges that are non directional. Reed network models are not widely known or used. They are an important part of multi scale analysis.

As complexity accumulates in a system, there is a progression in the network types that should predominate in modeling the populations of a system: Sarnoff → Metcalf → Reed → Sarnoff → … This progression is tied to the incremental costs of adding additional nodes and edges. This cost is measured in the “energy” needed to accomplish the addition. The least costly alternatives available to the system to increase the amount of complexity that it accumulates will persist and eventually predominate.

What is of special note here is the **Reed → Sarnoff progression**. The continued addition of members and relationships in the populations of a complex-system requires an additional scale of conceptualization in order for the added complexity to be humanly comprehended. In short, for a human being to understand what is happening, new frames of reference must be employed in the pertinent mental models. The frames of reference associated with Reed networking in this progression are not appropriate for the added complexity captured by an additional Sarnoff network model at a new scale of comprehension. The Reed network and the additional Sarnoff network use incompatible frames of reference. They cannot be simultaneously comprehended by a human being today. At a minimum the members of the relevant populations will seem to be different.


Sarnoff Network


Metcalf Network


Reed Network

A Sarnoff network uses one-directional edges, represented here as arrows. These edges represent one way flows or relationships from source nodes to sink nodes. Each edge must connect exactly two nodes, but each node can be associated with multiple edges. The second example is that of a simple Metcalf network. It is seen by many as a generalization of the Sarnoff network. There are no distinctions of node types now; they are all the same. As before, each edge connects exactly two nodes. But in this case, flows or relationships can be in one or both directions (from the first node to the second, vice versa, or both). Also as before, a single node can be associated with multiple edges. The third example is that of a simple Reed network, sometimes called a combinatoric network. All of the nodes in a Reed network are notionally the same as with Metcalf networks. In this case, dotted lines represent the individual edges. Individual edges are non directional and *one* edge can interconnect *any* number of nodes, not just two. As before, though, one node can be associated with any number of edges. It is well beyond the scope of this introduction to discuss how to use the various sorts of network models that are available. But it is important to appreciate that they are all available – especially the Reed network model.

In the interests of brevity, only an informal definition of complexity and fitness can be provided here.

Complexity: the measure of a thing's (a system's) available changes *that do not alter its identity*.
Fitness: the measure of a thing's complexity relative to that of the "complexity" of its environment.

Maintaining "identity" is important in understanding both complexity and fitness. Identity is one of the **Modalities of Conceptualization**. These would be introduced in a discussion of a finite human brain and its implications for the formation of mental models or human conceptualization. These Modalities are the facets of how the human brain functions without positing physiological explanations for them – for example in terms of neurological networking or chemistry. That can, of course, be done. Roughly speaking, "identity" is the functionality of the brain that aggregates patterns giving them a cohesion relative to one another that is greater than with all other patterns in a conceptualization.

The complexity of something is meaningful only as long as that something remains cohesive in the human conceptualization of it. This is its identity in a nutshell. Another way of saying this is that it does not die, or disintegrate or collapse. Excessive change can do this, and when it does, the meaning of complexity is lost. Fitness is another measure. It is based on complexity; it scores how closely a system's complexity matches the potential changes in its environment without the system losing its identity.

PULLING IT TOGETHER (PART 1)

Using these definitions, a clear and powerful distinction between traditional system engineering and complex-system engineering can be made. Traditional system engineering is directed at decreasing the complexity and increasing the fitness of a system at a single scale of conceptualization. Complex-system engineering is directed at focusing and accelerating increases in the complexity of a system without a loss of fitness at multiple scales of conceptualization. Other distinctions are also possible. For example, TSE is applicable to problems that can be stabilized, while cSE should only be applied to problems that cannot be stabilized.

And it is now possible to provide an initial definition of a complex-system. A **complex-system** is a system that evolves (self-directed increases of complexity and fitness) at multiple scales of conceptualization. Both this definition and the distinctions between TSE and cSE deserve further discussion, but that is precluded here in the interests of brevity.

