Engineering a Complex System:
A Study of the AOC

Douglas O. Norman
The MITRE Corporation[1]
dnorman@mitre.org

1.1 Introduction

The engineering of the Air and Space Operations Center (AOC) resists the application of traditional systems engineering approaches. More like an enterprise (with multiple sets of stakeholders representing separate fiefdoms and principalities, differing funding sources, and different management structures for the component pieces) than an airframe or a tank; the AOC is actually a "complex system" in the manner described by complexity science.

1.1.1. What is an AOC?

Described by the Secretary of the Air Force, James Roche, as the USAF's most effective terror war platform[2], the AOC (known as an AN/USQ-163 Falconer) is considered the senior element of the Theater Air Control System (TACS). The AOC is used by the Commander, Air Force Forces (COMAFFOR) to provide to the Joint Forces Air Component Commander (JFACC) tools and services for planning and executing theater-wide aerospace forces.

When the COMAFFOR is also the JFACC, the AOC is also known as the Joint Air Operations Center (JAOC). In cases of Allied or Coalition (multinational) operations, the AOC is also a Combined Aerospace Operations Center (CAOC).

The AOC, manned by a dedicated cadre of trained professionals, enables the JFACC to exercise C2 of aerospace forces in support of the Joint Force Commander's (JFC) campaign plan. The JFACC will employ the AOC to maneuver and mass overwhelming aerospace power through centralized control and decentralized execution to produce desired operational and strategic effects in support of the JFC's campaign.

1.1.2. Overview

Suggested in this paper are a set of processes which augment - and do not replace – traditional systems engineering. As a complex system, the AOC evolves. Drawn from principles of complex adaptive systems, the augmentations to traditional systems engineering, known as Complex Systems Engineering, seeks to mimic the forces which drive evolution and guide the results. In this paper, principles of complexity science which apply will be presented, and the engineering approaches outlined.

[1] The author's affiliation with The MITRE Corporation is provided for identification purposes only, and is not intended to convey or imply MITRE's concurrence with, or support for, the positions, opinions or viewpoints expressed by the author.
[2] Stated at a breakfast with reporters, March 17, 2004, in Washington, DC.

Technology has thrown us a curve. We can imagine ways in which it might be employed to help realize our visions, and it seems that the technical solutions are right in front of us, yet as we attempt to apply good engineering discipline to achieve the imagined result, the possibility of success seems to fade into the distance. This isn't unique to the military. Large, intricate, complex systems all seem to share this characteristic and cause similar frustrations among those charged with their engineering. The Air and Space Operations Center is one such example.

A few years ago General J. Jumper (then Commander, Air Combat Command; now Chief of Staff, Air Force) realized that the AOC (heretofore a virtual pickup game of people, systems and procedures) needed to have some discipline imposed to realize its true potential. To this end, he declared AOCs to be "weapon systems" which were to be treated as such. As an expedient, the AOC built and stood-up at AUAB was declared to be the first instance of the AOC Weapon System (AOC-WS), and its configuration was set as the baseline. This baseline was put under a configuration control process seemingly appropriate for a weapon system. To be charitable, this merely imposed a speed-bump on change. New systems and alterations to the "baseline" have occurred almost unabated; all for good and valid reasons from the point of view of the warfighters. Their needs seemed not to be met by the then-current systems and technologies present there. This begs the question: why is this the case? These are responsible, competent people clearly intending to do the right thing, and behaving in a corporately-responsible manner.

The timing of the declaration of an AOC-WS also corresponds to the desire of the Air Force leadership to reinvigorate Systems Engineering in the Air Force. An examination of how one applies Systems Engineering to the AOC-WS starts to show the seams in our approach to traditional systems engineering.

We have attempted to apply systems engineering techniques which have served us well in the past, and which were learned through hard lessons of failure over the past half century. Unfortunately, they seem not to scale to the enterprise. And the AOC seems to behave as an enterprise in the small.

1.2. Can Traditional Systems Engineering be applied to the AOC?

The AOC today is assembled from over 80 elements. There are infrastructure elements, communication elements, applications, servers, and databases. The goal is to compose the desired capabilities from the elements found in, or which can be brought into, the AOC. For the most part, today's systems are not composable. The systems:

- Don't share a common conceptual basis.
- Aren't built for the same purpose, or for use within specific (AOC) work flows, or for use exclusively at AOCs,
- Share an acquisition environment which pushes them to be "stand alone"3,
- Have no common control or management,
- Don't share common funding which can be directed to "problems" as required,
- Have many "customers," of which the AOC is only one,
- Evolve at different rates (as do individual system components) subject to different (generally uncoordinated) pressures and needs.

