Complex Systems Engineering in a Federal IT Environment:

Lessons Learned from Traditional Enterprise-Scale System Design and Change

Michael D. Norman
Systems Engineering Technical Center
The MITRE Corporation
Bedford, MA, USA
mnorman@mitre.org

***Abstract*—The fragility created by hierarchical organizational constructs crosses over into the design of many large scale IT systems that are distributed across an enterprise. This means, that for these systems, end-to-end system design comes from the top down, creating a situation in which all fragility rises up to the largest scales of the system; this is a result of these systems being centrally controlled, often at the top of a hierarchy. In order for enterprise systems such as these to augment or repair themselves, they must undergo a catastrophic, enterprise-wide failure and be reengineered, once again by top-down direction [1]. This is the opposite of resilient system design and represents a situation where federal IT can be very inefficient. The current climate in the US of proactive and aggressive infrastructure consolidation via the Federal Data Center Consolidation Initiative (FDCCI) and the National Defense Authorization Act (NDAA) only serves to further incentivize system designers to construct extremely fragile systems at both the application and infrastructure layers. This fragility puts these systems at greater risk for enterprise failure. An example of a critical federal enterprise IT system's design and resulting fragility when perturbed (i.e., via consolidation and modernization) will be examined in this paper. Engineering guidelines from a complex systems perspective will be recommended to counter this resulting fragility. These guidelines will be from both an IT and government policy point of view and are generalized for applicability to systems engineering outside the scope of just IT systems.**

Keywords—complex systems; complex systems engineering; critical infrastructure protection; enterprise systems engineering; decentralization; control structures; system resiliency; resilience

I. MOTIVATION

In the US, government programs have historically had control over the IT systems on which they depend. Due to the National Defense Authorization Act (NDAA) and Federal Data Center Consolidation Initiative (FDCCI), federal systems which had been operating autonomously are constrained in ways previously unimagined by the systems' designers. Two major US government thrusts with respect to IT systems are to decrease costs through consolidation and to increase system resiliency, with no clear guidelines as to how to balance these sometimes oppositional objectives. The US government has a strong desire to build resilient systems to compensate for anomalous behaviors that are either natural or the result of adversarial activity. We will discuss an example of a very large scale IT system distributed across an entire enterprise that was subjected to federally mandated efforts to simplify and reduce cost of operations. We will apply a complex systems perspective to our example to show that there are emergent phenomena that have both immediate and latent effects on the system due to consolidation efforts. These effects are antithetical to the government's motivation to design and field resilient systems. Consolidation has the strong potential to save significant amounts of money, but if the resulting effect on system fragility is not taken into account in the planning phases, it can lead to critical IT systems becoming more fragile; oftentimes, single points of failure are inadvertently created.

Decentralized control structures [2] occur in both natural and engineered complex systems. This pattern of control, which strives to push decision-making as far to edge of the system as possible, is the foundation of all successful complex systems [3]. We can apply this pattern to design self-healing networked systems that are imbued with agency. For example, if a management server were to be taken offline, any system endpoints that depend on a relationship with that server could recognize this and respond by searching for, and associating themselves with, another server *at their own direction*. Take, as a specific example, how a packet gets routed through a network. The direction of each hop is a decision made by the local router based on the dynamic condition of its relationships with the other routers it is connected to (i.e., it has a sense of agency); the path the packet takes is not centrally controlled, which is one of the reasons the Internet is so resilient. This agency is critical for the end-points of the system to make ecologically-informed (i.e., aware of all of the nuances of the operational environment) decisions in the interest of self-preservation. Out of these individual subsystems' efforts to preserve themselves, a collective system will emerge. This collective system is definitively resilient, because the system as a whole is now an emergent entity [4]. Let us compare this perspective with our real world example.

