A Universal Framework for Analyzing Complexity in Irregular Warfare

Introduction

Irregular Warfare (IW) and complex humanitarian emergency include operations of Disaster Relief (DR), Humanitarian Assistance (HA), Peace Operations (PO), Stability and Support Operations (SASO), Stability, Support, Transition and Reconstruction Operations (SSTRO), the Global War on Terror (GWOT), and others.

These situations are described by a set of state variables, organized within Political, Military, Economic, Information, and Infrastructure (PMESII) categories. The actions that can be taken to remedy these situations are organized within the Diplomatic, Information, Military, and Economic (DIME) categories.

The theoretical bases for the models come from all of the social sciences. These theoretical bases are referred to as Human Social Culture Behavior (HSCB) models.

Irregular Warfare (IW) involves many aspects of complexity at many scales. The scope of interconnectivity across disciplines of the social and physical sciences makes the modeling of IW particularly challenging.

A unifying Architectural Framework and common taxonomy is needed to unite this multi-discipline nature of IW complexity. Complexity theory[3] gives us some starting guidance.

- Do simple thinks first
- Make them work
- Build the next layer of simple things
- Don't change the simple things in the lower layers

"That it is possible to invent a unifying concept of structure within which all the various concepts of structure now current in different fields of art and science, can be seen from a single point of view. This conjecture is not new. In one form or another people have been wondering about it, as long as they have been wondering about structure itself; but in our world, confused and fragmented by specialization, the conjecture takes on special significance." If our grasp of the world is to remain coherent, we need a bead game; and it is therefore vital for us to ask ourselves whether or not a bead game can be invented."[6]

This paper investigates a framework that begins with some basic constructs of complexity and then builds out a framework by combining existing approaches to modeling systems of systems. These modeling approaches are loosely interconnected in a manner that enables the model to be distributed and redundant. The interconnections include hierarchical links as well and peer to peer networking. The framework includes a Static View of the hierarchies of PMESII organizations. This view enables the analysis of PMESII elements and variables independent of dynamic interconnectivity. This analysis focuses on determining the stability of the organization by examining it in layers. A framework includes a Dynamic View representing the lifecycle of the PMESII organization over lifecycles of operation. This view enables the analysis of the dynamic interconnectivity of PMESII elements and variables and the changing environment in which PMESII organizations operate. In each view, the concept of multiple perspectives is included to enable a multi-disciplinary analysis of the organization.

These two views help us to understand and appreciate a level of competition that exists across all complex organizations. In particular the competition between large scale functions and complex scale functions of the organization are represented in the Framework. Large scale functions provide for dynamics and information flow that occurs in the dynamic view. The complex scale functions provide for the complexity of each PMESII element/component. The more complex the PMESII elements, the more hierarchical they are, and the less dynamic. At this extreme the PMESII organization enables a high level of control, but lacks the ability to rapidly adapt and evolve, it is rather static. The more interconnected (networked), the more dynamic, adaptable, evolvable, resilient the organization becomes. At this extreme it is very difficult to control the PMESII organization. Aspects of swarm theory emerge. This makes it very difficult to anticipate what direction the organization will take next.

By analyzing these two views from multiple perspectives the framework enables us to better understand the interactions that lead to patterns of behavioral emergence as well as understand how significant (black swan) events can trigger permanent evolution and change within the organization.

With these concepts in place, mechanisms for representing the Static and Dynamic Views were selected. The DoD Architectural Framework[7] (DOFAF) was selected for representing the Static View. It is important to note that the DODAF is not the only architecture framework that could used. It is likely that different frameworks can and should be applied based on the specific organization being modeled. One size does not fit all. Use the right tool for the right job. The Missions and Means Framework [8] was selected to represent the Dynamic View for the PMESII life cycle.

Static View – DODAF

The DoD Architectural Framework[7] (DOFAF) was selected for representing the Static View.

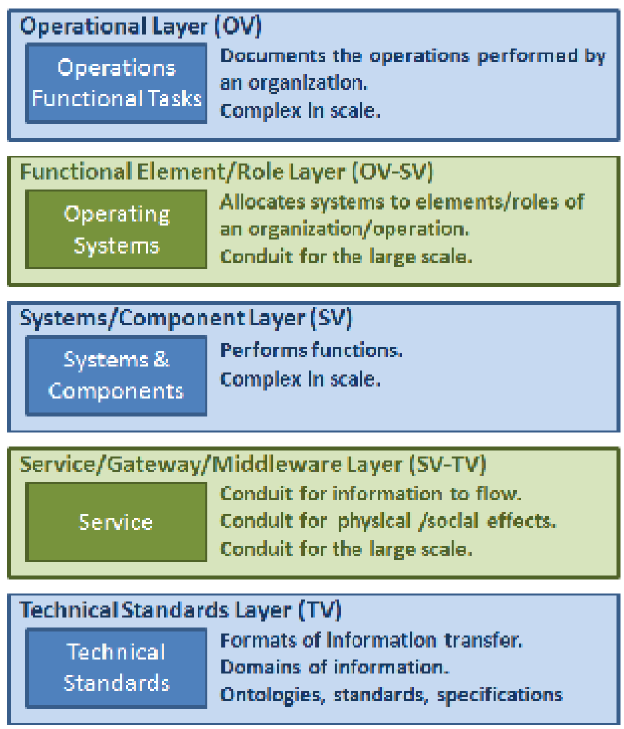

Figure 1: DODAF Layers

The Static View layers an organization from the bottom up. The hierarchy of the layers reflects dependencies of the higher layers on the lower layers. The DODAF definitions of these layers are tailored to the Framework as follows:

- Operational Views (OV) – Describe the operations performed by the organization as they relate to changing the current situation.

- Operational to System Views (OV-SV) – Identifies the organizational elements that at as roles in the operations (functional tasks). Roles encapsulate the Systems and Components that apply to each organizational element. This layer provides for the interconnectivity between roles and systems used by each operational element.
- System Views (SV) – Systems and Components provide the functionality of operational roles.
- System to Technological Standards Views (SV-TV) – Provides for the routing of information through technical standards. Middleware applications wrap the technical standards; gateways interchange data between technical standards. This layer provides the interconnectivity between systems and technical standards.
- Technical Standards Views (TV) – The technical standards are used to describe the information that ultimately informs the operations.

The OV-SV and SV-TV layers provide for large scale integration between layers. They provide for the interconnectivity that exists within the organization. These are the layers where external attempts to control the interconnectivity can lead to negative impacts such as interdependence of the influenced organization on the intervening organization. For example if during disaster relief one organization takes control over routes of supplies of another, the economic supply chain is replaced. The economic chain of supply and demand becomes interdependent on the organization providing relief. The original supply chain no longer exists. The new supply chain cannot be removed without leaving the entire economic chain of supply and demand in a dysfunctional state. Decisions concerning interconnectivity between layers must be well informed. These decisions affect the degree of large scale integration between layers. It is the specific form of interconnectivity that causes behaviors of a PMESII organization to emerge. Consider dictatorships. A single individual controls the interconnectivity between layers, resulting in a strong hierarchy that is highly susceptible to influence, less able to adapt or reform. While the individual (dictator) can adapt or reform, the dictatorship is rigid and rather fixed in structure. Functions are centralized enabling control. An organization can react very quickly, but be slow to change. The patterns of interconnectivity are analogous to genetic blueprint of the organization. Patterns of interconnectivity determine the organization's capabilities, flexibility, and limits of environmental conditions the organization can exist within.