A different name is needed for traditional system engineering now. The term traditional suggests orthodox – and so implies that any other approach is unorthodox. And for many unorthodox is very close to wrong, or at least to suspect. The two branches of system engineering should be distinguished according to the kinds of systems to which they are applicable. This is already the case for cSE. TSE should be relabeled reducible-system engineering or rSE.

In order to engineer complex-systems it is best to treat them as populations. When doing so, it is important to be alert to the possibility that a whole population might have properties that cannot be associated with the members of the population. These are its so called emergent properties. However, other member properties (or relationships) might well be related to these emergent properties. This is something that only multi scale analysis will reveal, always with the understanding that statistical correlation is not causation. Also, a whole population at one scale is not always a member of another population at a "higher" scale. This is reductionist thinking. Although this might be true in some cases, it is wrong to assume that this is so in all cases. It's not.

A system exhibits functionality, organization, and substance at one or more scales of conceptualization. Distinct and frequently incompatible frames of reference are necessary for each scale. These frames of reference are not simply those of as space and time, as was noted earlier. Frames of reference are necessary for *every* pattern that our brains assemble as the content of conceptualizations: like color, flavor, ownership, and authority, and not just space and time.

Lastly, in treating a system as a set of populations at multiple scales, a distinction should be made between **autonomic** and **autonomous** members and populations. Primary attention should be focused on the autonomous ones. These are the "agents" that, if present, are most easily associated with the self directed development of the whole system. Individual human beings in a society are examples of autonomous agents. They are driving the evolution of society. A watch and a cell phone are examples of autonomic agents, as is an automobile. Watches and cell phones and automobiles are not driving the evolution of human society – although they may be evolving themselves, *as populations*.

Here are some examples of complex-systems. Brevity, however, precludes their discussion here.

- The Internet; all of the software in the world.
- English (natural languages, generally).
- Ballistic Missile Defense System (not individual missiles, radars, etc.).

- Purpose built fleets of aircraft (not an individual aircraft).
- A human city (not a building or a road).
- The Human Immune System; the Health Care System.
- The Internal Revenue Service (IRS).
- Network Centric Enterprise Infrastructure.
- Sensor swarms.
- A corporation; the national economy; the world economy.
- Religious congregations; a political faction.

Complex-systems evolve. This can also be understood as learning. Whole populations have to learn, even if their members do not. Of course, members can learn too. So complex-system engineering can also be understood as focusing and accelerating the processes of learning. And many of the methods associated with encouraging learning can be transferred to cSE. Terms like "evolutionary engineering" also suggest a focus similar to that of cSE.

REGIMEN OF cSE

The Regimen of cSE is the set of methods that can be used to engineer a complex-system. They were listed earlier; each of them is now discussed very briefly.

The first two methods can be understood as umbrella methods. The other six methods can be understood as more specific elaborations of one or both of these.

- Define **OUTCOME SPACES** at multiple scales of conceptualization and for multiple regimes. This means thinking globally but not always disjointly and in terms of large sets of acceptable possibilities, not in terms of a lattice or web or hierarchy of specifically required outcomes (properties or behaviors).
- Selectively **PERTURB SELF-DIRECTED DEVELOPMENT** (organizational and behavioral) at specific scales and in specific regimes. This means intervening locally, without exercising control, and expecting side effects.

Outcome Spaces in cSE may seem roughly similar to establishing requirements in rSE – except that Outcome Spaces are far more general and less persistent than are requirements specifications. And they need not be complete. They can even seem contradictory to some extent. Outcome Spaces broadly identify what are currently viewed as desirable properties, relationships, or interactions for the populations and population members of a complex-system. Perturbing self directed development in cSE may seem roughly similar to taking direct action to control development in rSE. Control is important in distinguishing the methods of cSE from those of rSE. **Control** [of a thing] is the realization of predictable and persistent consequences through actions [on a thing] that supersede or preempt any other actions [imposed on or self-initiated by a thing]. cSE does not seek to control the development of a system; rSE does.

- **CHARACTERIZE CONTINUOUSLY.** This means tracking the populations and population members of a complex-system, noting the changes in their properties, relationships, and interactions, as well as in appropriate measures such as complexity and fitness.