The example we discuss is a multi-tiered, hierarchical enterprise IT system that was subjected to consolidation and modernization. The program management office (which could be considered the central design authority) determined that the

978-1-4799-5927-3/15/$31.00 ©2015 IEEE

number of servers could be reduced by 80% (which would also reduce the manpower required to operate and sustain them to achieve potential capital expenditure and operational expenditure reductions). In order to accomplish this, servers that had previously been distributed at many geographic locations were consolidated to significantly fewer locations, with many clients being managed by servers and operators that were now far removed from the ecology of their clients' environments.

The assumption underlying this effort was that the best way to control cost was to reduce the number of components in the system; to make it easier for a single person to comprehend. Why have 30 small servers when you can have one huge server? Though infrastructural redundancy (i.e., failover subsystems) can increase a system's robustness (it should be noted that the cost of truly robust hierarchical system redundancy can be quite high), the control over the system is still exerted through hierarchical relationships, which is ultimately antithetical to the US government's goal of resiliency, as the centralized control scheme creates a system that is completely rigid and unable to adapt at local levels to ecological stressors.

II. Results

In our example, consolidation resulted in application limitations that were not foreseen by system designers. Additionally, the system had embedded subsystems that were not designed to handle the consolidated load such as an application database that reached its connection limit. These limitations would never have been exhibited if the system had not been consolidated. The limitations caused nonlinear downstream effects such as a connection queue slowly growing to its limit over the span of weeks or months until a tipping point was reached and systems were no longer able to communicate. Performance degraded slowly and then system functionality was suddenly and drastically impacted. Imagine a system operator who has defined a database-intensive task on multiple servers. Upon consolidation, the new enterprise server will be running all of these redundantly, leading to a resource-starved operational environment. These types of systems should be designed such that IT system operators have shared responsibility over their own subsystems and are not relying solely on a centralized authority for decision-making. In our example, the centralized management construct takes on the accountability for the operators' actions. When these emergent dynamics [4] occur (regardless of their location in the system), the central design authority must react to them and this is a slow, cumbersome process for federal organizations as they attempt to maintain centralized control of the system. A decentralized system, while still subject to nonlinear effects, will prevent failures of the type described from propagating through the entire enterprise while also allowing tactical decisions to be made by those who are closest to the action.

Just as the US government has placed limits on how much an IT system is allowed to organically grow (for example, through directions to follow FDCCI and NDAA), it should specify guidelines for the consolidation of IT systems. To bound one side of the equation without bounding the other can lead to undesired consequences as described for our example. By allowing cost-savings to be the main driver of enterprise IT architectural decisions, the US government may be creating an extremely fragile cyber system of systems.

III. Discussion

The first recommended guideline is that systems should not be consolidated to the extent that losing one logical or physical component causes the system to be so substantially degraded that it has no real utility. Of course, systems should be designed such that no component or subsystem, if/when removed, would cause the overall system to fail.

The second recommended guideline is that centralization of control in a hierarchical fashion should be acceptable at the local edges of the system. That is, there should be assurance that any centralization of control does not migrate too far inward from the tactical edge. No single system should compromise the enterprise. Figs. 1 through 4 depict idealized control structures. A hierarchical-network hybrid is shown in Fig. 3. No single subsystem can compromise the enterprise.