What is not obvious is that these layers of interconnectivity provide for the interconnections within the higher layers. Consider C4ISR, most of these systems are stove pipes that can hamper communications between operating system elements of the military. The reason for this is the need for a clear command and control (C2) hierarchy in the military organization. Control over information flow provides the C2 element with influence and authority over the other operating systems. It should be noted that in the military world a bad IT decision can kill people. In the civilian sector these bad decisions mostly impact profit margins for a while. There are reasons for the military to be little more conservative; however, the amount of opposition to doing the right thing is often mind boggling. There is a need for the military's C2 to evolve into a more distributed peer to peer network. Too much conservatism means the gap will continue to widen. Too little conservatism means distribution could over-run the need and lead to a collapse in effective C2. In the following section the dynamic view of the framework explores mechanisms that can be used to keep evolution in pace with the changing needs of the military environment.

Dynamic View – MMF

The Missions and Means Framework [8] was selected to represent the Dynamic View for the PMESII life cycle.

The blue lines represent cycles of planning and synthesis of new components, systems, and other implementations. The red lines represent execution that causes the situation to develop or evolve.

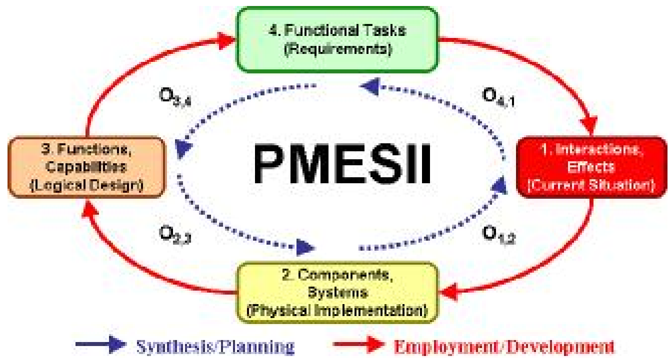

Figure 2: Initial MMF Lifecycles

- Level-4: Functional Tasks are the task-based, functional or goal oriented specification of the Operations that provide the Means to accomplish the Mission. These specifications identify conditions (Environment), measures (of Mission accomplishment), and standards (measure thresholds) of performance that are satisfied under fully functional system performance.
- Level 3: Functions, Capabilities are the Function-based, performance-centric "how well" specification of the Capabilities that enable systems and organizations to conduct Operations. Capabilities describe the external value provided (benefit); Functions specify the internal necessities required to deliver Capabilities (features).
- Level-2: Components, Systems are the Component-based, state-centric implementations of the Systems that provide the Means to accomplish a functional operation.
- Level-1: Interactions, Effects are the Interaction-based, phenomena-centric specification of the Effects that Operations have on Systems (systems of systems, organizations, etc). Interactions specify how execution changes the state of Systems, Domains or standards of (physics, chemistry, biology, psychology, sociology) that generate the interactions and effects.

The operators shown in Figure 2 are used to represent life cycle transforms from one level to another level across cycles of planning and employment.

For example, in the Planning cycle for software development the operators become:

- O4,1S = Needs of the current state of capabilities are transformed in to functional tasks/requirements for software development.
- O3,4S = Functional requirements are transformed into a logical design defining the functions and capabilities to be implemented.
- O2,3S = Logical design is transformed (developed) into code and unit tests of the physical implementation or functional solution.
- O1,2S = Deployment of the functional solution into an organization or system-of-systems which then affect the overall organization through interacts with the other components of the organization or system-of-systems.

In the Employment cycle for software development the operators become:

- O1,2E = Employment of the functional solutions affect in adverse (dysfunctional) or positive ways the overall organization or system-of-systems.

- O2,3E = The positive or negative impacts to an organization are transformed into an increased or decreased functional ability to perform tasks.
- O3,4E = The diminished or strengthened functions and capabilities affect the ability of an organization to perform necessary functions.
- O4,1E = The increased or decreased function of an organization or system-of-system transforms into an increased or decreased level of performance respectively.

One important aspect that was missed in the development of the MMF was evolution. The employment operators reflect a domino effect where physical effects, affect systems, which affect capabilities, which affect the functions that can be performed. But more is going on here. When the current situation changes irreversibly, older systems, capabilities, and functions no longer fit the new environment. Evolution occurs through necessity. However for change to be effective, the cycles of planning and execution must be in synch with each other. Imagine if planning outpaces execution meaning planning is being informed by outdated information on the current situation. In the military, efforts to analyze and wargame options would be out of synch with reality. That could result in the selection of targets that no longer pose a threat or worse yet, now house civilians or friendly forces.

Combined View – The Framework

The Static View and Dynamic View were then combined into an over arching Framework View. There are many details concerning how these views integrate. First let us example the parallels in both frameworks.

If we layer each top down, we discover an important insight. The life cycle levels/stages align with the architectural layers. We can now understand what life cycle phases influence an organization's architectural layers in planning and in execution.

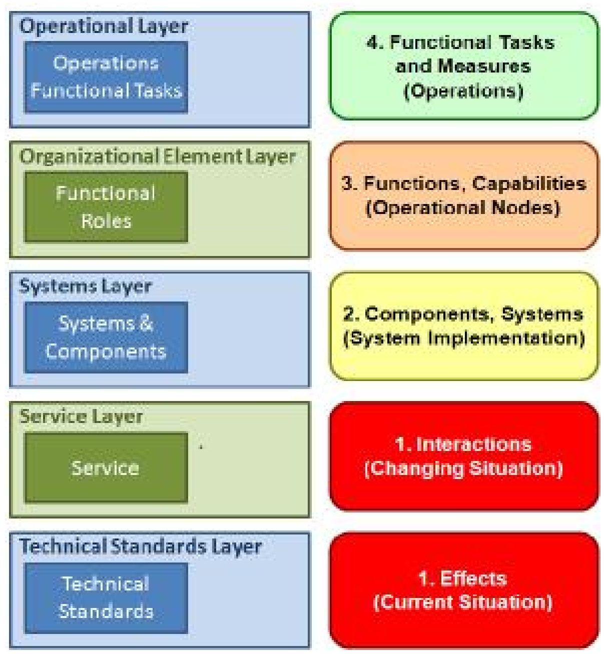

Figure 3: Parallels of DODAF and MMF

- DODAF operations adapt to (1) fill the gaps in the emerging MMF functions and tasks and (2) affect change in the current environment/situation.
- DODAF elements/roles adapt to (1) change the way systems and components function within the roles of operations and (2) coordinate the emerging operations.
- DODAF Systems and Components adapt to (1) use services that provide information on the changing/evolving environment (current situation), and (2) deliver effects to change the situation.
- DODAF Services adapt to (1) deliver effects from the Systems and Components, and (2) deliver information on the emerging situation.

- DODAF Technical Standards adapt to capture information on the emerging ontology of the operational environment.

A second important detail is the necessity for two Static Views. One Static View is for the planning organization and second for the executing organization. This abstraction enables the Framework to represent the interconnections between the execution and planning.

Framework Overview

The blue top layer represents the planning organization that is influencing actions of the executing organization, shown in red at the bottom. Likewise the planning cycle is in blue, and the execution cycle in red. The original MMF depicted these cycles as shown in Figure 2. Other additions to the MMF include:

- Layering/aligning the lifecycle stages with DODAF (more on this later).
- Multiple organizations were added to represent the high interconnectivity between organizations.
- Multiple roles or organizational elements were added to represent the use case actors of operations.
- The planning cycle is represented as being within, but more accurately it should cycle ahead of the execution loop, but not out pace it.

It is important to note that the lifecycles are strongly linked. If one outpaces the other they drift apart because the data collection (measurement and adjust) and decision making are then out of phase/synch with each other. This is why so many large system of system efforts fail. They simply cannot be planned top down then built bottom up. They approach must be incremental, and planned one or two cycles out. Cycles of execution changing the current situation cannot outpace planning cycles. If they do, planning and execution drift apart. Cycles of planning must measure (through execution) and adjust to keep up with the changing situation.