A complex-system operates and develops, and therefore changes, continuously. It never shuts down or is "turned off." That is because a complex-system is one that is evolving. Roughly speaking, to "turn off" a complex-system, would mean to kill it. Its operation and development are self-directed. It is important to keep track of this evolution – especially after efforts are directed at influencing that evolution. Continuous characterization is also where multi scale analysis fits most naturally into the Regimen. The characterizations produced are made available not just to the system engineer but to the autonomous agents of the complex-system.

The continuous characterization of a complex-system can be understood to correspond roughly, in rSE, to specifying in advance what a system is to do or to be (the requirements specifications of rSE) in that many of the same skills are needed. But the application of those skills is very different. For those familiar with architecting, continuous characterization is somewhat akin to developing "as is" architectures – except that the "as is" architectures must be continually updated. And attention to continuous characterization is far more important than that of any "to be" architecture in rSE – which might loosely be associated with a set of Outcome Spaces, except that Outcome Spaces are not persistent and should be updated as the system evolves.

- Temporarily **PERTURB ENVIRONMENTALS** in order to influence the self-directed development of a system.

An updated definition of a system is needed. That has been provided elsewhere (Kuras, 2006). For now, it is assumed that the distinction between a system and its environment is understood. One of the ways that the self-directed development of a system can be influenced is to perturb the system's environment. This perturbation can be localized and brief – or more sustained and pervasive. An example that is frequently used is that of watering a garden. This alters the environment of the garden (the garden is the complex-system in the example). But it leaves to the plants in the garden (some of the autonomous agents of the system) to interact with that temporarily modified environment as they see fit as part of their self-directed development.

- Establish specific **REWARDS AND PENALTIES** for autonomous agents in order to influence their self-directed decision making and development.

- Judge cumulative and collective results, not just specific outcomes, in **ALLOCATING PRIZES**; account for side effects.

These next two methods are most easily discussed together – but they are really separate. Rather than specifying what autonomous agents should be doing, the establishment of rewards and penalties alters the factors that autonomous agents might consider in making and carrying out their own decisions about what to do. NASA has begun to use this method in their effort to engineer a persistent return to the Moon and the human exploration of Mars. This contrasts with what they did earlier in the Apollo Project – an effort that eventually died. Rewards involve the establishment of prizes and the criteria to win them. Judging refers not just to assessing the fulfillment of these criteria, but to examining any side effects that accompany them, and then allocating the prizes accordingly. Of course, if autonomous agents do not recognize a persistent linkage between criteria, judging, and prizes, then the efficacy of these methods will be diluted or worse.

- Formulate and apply **STIMULANTS FOR SYNERGISTIC SELF-DIRECTED DEVELOPMENT**.

This next method involves techniques that will alter the number, or the frequency, or the intensity, or the persistence of interactions or relationships among the autonomous agents of a system – with little or no regard for the details of specific interactions or relationships. This method is very much like stirring the pot. The necessary ingredients are presumed to be in place. This method involves techniques that accelerate – or decelerate – the interactive processes of evolution.

- Formulate and enforce **SAFETY REGULATIONS**.

This last method explicitly recognizes that cSE focuses and accelerates what is otherwise a natural unfolding of development in a complex-system. This acceleration (or deceleration) can increase the likelihood or the severity of the natural risks to the system (collapse, disintegration, etc.). This method involves techniques that can detect and offset or neutralize such risks, both before and if they materialize. So called “leading indicators” are important in this regard. These are acknowledged statistical precursors of more serious situations in complex-systems. Another way to think of this method is policing its autonomous agents.

But all of the methods are still generalizations. For most engineers, until there are explicit examples, these methods will remain cryptic. Unfortunately, brevity precludes a discussion of any example of any of these methods in any sort of detail.