Because of the above, **integrating the AOC is an unbounded, unpredictable engineering activity**. Each of the various systems which are brought together and form the elements of an AOC are brought whole, built to satisfy some other need for some other set of stakeholders.

Reflect on Systems Engineering. Among the characteristics one requires to have a successful (or at least a low risk) outcome, there are a few which are absolutely required to ensure success using traditional Systems Engineering. These serve as boundary conditions for applying TSE:

- The specific desired outcome must be known *a priori*, and it must be clear and unambiguous (implied in this is that the edges of the system, and thus responsibility, are clear and known);

[3] The DoD's acquisition system is built around the concept of a "system" which seeks to separate a given system from every other. This separation extends from the concept through delivery and sustainment. Funds executed on behalf of the system acquisition are, by law, separate from all other monies with Congress carefully monitoring expenditures.

- There must be a single, common manager who is able to make decisions about allocating available resources to ensure completion;
- Change is introduced and managed centrally;
- There must be "fungible" resources (that is money, people, time, etc.) which can be applied and reallocated as needed.

Failing to have any of the above raises risk dramatically; and it is unlikely that mitigation strategies can be found which will reduce the risks introduced. How many of these boundary conditions are respected in an enterprise? Our sense is that the number is zero. There seems to be a correlation between small projects which build stand-alone, fairly simple applications/products which are under the complete control and management of a single party, and the likelihood of having these boundary conditions satisfied. Unfortunately, when one considers an *enterprise* every one of the characteristics mentioned above is violated. This isn't too surprising as Systems Engineering has evolved (very successfully) from an industrial, element-manufacturing point of view.
It is clear that processes must be applied where they fit. If boundary conditions for applying a process, or a set of processes, are violated the processes are not really applicable. Is an AOC such a situation?
We can test whether the characteristics we previously argued were required for successful outcomes using TSE fit an AOC:

- *The specific desired outcome must be known* a priori, *and it must be clear and unambiguous (implied in this is that the edges of the system, and thus responsibility, are clear and known);*
    - Test Result: Failed.

  There are expectations expressed in documents known as Block Requirements Documents (BRD) which lay out an AOC's planned functionality over time. The fly-in-the-ointment is that the plan implies a convergent set of developments which would deliver the capabilities found in the AOC BRD. This isn't the case; and it leads to the next characteristic to test.

- *There must be a single, common manager who is able to make decisions about allocating available resources to ensure completion;*
    - Test Result: Failed.

  As observed above, there are many component systems which are managed by many different organizations responding to many constituencies on behalf of a set of users, of which one user community is found at AOCs.

- *Change is introduced and managed centrally*;
    - Test Result: Failed.

  It is certainly the case that senior AF management has (and is) attempting to apply centralized management to bring AOCs under control. AOCs have been declared to be "Weapon Systems." To bring order and predictability to AOCs, the senior acquisition authority has asserted personal control over the official configuration; and detailed configuration control processes and measures have been imposed.

  To date, these measures have not worked; and we suspect they still won't. The only proximal result is a sense of stasis hovering over the AOC formal definitions. This invites the formation of black markets. Each Combatant Commander has funds that can be spent on their AOCs (the OIF AOC was built on Commander's Initiative funds). Bottom line: they have the means; and when a need surfaces, they can fix their own problems. And they are independent of the corporate AF staff.

  Besides, stasis of the AOC definition imposes no stasis on the component systems used to build the AOC, so what does a firm baseline mean in this case?

- *There must be "fungible" resources (that is money, people, time, etc.) which can be applied and reallocated as needed*.
    - Test Result: Failed.

  As mentioned, few of the total set of resources required to produce an AOC are controlled by, or in a way, that renders them fungible.

The conclusion is pretty straight forward. TSE doesn't lend itself to engineering or managing the engineering of AOCs (and by extension, enterprises). Can one take organizational or management steps to bring the characteristics which are outside of the boundary conditions back in line? Perhaps one; but they are all violated. A reasonable guess is that it is not likely that there is management or organizational changes which would allow TSE to be applied successfully to the aggregation of systems known as an "AOC."