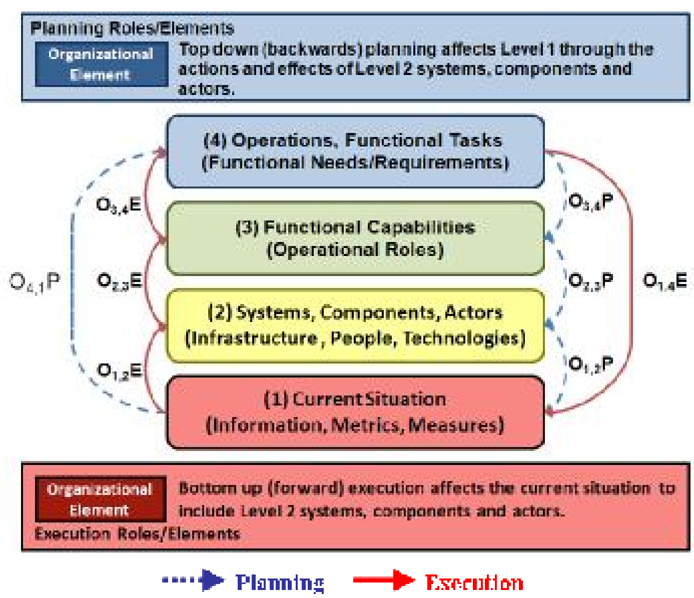

Figure 4: The Combined View

This can be interpreted as grounding; planning grounds execution, and execution grounds planning. These cycles are not reciprocals, but instead form a symbiotic relationship, a mutual purposive existence.

Dynamic View – The MMF

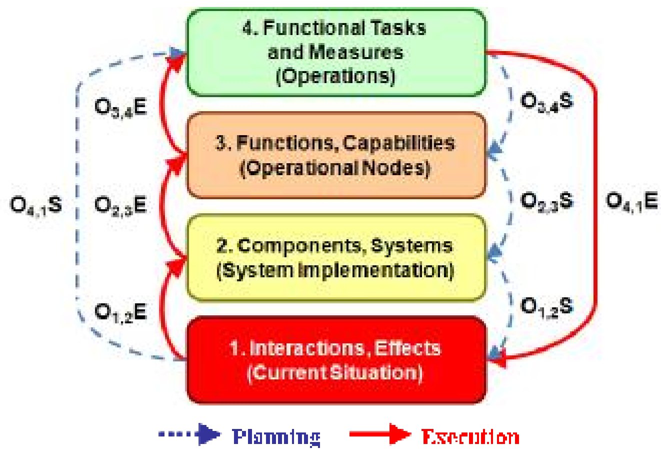

Figure 5: Dynamic View

The planning cycle represented in blue is informed at Level 1 where execution and planning converge from the bottom up. Execution informs planning of the functional gaps (needs) remaining after a planned action/artifact is fielded or injected into the current situation. The planning and executing lines/connectors are operators or transformations. The specifics of those fall into the area of models, physical and HSCB. More importantly they are large scale components of the model through which information flows.

Returning to Figure 1, the execution cycle represented in red is informed at Level 4 where planning and execution converge from the top down. Planning informs execution of the capabilities or changes that are coming and enables the execution agents to mitigate the risks associated with the gaps while solutions are being prepared.

Adaptation and Evolution

Adaptations are planned top down in blue. After fielding, the situation changes (possibly irreparably), leading to the "evolution" of PMESII actors during execution. In the original life cycle publication the "Missions and Means Framework", the execution life cycle was depicted like a domino effect. (1) Systems get damaged in the current situation, (2) systems, components and actors are no longer available, (3) functions and capabilities are adversely affected, (4) Full Spectrum Operations are compromised. However, there is much more going on here. Sometimes changes like black swan events change the situation irreparably. Evolution occurs out of the necessity of adapt to the new situation. In biology and nature, this evolution includes competing different adaptation, the best lead to longer life and/or increased rates of reproduction. In this case the framework enables organizations to model and estimate benefits of approaches w/o having to deploy them first.

Now consider that evolutionary changes occur bottom up. The current situation changes in a way that means the methods previously used (operations, roles, systems, services, standards) are no longer sufficient to deal with the changing situation and resulting effects. The need for change flows bottom up, initiating adaptations that lead to an evolution of function across the organization. Bottom down. What social models might apply to give us insights in to the emerging needs that drive evolution? Once changes are put in place, top down, what is the effect on the lower layers? Does the effect amplify and stay constant? Amplify then dampen? Or does the effect oscillate and tear the organization apart? This is a class of problem that has already been solved in any number of disciplines. The framework cannot be implemented from the top down. Just like evolution, the framework must be implemented from the bottom up, but planned from the top down. Each discipline of PMESII elements already has models of the respective operational functions. But how do we bring those together? What I'm getting to here is that existing models define the states and transformations applicable to PMESII elements. But no model tells us how the influence one layer has on another drives adaptations and evolution. Evolution occurs through competitive mechanisms, which behaviors are more effective than others? Many will be destructive. The greater the influence the greater the potential for instability. This seems to fit the concept of Weber–Fechner law:

> "The Weber–Fechner law attempts to describe the relationship between the physical magnitudes of stimuli and the perceived intensity of the stimuli. Ernst Heinrich Weber (1795–1878) was one of the first people to approach the study of the human response to a physical stimulus in a quantitative fashion. Gustav Theodor Fechner (1801–1887) later offered an elaborate theoretical interpretation of Weber's findings, which he called simply Weber's law.
>
> In one of his experiments, Weber gradually increased the weight that a blindfolded man was holding and asked him to respond when he first felt the increase. Weber found that the smallest noticeable difference in weight (the least difference that the test person can still perceive as a difference), was proportional to the starting value of the weight. That is to say, if the weight is 1 kg, an increase of a few grams will not be noticed. Rather, when the mass is increased by a certain factor, an increase in weight is perceived. If the mass is doubled, the threshold called smallest noticeable difference also doubles."[16]

For example, personal conflicts between a town's police chief and mayor would negatively affect the town greater than personal conflicts between two neighbors. It might lead to personal conflicts between many neighbors, citizens, families.

Competition

Competition is represented in four ways. First, the operational elements of the planning and execution organizations each compete for individual interests, but must balance that by cooperating for the mutual good. Second, in the lifecycle the operators $O_{3,4}P$ through $O_{1,2}P$ compete the implementations fielded into Level 1. Third, there is competition or friction between the planning organization and the executing organization. Simply put, execution never goes according to plan, and there is the rub. Forth, there is competition between the complex scale and the large scale. The problem is in determining what processes are centralized to an organization or organizational element (individual), versus what processes are shared or distributed across the lifecycle (group).

Note, $O_{2,3}P$ represents a trade off analysis of what systems/actors can fulfill the needed functions/capabilities. This implies a basic level of creativity (depending on your definition of creativity).

Let us examine the nature of change that occurs when execution does not go according to plan. Yaneer Bar-Yam describes the separation of completion and cooperation by levels[2] in his book "Making Things Work, Solving Complex Problems in a Complex World."

From the bottom up, players compete (try out) to be on teams. Players then cooperate to compete against other teams. Teams cooperate to promote their sport. Sports compete for fans and money.