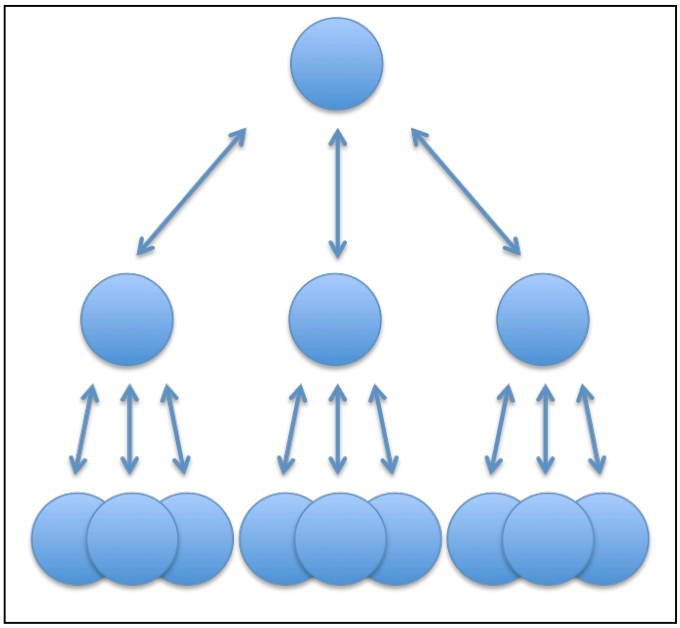

Fig. 1. Idealized hierarchical control structure in a consolidated environment.

The third recommended guideline for building truly resilient systems is that decision-making, whether it be details of implementation, human roles or anything else, should be pushed as far to the edge of the system as is possible. In order to prevent decisions (i.e., implementation details) from being made at a more local (i.e., lower hierarchical) level, it should be demonstrated that a dependency exists between the higher level and lower level, and that this dependency cannot be resolved via standard interfaces and/or functional equivalency. In other words, if the enterprise system can achieve its goals with any number of downstream implementations, then the design of the downstream/local subsystems should not be dictated at the level of the enterprise system. This actually leads to a more resilient system for a couple of reasons. First, this type of 'open requirement' enables an organic ability to evolve a heartier system variety by selectively choosing the

'best of breed' for future standardizations [5]. Second, the organic variety creates a much less fragile system and lays the foundation for the evolution of an ecosystem [1].

The fourth recommended guideline is that no one component/subsystem of any critical enterprise system should be allowed to grow so large that it subsumes other, formerly independent systems without generating internal redundancy to compensate for what would otherwise be an increase in fragility.

Relative fragility of critical IT modernization efforts should be part of the trade-space analysis done to support funding decisions. To generalize this specific situation further, we will utilize complex systems science.

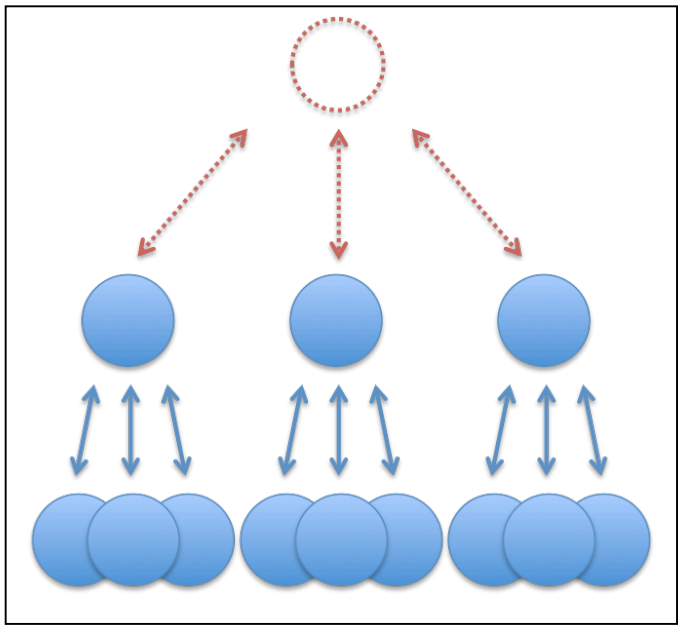

Fig. 2. Idealized hierarchical control structure after losing the centralized control subsystem in a consolidated environment.