1.3. What is a Complex System?

Complex Systems are alive and constantly changing. They respond and interact with their environments – each causing impact on (and inspiring change in) the other. Holland [95] has attempted to discover and describe the fundamental principles of any complex system. He has identified seven: aggregation, tagging, nonlinearity, flows, diversity, internal models, and building blocks. These principles can be identified in ecosystems, the internet, the immune system, the conduct of warfare, economics, the emergence and persistence of cities, nervous systems, and countless others. Bar-Yam [97] argues that human civilization is a complex system. A major discriminator between complex systems and those which are not is the amount of interaction among the composing elements. If these are indeed principles of all complex systems, then they should be present in an AOC; and if we look, we find them.

From the point of view of developing an engineering approach, a complex system has the following characteristics [Bar-Yam 97; Heylighen 95; Holland 95; Kauffman 93] which must be considered:

- Its structure and behavior is not deducible, nor may it be inferred, from the structure and behavior of its component parts;
- Its elements change in response to imposed "pressures" from neighboring elements (note the reciprocal and transitive implications of this);
- It has a large number of useful potential arrangements of its elements;
- It continually increases its own complexity given a steady influx of energy (raw resources);
- It has independent change agents present.

These characteristics violate the boundary conditions required for applying traditional systems engineering – either for initial design or to support subsequent change.

For systems developed using TSE, the primary mechanism for change is a deliberate "requirement" which defines a specific outcome. If one wishes to achieve a specific outcome with a complex system (actually, for any system) one has to reduce the dimensionality of the problem until a clear solution forms. TSE provides processes for just this purpose. Those approaches variously described as Top-Down or Structured Design are based on a linear, stepwise partitioning of the problem until one can directly implement the design.

The primary mechanism of change within a complex system is evolution. An engineering process for a complex system must guide the evolution of the complex system into the general outcome regions desired. This approach uses the high-dimensionality to help resolve local "pressures" with local solutions, rather than the high-dimensionality serving to paralyze a design process, or drive the engineer to make simplifying assumptions which then may drive an engineered solution into what will be found to be an unacceptable solution once in the field.

Complex systems are bottoms-up affairs, not top-down designs. Change ripples through complex systems causing local "pressures" among juxtaposed systems against which the systems respond to the pressures by undergoing change themselves and/or passing on the "pressure front." This is known as co-evolution, and in this way complex systems evolve. This is very much like what is seen and understood within ecosystems. In fact the science of natural ecology is the study of energy flows and evolution (actually co-evolution of juxtaposed organisms) and the description and examination of niches. Another obvious example is macroeconomics which is about the flow of money through "markets" and the creation, maturation, and extinction of players in markets.

The differences in the thinking required for traditionally-engineered systems and complex systems are significant. As an example, consider the principle of diversity. From a traditional stance, diversity is something which is eliminated: it immediately implies redundancy, non-optimality, and wastage. However, from a complex systems point of view, with recognition of the high-dimensionality of the problem space, variety leads to choice which may better fit specific problem spaces. The desire, or need, for efficiencies will likely drive a coalescence among the seemingly redundant pieces; but it may also illuminate where seemingly redundant pieces actually partition the space; that it was an overloading of terms which led to the seeming redundancy. Again, these are discovered characteristics, not engineered, at the complex systems level.

It's useful and important to note that flows in a complex system are emergent, not engineered. Organisms don't exist to satisfy a pre-determined energy flow, nor do companies and consumers exist to satisfy a pre-determined money flow. The flows in both cases are emergent, not designed. The interaction among the elements which compose to form the complex system form patterns of interaction which are the flows. Generally, the manner of composition and interaction is through events, not design points in a flow.

To convince yourself of this consider how you live your life. How about today? While you may have had a general plan on how this day should go, I'd bet that things are not turning out how you planned. The environment signaled you through events of some sort, and you responded to the events. You may well guide the day in general directions (i.e. into outcome spaces), but you probably are not strictly following a detailed plan of action and activities (i.e. a specific outcome). In fact, to do so would require one to isolate oneself from the environment. Sometimes we do this; but it is a particular decision which requires us to isolate ourselves to follow a specific, detailed, plan. And, to isolate oneself means separating oneself from the intrusion of events.