“The basic point here is this: the interplay between competition and cooperation can only be understood by using a multilevel perspective. Competition and cooperation will tend to support each other when they occur at different levels of organization, but they will generally be in conflict if they occur at the same level.”[2]

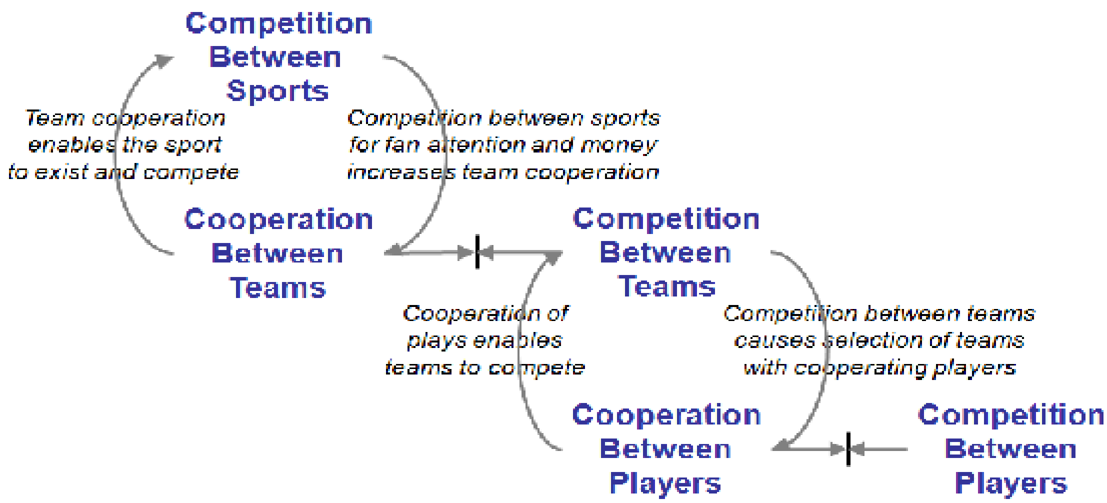

Figure 6: Competition and Cooperation in Sports[2]

Through inspection, we find that cooperation and competition do occur at different levels of the Framework:

- Level 4 Operations – Operations compete for roles/resources. Operations cooperate to change the operational environment (current situation).
- Level 3 Functions and Capabilities – Roles compete systems to be applied to operations. Roles cooperate to perform operations.
- Level 2 Systems – Systems compete services for information exchange. Systems cooperate to function together within roles of operations.
- Level 1 Services – Services compete to deliver the effects of systems to the current situation and to provide. Services cooperate to deliver information on the current situation to the systems.
- Level 1 Effects – The delivery of effects compete to change the environment. The environment is changed by the combined effects (cooperation).

Consider these implications in context to the planning cycles/organization and the executing cycles/organization. Levels generally compete in top down planning. Levels generally cooperate in bottom up execution. When competition and cooperation occur at the same level, dysfunction is the result. For example, recent politics provide a good example. During elections (planning) parties compete for representation and influence.

After elections (execution) elected officials work together to conduct business. Partisan efforts after an election represent miss-placed competition in a tit for tat cycle of rhetoric not unlike the Hatfield–McCoy feud. Nothing good comes from miss-placed competition, everyone loses. No one wins.

Influence, Intervention, and Interdependency

Influence and intervention can represented by the Planning organization/layer. Under normalcy, this layer simply represents the planning roles of the organization that executes the plan. However, as depicted in the figure, we have a planning organization that is intervening with the executing organization to influence their behavior. In this case the Framework is used to define the relationship between the Intervening organization and the PMESII organization being influenced.

In this context influence is the direct or indirect effect of affecting the executing organization's planning cycles. Intervention is the direct effect of affecting the cycles of execution. Intervention leads to dependency, while influence does not (by my definition). The difference is that once intervention has occurred you can't easily remove yourself from the situation w/o affecting the balance of life cycles and the current situation.

Consider a teacher. A teacher effects a student's ability to plan but does not intervene in their decisions on how to live, plan or execute. The dependency chain is not created.

At Level 1 and Level 4 the cycles of execution and planning converge. These are the points in the life cycle where metrics on the lifecycles is measured and interchanged. Planning and execution informed each other. The rate of cycles vary, they are not constant. Without the ability to measure then adjust, one lifecycle would outrun another. When execution over-runs the plan, decisions are no longer informed. When planning out runs execution, the plan drifts, it is no longer grounded to the situation because planning is no longer informed by reality. Planning relies on expectations and perceptions of the planning organization.

A good example of this is found in the National Security Archive "The Soviet Experience in Afghanistan" [10]

> Afghanistan did not fit into the mental maps and ideological constructs of the Soviet leaders. Their analysis of internal social processes in Afghanistan was done through the conceptual lens of Marxist-Leninist doctrine, which blinded the leadership to the realities of traditional tribal society. Believing that there was no single country in the world, which was not ripe for socialism, party ideologues like Mikhail Suslov and Boris Ponomarev saw Afghanistan as a "second Mongolia." Such conceptualization of the situation led to the attempts to impose alien social and economic practices on Afghan society, such as the forced land reform.
>
> The Soviet decision makers did not anticipate the influential role of Islam in the Afghan society. There were very few experts on Islam in the Soviet government and the academic institutions. The highest leadership was poorly informed about the strength of religious beliefs among the masses of the Afghan population. Political and military leaders were surprised to find that rather than being perceived as a progressive anti-imperialist force, the Afghanis as foreign invaders, and "infidels." Reports from Afghanistan show the growing awareness of the "Islamic factor" on the part of Soviet military and political personnel. [10]

Autonomous Subunits

Now consider any element of the planning or executing organization. Each element is an autonomous subunit, a complex organizational element of its own. Each can be represented by this framework. The interconnectivity between elements provides the potential for cooperation and competition. The interconnectivity between the planning and executing organizations provides stability, the lack there of results in dysfunction as planning and execution drift apart, neither cycle is enabled to properly inform the other.

The MMF can be used to represent this drift. Consider an organization comprised of eight organizational elements (operating systems). Each organizational element is interdependent of each other organizational element, a mistake by one affects them all, but in different ways. This would indicate, the actions of any one element affects all others across both cycles of planning and execution. A single mistake can result in a "defect explosion." A first order model estimate of a single mistake at Level 1 results in a conservative estimate of 8192 possible adverse effects (8 roles/elements) in a single lifecycle.

This estimate of defect rate forces the question "How we succeed at all?" The answer is of course is that through planning we collect data and make informed decisions. We analyze and "wargame" options before putting them in place. The problem is knowing what data to collect and what decision to make. Large scale integration keeps all elements informed, leaving the complexity and complexity to individual elements. But more happens, the high levels of interconnectivity distribute creativity and complexity across the organization. Large scale integration enables autonomous subunits to collaborate, creating a collective distributed complex organization.

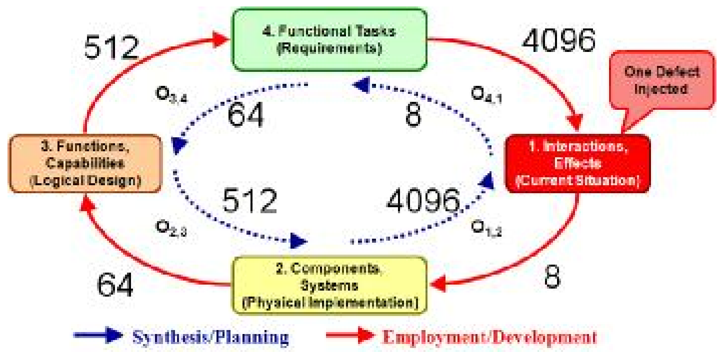

Figure 7: Dynamic View

Complex and Large Scales

The nature of the complex and large scale aspects of an organization can be viewed as a competition between the individual complex elements of the organization and the common mutual purposive needs of the other organizational elements. Most people have witnessed a failure in command and control that could have been avoided had management's decisions been informed by facts you were already aware of. But when this insight was pointed out to their superiors, management responded with "I don't need your permission to make decisions!" This is a good example of the difference between what someone *wants* to happen and what *needs* to happen. The transition from a centralized, hierarchy, to a decentralized highly interconnected/networked organization occurs through necessity. Giving up centralized control to enable distributed control scales people, but it is necessary for progress. This is the competition that drives evolution of an organization/actor.

Data collection is the large scale part of the problem. When an organization's elements are highly centralized, information is tightly controlled, hard to collect, nearly impossible to verify or validate. North Korea provides a good example of this.

When an organization's elements are distributed, data collection is enabled. It is easier to define, but and implement. Data can be verified and validated due to the distribution of organizational elements which provide for multiple perspectives from which the data can be analyzed. Data collection is conducted throughout the lifecycle. Data collection on planning, informs execution.