As an illustration, just one of the ways that an engineer can formulate and apply stimulants that promote synergistic self-directed development in a complex-system that is also a social system is to facilitate the appearance or spread of markets. There are various sorts of markets: for example brokered markets and unregulated markets. Each has its strengths and liabilities. Given a particular complex-system, a brokered market may seem to be the most advantageous or practical. So, how does one go about facilitating a brokered market in such a case? And given that such a market develops, what does one do then? Here are some of the issues. It is necessary to identify autonomous agents that might function as buyers and sellers – and of course as a broker. It might even be necessary to create or inject an agent that will assume the role of broker. It will be necessary to characterize the goods or services that might be exchanged. It will be necessary to characterize the value propositions that will be used to mediate exchanges in the market. And it will be necessary to identify the impediments to such exchanges and how a broker can neutralize such impediments at a profit to itself without undermining the propensity for exchanges between buyers and sellers. Brokered markets succeed because they are win-win-win propositions. Complexity is being accumulated. Needless to say, answering these and other questions in any detail is context sensitive. This is well beyond the scope of this introduction.

PULLING IT TOGETHER (PART 2)

When TSE is not applicable, cSE will be. Since the system in question will be one that is evolving, the very first thing to be done is to characterize the system as a set of populations at multiple scales of conceptualization, and then how the predicates of evolution are being expressed in those populations. This includes the five classical predicates; but it should also include attention to the less familiar ones such as characterizing the life cycles of the members of critical populations. Beyond this, engineering estimates for the complexity and fitness of the populations and their members should be developed and maintained as the populations evolve. And Price's equation should be used in relating fitness to the properties and relations of the members of populations.

Multi scale analysis is applied as well – but not just once. This is done repeatedly as the system continues to evolve. In doing this it is important to be clear about characterizing the relevant scales of conceptualization involved, including as needed the associated frames of reference. Second, reductionist techniques can be applied at each individual scale of conceptualization. This includes analysis based on Sarnoff and Metcalf network models. But this might not always be feasible even if it is theoretically possible. Applying reductionism depends on completeness and detailed thoroughness. The effort required can be computationally prohibitive. And even if it is practical to apply reductionist methods, the significance may not be great. It is the autonomous agents in a complex-system that, if present, drive its development – and reductionism seldom provides useful insights into their motivations. Third, it is also possible to analyze the single scale aspects of systems using regime analysis. Radial Category Theory and Reed networks will aid in this regard, as will straightforward statistical analysis. Fourth, in terms of cross scale analysis, the only available techniques today are those of statistical analysis. This will reveal correlations and dependencies – but generally not causality. All of this will be important to the extent that the analysis informs the application of the Regimen of cSE.

And, of course, the appropriate forms of the methods in the Regimen must be formulated and applied as the system evolves.

Finally, it is always important to keep in mind that engineering a complex-system precludes direct control. It is the system itself that is in control, not the system engineer. The system engineer will be accountable, but he or she will not be in control. If it is initially difficult to understand what this might mean, then the example of a teacher and his or her student should be considered. It is the student who must learn. The teacher can only teach. But both will be held accountable.

Conclusion

Now, if it is desired to know more about cSE, or if there is an interest in applying what is already known, then call me. I might be able to help. If this is not appropriate, there are alternatives. It is possible, for example, to figure out how to do cSE on your own. A reading list (bibliography) is included to use in getting started. And, of course, this introduction is available as well.

Bibliography

Anderson, P. W. (1972, August 4). More is Different. *Science, New Series vol. 177 no. 4047* .

Gladwell, M. (2000). *The Tipping Point.* Little, Brown, and Company.

Heylighen, F., Cilliers, P., & Gershenson, C. (April, 2006). *Complexity and Philosophy.* arXiv.org pre-print.

Kuras, M. L. (2004). *MP 04B000038: Complex-system Engineering.* Bedford, MA: The MITRE Corporation.

Kuras, M. L. (2006). *MTR 06B000060: A Multi Scale Definition of a System*. Bedford, MA: The MITRE Corp.

Lakoff, G. (1987). *Fire, Women, and Dangerous Things.* University of Chicago Press.

Malone, T. W. (2004). *The Future of Work.* Harvard Business School Press.

Marwell, G., & Oliver, P. (1993). *The Critical Mass in Collective Action.* Cambridge University Press.
Okasha, S. (2006). *Evolution and the Levels of Selection.* Oxford: Clarendon Press.
Surowiecki, J. (2004). *The Wisdom of Crowds.* Anchor Books.
Weinberg, G. M. (1975; 2001). *An Introduction to General Systems Thinking.* Dorset House Publishing.