To summarize, complex systems have flows of some sort (or many flows) which are an emergent property of the interaction among elements of the complex system; and generally, the manner of composition and interaction is mediated through events.

1.4. Is the AOC a Complex System?

We can test whether AOCs fit the definition of Complex Systems by comparing the two:

A complex system is a system:

- Whose structure and behavior is not deducible, nor may it be inferred, from the structure and behavior of its component parts;
    - Result: Marginal Pass

  The AOC's desired aggregate behavior is reasonably well known; even it's desired changes, so this characteristic doesn't necessarily fit. However, if we take a broader view of an AOC as an element of C2 (i.e. the enterprise), then this statement becomes more correct.
- Whose elements can change in response to imposed "pressures" from neighboring elements (note the reciprocal and transitive implications of this);
    - Result: Pass

  This is certainly the case in the AOC. Independently-introduced applications (through independent agents) such as (for example) ADOCS and Falcon-View cause direct "pressure" on those applications which perform similar roles, or which could potentially act in concert with these introduced applications. As an example of resolving the introduced pressures, TBMCS specifically added certain Information Services to interact with ADOCS without the need to own and control it.
- Which has a large number of useful potential arrangements of its elements;
    - Result: Pass

  Since the AOC's workflows are numerous, and are in flux due to new missions and doctrine, this fits. It is also true, though, that it is the people who supply the flexibility; and they often fight the automation present.

- That continually increases its own complexity given a steady influx of energy (raw resources);
  - Result: Pass

  This also seems to be the case. For example, TBMCS re-architected from a monolith to a set of applications riding on a set of Information Services precisely to increase the number of possible connections and relations, and to allow more independence of creation and use of new clients of the services offered.
- Characterized by the presence of independent change agents.
  - Result: Pass

  The AOC has upwards of 30 independent agents – in the form of separate Program Elements (PEs). PEs are, by their definition and nature, independent agents.

It seems reasonable to conclude that AOCs are Complex Systems; and, since there is a need to apply a Systems Engineering approach to the AOC which is beyond the traditional Systems Engineering approach (see the earlier discussion), the AOC will benefit from a Systems Engineering approach which acknowledges the differences between the AOC and other more-traditional developments to which TSE can be applied.

1.5. How might one engineer a Complex System?

Having made the argument that AOCs are complex systems, how might we apply engineering discipline to them? Reworded: how is a complex systems engineered? Since complex systems are changed through evolution, how might we deliberately mimic the process of evolution; and in that way cause local engineering solutions to converge on the desired results? Complex Systems Engineering changes the focus from *"…here is the solution designed from the requirements, now go implement it…"* to *"…here are the selective pressures acting on the elements present (likely built using TSE), now resolve or reduce them…"*

To paraphrase a colleague, Mike Kuras, *complex systems engineering* is not a new or renewed attention to detail; it is an attention to overall coherence.

Complex Systems Engineering acknowledges the presence and action of "autonomous agents" as important elements of a SoS. These autonomous agents are precisely the effectors which must be (and are) eliminated to apply TSE. Again, to apply TSE one needs to eliminate the independent agents, or one needs to augment the set of tools for dealing with their continued presence.

To engineer a system in the presence of active, independent agents, we apply "selective pressure" to the aggregate of interacting elements allowing them to manage their own response and their own changes. This approach consists of:

- an environment within which the pressure can be brought to bear,
- the goals for the broad aggregate (described as an *outcome space*, not an *outcome*) – either behavioral ("it" has "this" capability), or structural ("it" has "this" characteristic),
- the rules of interaction among the independent agents,
- the broad reward (or punishment) structure for achieving pre-described goals (based on achievement, not promises),
- continuous characterization of the environment (to provide visibility to the participants),
- judgments of how, or whether, the outcome space has been reached.

As with natural ecosystems, the interactors must be in touch with one another chronically and continually. Right now, we have elements of the "environment" in place, but we have reasonable hope that establishing such an environment in the large is both practical and doable. Holland [95] points to its natural occurrence when the basic elements are present[4].

[4] Holland [95] talks of the basic elements of: aggregation, tagging, nonlinearity, flows, diversity, internal models, and building blocks.

1.6. Summary

The challenge is moving from “things” to “integrated collections of things” which are governed and managed independently. Although we presented the problem as an issue for AOCs, it is not confined to the AF, or to joint forces, or to DoD, or the US Government, or to the US. The observation that one must use methods which respect the characteristics of the enterprise; and which don’t require complete control, or complete knowledge in one place.