Data collection on execution informs planning. This mutual feedback ensures neither organization “drifts” away from the other. Decision making is the complex scale of the problem. The significance of large scale integration is that it makes an organization more complex. This insight has implications to data collection and operational security.

Emergence

Emergence in the dynamic view occurs largely because the framework imposes no central point of control. Levels of the lifecycle are peers. The planning and executing organizations inform each other and impose influence through information sharing, but don’t control each other. The nature of emergence will be highly dependent on the interconnections of the Operators.

Static View – The DODAF

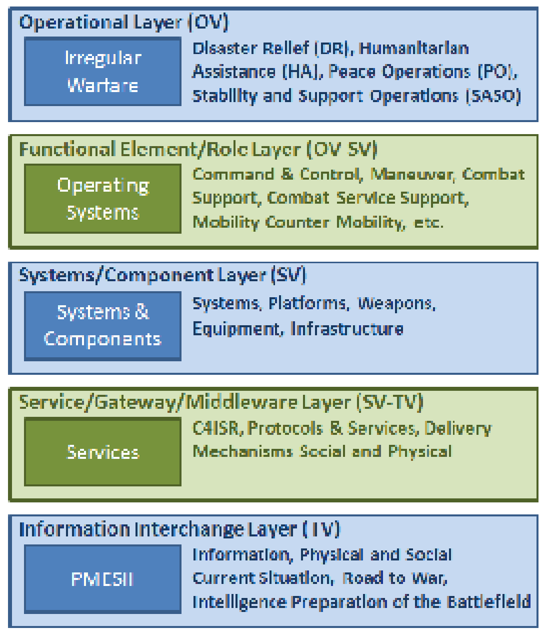

Figure 8: Static View - Military

Figure 8 depicts a conceptual Static View of the US Military. The OV layer represents operations of IW and full spectrum operations.

The OV-SV layer represents the operational roles of a military organization. Figure 8 is representative of the planning organization/layer of Figure 4.

The SV layer represents the systems employed in IW operations focused on specific operational roles of operational elements.

The SV-TV layer determines which systems receive what PMESII information.

The TV layer determines the information made available through services to systems that ultimately inform the operations.

Note the missing ontological layer. In order for information to be shared across organizations that interact and influence each other, a common ontology is required. Otherwise the shared information will not result in a common understanding between groups. This is a significant part of the problem. The PMESII layer represents the common ontology of the framework. It is like a blackboard available to every element of the organization. Each element can analyze and act on the information found on the black board leading to a better understanding of the current situation. Likewise each element can contribute to the blackboard by changing the current situation through execution.

Organizational Scope

What separates one organization from another can largely be viewed as differences in ontology. One group views the world differently than the other, and therefore they operate differently in the same context (OV), use different operational elements/roles (OV-SV), different systems (SV), different middleware (SV-TV), and they measure success in different ways (TV). As noted previously, differences in the level of interconnectivity drives the distribution/centralization of complexity. The scope of large scale integration is the primary factor that leads to behavioral emergence of the organization.

Figure 9 depicts a Static View of the PMESII organization. As with Figure 8, the TV layer represents information on PMESII. What differentiates these two layers is the contrast between the inter-view (shared/external) and intra-view (secured/internal). The PMESII organization will encapsulate information that is sensitive to its operation and security. The PMESII organization will publish/share information to inform other organizations.

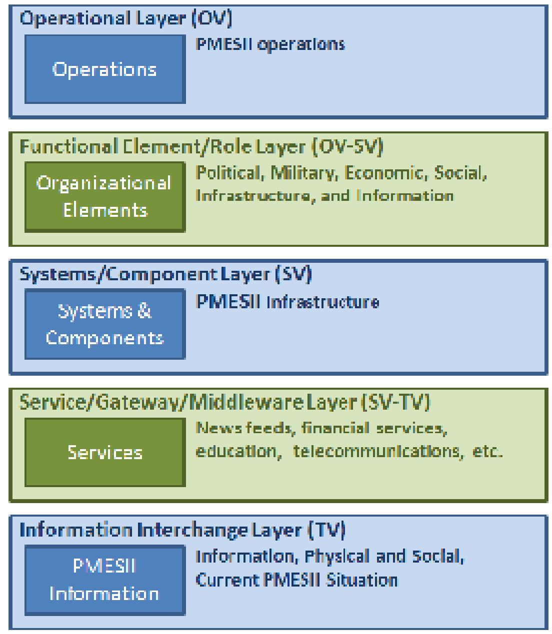

Figure 9: Static View - PMESII

Measures/Metrics and Technical Standards for Collection

As noted earlier data collection is a challenge in knowing what data to collect. How can you know when you are missing a key metric? Correlation analysis is the key. If the variance of a measure rivals the mean value of the measure, then something else is correlating to create that variance. Under the discipline of statistics, this is referred to correlation analysis.[11] The concepts of Personal and Team Processes and from Carnegie Mellon's SEI, presented later in the paper, describe this relationship well.

Emergence

Emergence in the static view occurs from the bottom up. Changes in the information layer cause the services to adapt by routing information in new ways to the systems and components. This delivers new information to the systems and components that integrate within the organizational elements. This creates new connections between the information and organizational elements, which then adapt by changing how systems and components integrate with the organizational elements. Finally these changes to organizational elements cause behavioral changes in the operations of the PMESII actor. Evolution, and adaptation are emergent, beginning with the information layer that is the current PMESII situation/context through to the behaviors and operations of the PMESII actor. As with the dynamic view it is the interconnections that give rise to emergence. The difference here is that the interconnections occur horizontally and vertically. For example, horizontally the Organizational Elements are interconnected in order to cooperate in the execution of PMESII operations. Remember competition occurs at different levels. In the static view the levels of competition are reflected the green layers. Evolution and adaptation occur by changing how services connect to information, and how PMESII operations connect/integrate systems and components. Relatively speaking the blue layers are much more constant/static and resistant to change. The green layers are dynamic and ever changing enabling adaptations and evolution to emerge from necessity with limited destructive impact to the blue layers. The components of the PMESII actor remain intact, while the "culture" integrating these components changes. This implies that failing nation states are failing to adapt or evolve, not because of the component systems or organizational elements, but because of how they are interconnected. Reform can then be viewed as changes to how information flows and how decisions are informed at these two levels. We can now appreciate why dictators simply cannot be complex enough; the challenge in effective information flow denies control, it is too dynamic and complex.

Combined View – The Framework for IW

In the Combined View for IW, the planning organization is reflected as the intervening military organization. The executing organization is represented as the PMESII actor which the military is intervening with. This is a direct result of instability and conflict inherent in irregular warfare. A military organization intervenes in the operation of a PMESII organization.

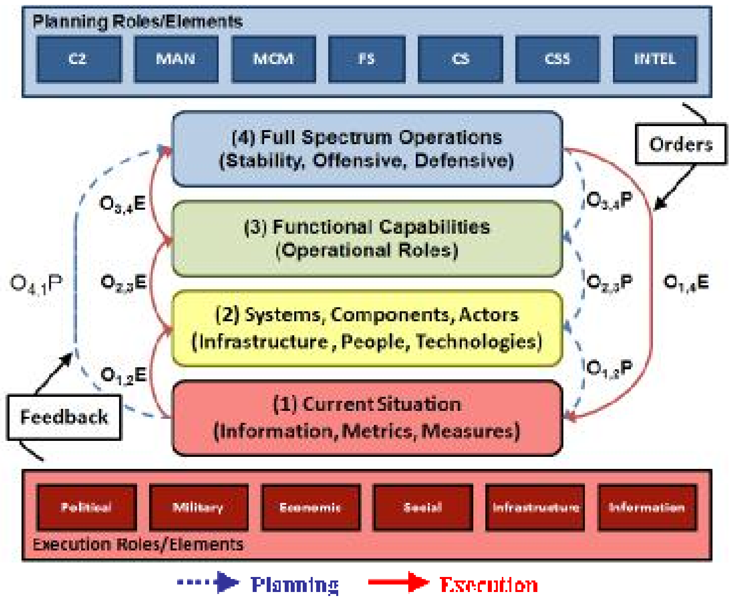

Figure 10: IW Multi-Disciplinary Framework

Level 1 represents the context in which the situation is changed by interactions and effects (social and physical) delivered by systems and actors. Level 2 systems and actors implement the functions and capabilities called for in the plan. Level 3 function and capabilities represent the functions required full spectrum operations. Level 4 are the operations of Disaster Relief (DR), Humanitarian Assistance (HA), Peace Operations (PO), Stability and Support Operations (SASO), Stability, Support, Transition and Reconstruction Operations (SSTRO), the Global War on Terror (GWOT), and other operation types. The planning elements plan top down from Level 4 through 1 informed through information feedback on the PMESII measures of performance in the context of tasks previously or currently executing.

It is important to note that C4ISR deliberately restricts the interconnectivity of information between operational elements of the military. However, developed PMESII organizations have a higher degree of interconnectivity and less hierarchy. Command and control is being decentralized to lower levels and a broader set of roles. This is the driving factor within the military today for smaller teams to utilize higher levels of command and control. The paper "The Evolution Towards Decentralized C2"[12] talks to the strategic corporal in a three-block war as follows; "The corporal performs many high-level command functions and autonomously directs his small unit."[12] This evolution of the military is a reflection of competition between the complex and large scales. The complexity of command is being distributed to lower levels of the military organization through the large scale nature of increased interconnectivity. Command and control becomes more distributed and less centralized. This has implications to the size of the command and control network. The growth of networks resulting from additions of new connections can be described as competing and/or cooperative approaches. When two or more different links provide the same result (are in competition), the one that creates the smaller network should be used.[18]

Data Collection

Data collection is critical to understanding the interconnectivity within an organization. It enables us to understand the current state of an actor and the environmental conditions that drive emergence of behaviors. Data collection is enabled by the large scale aspects of the organization, and is increasingly more difficult in organizations that centralize complexity of function.

This concept is not necessarily a handicap to the Framework. Knowing the level of large scale integration represents the critical information needed to develop a framework specific to the organization. With the framework in place the strengths and weaknesses of the organization can be determined. This enables one organization to influence another by being more complex. While information is easier to collect from the highly interconnected, distributed, complex organization, it is also harder to influence these organizations. These organizations adapt and evolve so much more readily, organizations of centralized complexity just cannot keep up. Centralized organizations cannot easily capitalize on the information available to them. In fact centralized organizations struggle to understand the distributed highly interconnected organizations. As discussed in the earlier sections covering Emergence, how can information flow be controlled if the centralized organization does not understand the interconnectivities themselves?

The point being made is that data collection is not as useful in context to centralized organizations. Due to the lack of deliberate interconnectivity, the random scale replaces the large scale resulting in an increased chaotic pattern of emergence. The solution is not to collect data but to evolve and mature the organization. The data matters less since the solution is the same. Data collection should be focused at diagnosing the organization to determine the specific Framework that applies. Consider a juvenile delinquent. No matter what the events and interactions were that led to the juvenile's delinquent behavior, providing a structured environment and building self discipline provides the solution to eliminating the emergence of delinquent behavior.

PMESII - Multidisciplinary Nature

Consider the challenges in data collection for a PMESII organization. If you wanted to monitor the health/stability of each PMESII element, what metrics would be measured, and how many would there be? This is an area where experts across a wide variety of disciplines would have to collaborate on the data to be collected and shared. That collaboration would require a common ontology or language to describe the information shared within and across PMESII elements. Without that ontology, the elements could not work effectively together. Each element's view of the world would drift without a common ground; blackboard.

Even if a common ontology did exist, how would someone go about determining the metrics of economics that could be used to measure the health of one PMESII element, such as the economy, and diagnose problems to resolve health issues? The following sections uses the Personal and Team Software Process (PSP/TSP) from Carnegie Mellon's SEI as a practical example. The lesson learned is that while this is a hard problem to define, once defined, the solution can be much easier.

Complexities of Software Development

People frequently think of software development as a straight forward process of design and implementation. We try to manage software development in a predictive, build to print approach, and are then surprised by project failures, cost and schedule over-runs. Frequently we blame these failures on changing requirements and rationalize that with statements like "The customer does not know what he wants." In other cases we deny the customers requirements under the belief that as developers we know best. "We are not going to give the customer what he wants, we are going to give him what he needs." We fail to understand that software implementations must be diverse and dynamic. One size does not fit all. Trying to predict the form a software project will take from state to finish, months or years, is like to trying to predict weather over that same time frame. Hindsight not foresight sometimes brings clarity into the form the software project should have taken, but rarely does it enlighten the software developer on what factors ultimately doomed the development efforts. To confront this need for keeping software development on tract Carnegie Mellon Software Engineering Institute (SEI) developed the Capability Maturity Model as means to measure maturity of a software development organization, and hence the organization's ability to repeatedly produce software products at a defined level of quality. However CMMI did not provide a clear path on how to reach maturity. The Personal Software Process (PSP) and Team Software Process (TSP) provide a guide for reforming a software development organization in order to achieve high levels of maturity through a holistic team approach of measurement and adjustment. There are three notable aspects of TSP/PSP that are worth mentioning:

- The application of TSP/PSP to an organization results in both higher quality software and increased productivity resulting in lower costs. You can have your cake and eat it too.
- TSP/PSP relies on releasing centralized control over software development decisions. Software development functions and decision making are distributed across the team and organization.
- Metrics are collected and shared seamlessly across the organization. Metrics provide a means to measure deviation of progress from the plan, enabling adjustment to the plan as well as to execution of the plan.

Large scale failures in software development abound, from the failed attempts to re-write the air traffic control system, to the Joint Simulation (JSIMS) efforts, and the more currently failures of the Future Combat Systems embedded battle command and training systems.

Personal and Team Software Process (PSP/TSP)

This section explores how TSP/PSP approaches data collection. What data is necessary and how it is used to measure and adjust the complexities of a software development organization.

Team Software Process is described on the Carnegie Mellon Software Engineering Institute web site (http://www.sei.cmu.edu/tsp/) as follows:

> Team Software Process (TSP) guides engineering teams that are developing software-intensive products. Using TSP helps organizations establish a mature and disciplined engineering practice that produces secure, reliable software in less time and at lower costs.

> TSP has been applied in small and large organizations in a variety of domains with similar results on first use, including
>
> - productivity improvements of 25% or more
> - reductions in cost and schedule variance to less than +/- 10%
> - testing costs and schedule reductions of up to 80%

TSP combined with the Personal Software Process provides a "How to" guide for achieving high levels of capability and maturity for a software development organization. Team work and data collection are strong tenants of the approach. What is interesting about TSP/PSP is the analyses that lead to the data collection approach. For TSP/PSP to work effectively, the team must work effectively first.

The approach is both simple and powerful. As with most problems, the hard ones to define are the easy ones to solve.

Process Quality Index (PQI)

Data collection is driven by the need to estimate deviation of execution from the plan. The approach to data collection begins with simple things. First data collection identifies the metrics to be measured. For PSP metrics these are:

- Size – Source lines of code (SLOC)
- Effort – Time on Task
- Schedule – Schedule and Work Breakdown Structure
- Quality – Defect Rates

Quality Indicators then use these metrics to establish measures of effectiveness for each phase of personal software development as follows:

- Standard Design Time >= Coding Time
- Standard Design Review Time >= 50% Design Time
- Standard Code Review Time >= 50% Coding Time
- Code Quality <= 10 Defects/KSLOC
- Unit Test Quality <= 5 Defects/KSLOC

Each Quality Indicator is normalized for values between 0 and 1. The overall PQI for any software development task becomes the cumulative product of these indicators.