Our summary would be incomplete if we didn’t consider the insights offered by other professions who have faced similar challenges. Architecture has. Christopher Alexander (of architectural fame) talks about the illusion of control; and he observes that attempts to assert control generally has the opposite affect from what is desired; things tend to get worse, more out of control. He notes that the tendency “…to gain ‘total design’ control of the environment…makes things still worse…”[Alexander 79, pg 238] He points to the need to construct towns and cities using “patterns” which preserved the correct “nature,” and became “alive” in their own right. He calls this the “quality without a name.” He’s talking about complexity and adaptability.

Traditional Systems Engineering has always attempted to understand and deal with complexity; but the nature of that which was being engineered tended to be stand-alone with well-defined edges. Simple rules could tell what was “in the system” or “out of the system;” and the engineering activities started with, or required that, the requirements were well known, understood, and stable. As systems engineering came to deal with collections and aggregations of elements which were to be integrated into definite, well-understood (and understandable) forms-and-function which were to be stable over time. As we scale this approach up to the enterprise and find ourselves dealing with complex systems, we fall directly into the trap outlined by Alexander: things become worse.

Our traditional systems engineering has been concerned with finding those well-bounded subordinate elements, then (in essence) isolating them so they may be “engineered.” From this point one proceeds as if the element is isolated and unmoved by other juxtaposed elements. It’s this desire to “divide and conquer” which characterizes our tradition approaches. Is this wrong? No! But it’s not always correct; nor is it complete.

Consider the richness possible due to the potential interconnectivity now available, and the interdependence among elements implied. What is formed is an ecosystem where each element responds to its context through some accommodation - potentially evolving to respond (those elements which are “alive” respond and change). Consider further that each element’s context is set by the elements which juxtapose it in almost countless ways (forming a hyperspace of pressure). This is certainly an intricate, hard to understand-and-appreciate situation; and in that way, it may be thought of (in the usual vernacular) as complex. Using our traditional divide-and-conquer systems engineering (TSE) we would likely measure the external world, then make an assumption of constancy with respect to this external surround. Engineering would proceed on the element from this point of view.

But, the realization that the element under study also forms part of the context for every element which juxtaposes it starts to hint at the limit of the simplifying assumption made to perform the TSE: It imposes a pressure (an influence) on its surround in addition to feeling the pressure of the surround. Note the implications of the transitive nature of these influences. This is what is referred to as *complexity* as opposed to *intricacy* or *difficult to understand*. One could imagine waves and ripples of change flowing through this system of system. Likely, patterns will emerge when viewed from a higher level of abstraction. This is likely where we find ourselves with respect to C2 in a joint and coalition world, and where independent agents can introduce change according to their own agenda and timing.

The big questions: can such an aggregation be engineered at all? Can it even be understood? Will useful patterns present themselves? Should we even bother? We can say that we’ve failed many times in the past because we’ve made the simplifying assumptions I mentioned earlier.

One of the principles that pops out right away from taking this complexity point of view is the need to have tight control over all (or as many) characteristics of a system to be engineered as possible if one wants to apply TSE. Especially the ability to direct resources to problem areas as they arise. Is this insight new? No; but maybe, where it is impossible to meet that control and authority boundary conditions, there might be other approaches which can be brought to bear. And we believe the codifying of CSE is a step in the right direction.

Understanding complexity, and engineering complex systems is the next step. Systems Engineering is taking this next step. It is maturing past the point where a one-size-fits-all process is what's thought of as "correct." It is finding a new language with which to understand that which it attempts to engineer. This is maturity, and this is its future.

References

Alexander, Christopher; 1979; The Timeless Way of Building; Oxford University Press

Bar-Yam, Y.; 1997; Dynamics of Complex Systems; Perseus;.

Heylighen F., Joslyn C. & Turchin V. (eds.); 1995; The Quantum of Evolution. Toward a theory of metasystem transitions, ; Gordon and Breach Science Publishers, New York

Holland, J.H.; 1995; Hidden Order: How Adaptation Builds Complexity.; Reading: Addison-Wesley.

Kauffman S.A; 1993; The Origins of Order: Self-Organization and Selection in Evolution, Oxford University Press, New York,