What is significant here is that only four metrics need to be measures from five perspectives of the quality indicators. Data collection is very focused and well defined.

Complex and Large Scales

The TSP and PSP rely on process tools to integrate the complexity of software developers within a team by enabling rapid measurement and adjustments of metrics that inform team decisions on how to adjust the plan or execution. These tools enable managers to measure the progress of a team's effort as well as the quality of the products produced, all without the team having to directly interact with management. The large scale integration of process tools creates a distributed means of information management. This loose coupling makes each team autonomous, they lead themselves in a highly networked environment, but are tasked by management. The tools used are referred to as the PSP and TSP dashboard. The PSP dashboard collects metrics on an individual's software development efforts (coding time, design time, defects, etc). The TSP dashboard uses that information to project team status, compare status to the plan, adjust the plan, and processes to keep the program on track. The dashboard is in effect the individual and team blackboard through which information about the current situation is shared in a common context.

Measurement Approach

In TSP/PSP the planning team is the executing team, so naturally the data collection is easier. The hard part is establishing common process tools. These tools are used across the organization enabling the collection of metrics covering all aspects of the software development team. The metrics are classified by Size, Effort, Quality, and Schedule. Earlier the point was made "don't change the simple things". The exception comes in large scale integration. What appears to be simple for the individual can be complex to the group. This is an example of how factoring the complex (software developer) out of the problem space through large scale integration benefits not only the team, but the larger organization.

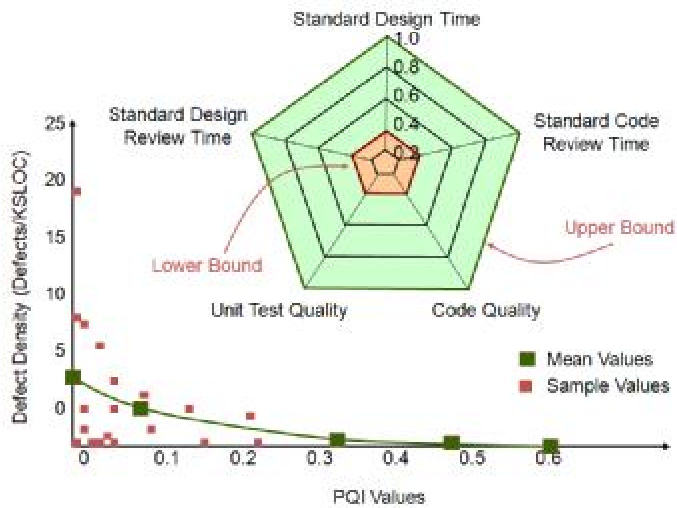

Figure 11: Process Quality Indicator

In training the process, one quality improvement is made each day (iteration), code review, design/design review, unit test, unit test review. The PQI is then calculated for each student's assignment, the mean is taken across the class, and the results are displayed as a trend in Figure 11. The green "Mean Values" are the class's average PQI for each day. The red "Sample Values" represent the defects found on the first day only.

This trend depicts the decreasing mean defects found in all phases of development per 1k source lines of code (KSLOC). Notice the variance in the sample values from day 1. A mean defect density under 5, but a variance of one individual of 15. Each day the mean decreased, and the variance tightened. In order to level set the proper ratio of mean to variance, "normalized" (0 to 1) standard quality factors were specified.

Early Warning System

Remember that when the variance of a measure rivals the mean value of the measure, then something else is correlating to create that variance. In PSP training, one step is added to the software development approach each day and the quality indicator of that step is collected. The variance continues to drop. The insight here is that Watts Humphrey had to determine what the principle indicators of quality were for the process to be reliable and have a low variance related to software quality and productivity (enables prediction of both within a defined bound; means and variance). This enables the "Dashboard" to model the developer's efforts so it can be used by management and the team coach to intervene before things get out of control; an early warning system is in place.

Measures of stability were defined as PQI values above .4, historic data indicates values under .4 have failed and must repeat the development stages indicated by the standard quality factors. Meaning the development stages that resulted in low quality indicators must be repeated. This is because all indicators are normalized, once the PQI drops to .4 due to one indicator, nothing can bring it back up except for rework and re-measurement.

Team Work

During the PSP course a question emerged. *When teams in an organization don't cooperate doesn't that make the effort futile?* The answer is yes, remember the 8192 adverse effects? Team Work is not an option if a complex organization is to function through mechanisms of large scale integration. Under the Team Software Process (TSP), there are eight lead roles:

- Team Lead
- Planning Lead
- Design Lead
- Test Lead
- Quality Lead
- Process Lead
- Configuration Management Lead
- Customer Interface Lead

When the team launches a new project, there are eight kickoff meetings; one meeting for each lead role. The respective leads actually lead the team in their efforts. The Design Lead coaches and mentors other team members in their approach to design. The Design Lead does not do all the work they lead it. The leads are in fact leaders in their areas of expertise; they don't control the group, but instead enable success by distributing the effort across the team. A successful launch leads to a highly interconnected team. Connections are made across the eight lead roles, all roles executing all of the time. All roles interdependent, meaning the failure of one causes the team to fail. But the distributed nature of each role means the structure of the team is redundant, adaptive, and resilient.

Team Work and Misplaced Competition

If strong team work is not in place, members will complete for leadership and work share at the same level. In such cases, the organization focuses inward. Planning and execution no longer constructively inform each other. What results is similar to the partisan rhetoric that preceded the Arizona shooting in January of 2011. There can be no winners in miss-placed competition simply because there is nothing to be gained. This tragedy broke the rhetoric and restored a bipartisan if only temporary. President Obama and Senator McCain mended fences torn down during the presidential election. Still not everyone learned the lesson. Some level of party extremism will always remain, but should not be central to the decision making process as that will omit the other party and their constituents from consideration in the decision making process. Effective decisions must be informed by both sides of all issues.

Without team work there cannot be a common process. Personal cultures collide with negative effects that disrupt any effort to control or foster common process. Building the team is priority one. When a team comes together through defined roles, responsibilities and process/culture, they become an integrated collective group. What emerges is influence and team autonomy. For example in TSP/PSP individuals software engineers are subordinate to superiors, but the team emerges as a peer to management, each informing the other's decisions.

TSP/PSP lead managers do not do all the work that falls to their roles; they plan the work for the team. The team does the work (execution); the team leads measure the deviation from the plan in context to their roles. The team determines corrective action. Separation of competition and cooperation by levels does not necessarily equate to the levels of an organization, but of the functional roles; separation of roles.

The roles, not the leads, compete for decision making as organizational elements of the team. Each team member is a system or component that supports each role in varying ways. In TSP "launch" meetings individuals compete to lead one or more roles. But in the context of planning and executing software development, the leads cooperate through their roles for the common mutual benefit to the group, but also for personal individual gains (raises, etc). Process provides for the integration of the large scale into the individually complex behaviors of software developers. This in turn enables data collection, through the tools, to inform decisions of the organization's management.

Start with a data collection plan for a stable functioning organization. Expand from there. Simple things first.

Consider swarm systems in context of a team. Flocks of birds follow a relatively simple set of rules/constructs; don't bump into one another, keep up with your neighbors, and don't stray too far away. However, there is a second construct, the basic patterns of flight and size of the bird must match. Imagine a goose flying in a flock of starlings. The goose is not able to maneuver as well and would result in many collisions. This is analogous to cultural mismatches where the swarm behavior of one culture collides with another. The mistake frequently made is to try to eliminate the collisions by eliminating the other culture. Success is found in reforming cultures so that they work together, cooperation vs. competition.

For PMESII, there is an analogy to swarm behavior. Each PMESII element is like one or more swarms. The rules/constructs of these swarms can be viewed as distinct cultures within each element. Each PMESII factor has an individual culture with associated behaviors. Immature PMESII actors frequently have a center of gravity that is centralized to one of the PMESII elements, most commonly Political or Military in nature. Mature PMESII actors on the other hand are much less centralized. From the integration and cooperation of PMESII elements emerges as a holistic decentralized organization with a common culture built upon the necessity for mutually purposive goals and behaviors to the benefit of all members. This decentralization means that the failure of any one PMESII element will not cause the actor to fail. The actor is more stable, resilient and as a result denies attempts to control it. The organization becomes mature and self managing, growing to become a contributing adult in the global community.

Effective team work gives rise to distributed command and control. Without it an organization cannot adapt and evolve effectively. Team work requires the sharing of information that results in a common understanding of the current situational context. A virtual blackboard emerges as the common context for information interchange. The sharing of information through this blackboard, provides opportunities for data collection that in turn enables the application of the Framework.

Conclusion

Most publications on complexity are intriguing and enlightening, but also end with an anti climatic tone. They give us insights into complexity, but the diversity of solutions tends to sideline the whole topic because there is no common holistic process that can be applied to solving complex problems. This framework is an attempt to answer that short fall by modeling:

- Organization – The organization's functions, roles, and component systems. The standards through which information and resources are interchanged.
- Behavior – The interconnectivity of organizational elements internally and externally as well as the common mutual purposive goals/behaviors the elements integrate with emerge as a common shared culture.
- Environment – Current situation of the environment is the common context shared by all members of a complex organization. The environment derives the necessity for adaptation, reform, and evolution.
- Life Cycles and Timing – The timing of how lifecycles of planning and execution measure and inform each other. Proper timing keeps the planning and execution elements from drifting apart.
- Evolution and Adaptation – The push and pull interplay of planning and execution cycles which change the situation through informed actions enables organizations to be introspective, to analyze and consider the benefits and consequences of actions before they are taken.

Now we have a framework, a tool for analyzing (1) what data is necessary and (2) the possibilities of how our decisions will affect stakeholders across all of the PMESII variables. Many questions remain:

- Can we identify the metrics that integrate PMESII elements?
- Can we define the lifecycle operators/transforms specific to each PMESII element?
- Can we identify the multi-disciplinary models to be integrated?
- Can this framework be used to instantiate a decision support model of?

Today modeling and simulation is applied to training. It is used to reduce costs related to training and the environmental damage that would result from live fire exercises. Can this framework enable the M&S community to reduce the cost of IW on a greater scale?

These questions cannot be answered by any one individual. The answers are too complex to emerge from a single centralized individual or discipline. A larger team/organization is required with well defined roles and processes. In order to take this Framework forward scientists and experts from all PMESII related disciplines will have to work together. The need for change is clear.

The obstacles for change are within all of us. It has been argued that evolution does not necessarily apply to human societies. In too many cases top decision makers repeatedly apply the same course of action in spite of the fact that they lead to disastrous outcomes.[21] "If you do what you have always done, you will get what you always got"; Mark Twain. A mixed disciplinary culture of mutual progress will have to emerge with common process, open information, and distributed decision making. It will require commitments from many individuals interacting in a virtual community to accomplish this.

A Personal Note

I came to understand this framework out of necessity. I was part of a team working to integrate applications and development teams that collectively managed over 10 million lines of code. We had three people from very different backgrounds, who when combined emerged as a team that could solve complexity contractually, technically, and politically. Having lived the success and understood the tenants we had followed, I found I had this "thing" in my head, this concept that would give me no peace. I had to get it out. I brought in Jack Sheehan author of the MMF, and Stan Levine, a retired SES whose job it was to keep Jack and I grounded. Jack and I immediately found comfort in realizing we were not alone in our understanding that architectures and lifecycles are interdependent and dynamic. Together we arrived at a functional framework that integrated the DODAF with MMF, but none of us realized just how far we had actually come.

Reading "Making Things Work; Solving Complex Problems in a Complex World" convinced me that this could be a significant step forward. It provided formalisms that let me take the Framework forward. I realized I was only just scratching the potential formalism would bring. But how could one person research all the related literature across of the related disciplines? The task is simply too complex for any one person. You have to be complex to solve complexity. One single person is not complex enough when confronted with multi-disciplinary complexity.

When Dean Hartley created the DIME/PMESII, HSCB, and IW, I seized the opportunity and began contributing and making contacts. In this context I challenge our members to (1) define the measures/metrics that bind the PMESII elements together, (2) identify models appropriate to each PMESII element, (3) pursue sponsorship to analyze the complexity of creating a generalized PMESII model based on this framework.

I am putting this work on the "blackboard" of Dean Hartley's DIME/PMESII, HSCB, and IW site. I hope that our members will contribute to this blackboard; expand our mutual understanding of the PMESII elements and how they integrate. What might emerge?

References

[1] Watts Humphrey; "Team Software Process(SM)" Software Engineering Institute (SEI), http://www.sei.cmu.edu/library/abstracts/reports/00tr023.cfm, Carnegie Mellon University. November 2000

[2] Yaneer Bar-Yam; Making Things Work; Solving Complex Problems in a Complex World

[3] Bootstrapping Complexity; learning from self-organizing systems in nature and technology; - a remix of Kevin Kelly's book "Out of Control" (1994) by Andreas Lloyd 2009

[4] DIME/PMESII, HSCB, and IW, http://www.linkedin.com/groups?mostPopular=&gid=2550249

[5] Hermann Hesse, Magister Ludi, aka The Glass Bead Game

[6] Christopher Alexander, A Pattern Language.

[7] DODAF

[8] Sheehan Jack H., DoD DOT&E/C3I & Strategic Systems, Paul H. Deitz, AMSAA, Britt E. Bray, Dynamics Research Corporation, Bruce A. Harris, Dynamics Research Corporation, Alexander B. H. Wong, AMSAA (2004), The Military Mission and Means Framework TR-756, October 2004

[9] Jeff Abbott, Jack Sheehan, Stanley Levine, "An Executable Architecture Framework Built Upon Frameworks of the MMF and DODAF" Simulation Interoperability Standards Organization Interoperability Spring Workshop 2010.

[10] National Security Archive; "The Soviet Experience in Afghanistan: Russian Documents and Memoirs", ed. Svetlana Savranskaya, October 9, 2001

[11] Draper, N.R. and Smith, H. (1998).Applied Regression Analysis Wiley Series in Probability and Statistics

[12] M. S. Vassiliou, "The Evolution Towards Decentralized C2", 15th ICCRTS, Paper 054 2009.

[13] Paul E. McMahon; "Distributed Development: Insights, Challenges, and Solutions," PEM Systems; CrossTALK the Journal of Defense Software Engineering, Nov 2001 http://www.stsc.hill.af.mil/crosstalk/2001/11/mcmahon.html

[14] Paul Clements, Felix Bachmann, Len Bass, David Garlan, James Ivers, Reed Little, Robert Nord, Judith Stanfford; SEI Series in Software Engineering Carnegie Mellon Software Engineering Institute; Documenting Software Architectures Views and Beyond 2003.

[15] Max Kleiber, "Kleiber's Law", 1930, http://en.wikipedia.org/wiki/Kleiber's_law

[16] Ernst Heinrich Weber, Gustav Theodor Fechner, "Weber–Fechner law", Wikipedia, 2011, http://en.wikipedia.org/wiki/Weber%E2%80%93Fechner_law

[17] Niles Eldredge, Stephen Jay Gould, "Punctuated equilibrium", 1972, Wikipedia, http://en.wikipedia.org/wiki/Punctuated_equilibrium

[18] Dr. Marc Timme, Dr. Birgit Krummheuer, "Beating the Competiton", Max Planck Institute for Dynamics and Self-Organization, January 17, 2011, http://www.mpg.de/1019259/Network_connections?page=1

[19] Fredrick Brooks, "The Design of Design; Essays from a Computer Scientist"

[20] "The Mythical Man-Month"

[21] Raphael D. Sagarin and Terence Taylor, "Natural Security; A Darwinian Approach to a Dangerous World" Regents of the University of California, 2008.