In Figs. 1 through 4, each blue circle represents a subsystem and each arrow represents a relationship. The red dotted lines indicate lost relationships and the red solid lines indicate newly formed relationships. Likewise, the red dotted circles indicate lost subsystems. Fig. 1 depicts a standard hierarchical control structure. If the top-most subsystem is removed from the hierarchy, as shown in Fig. 2, the lower subsystems cease to function; this is not resilient design. Fig. 3 is a network-hierarchy hybrid control structure. If any top-level (from an edge-in perspective) node is removed from the network-hierarchy hybrid model, as shown in Fig. 4, the newly orphaned nodes can re-associate themselves with another parent. Since the migrated load is distributed across the control network, there is little risk of over-burdening any individual control subsystem. A network of control subsystems enables the control structure of the system to continue to operate even when a top-level subsystem is removed; this is resilient design. This dynamic behavior requires a certain level of agency be granted to the subsystems so they are able to appropriately re-associate themselves if communication with a parent-node is lost.

In this paper the limitations of hierarchical control structures have been demonstrated and solutions to surmounting them have been offered. By using the concepts of system fragility and complex systems to guide our design philosophies, novel recommended guidelines for federal IT policy have also been given, which, if followed, should lead to much more resilient system design.

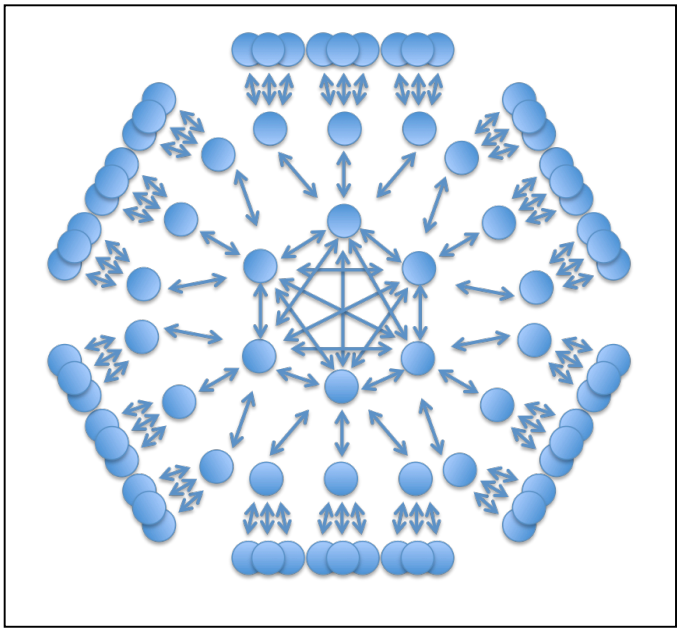

Fig. 3. Idealized network-hierarchy hybrid control structure.

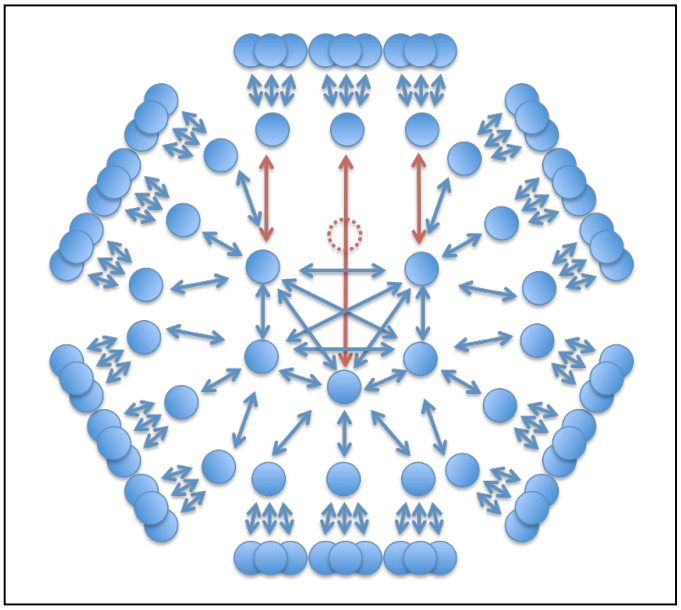

Fig. 4. Idealized network-hierarchy hybrid control structure after losing one node in the network of control subsystems.

REFERENCES

[1] N. N. Taleb, Antifragile: things that gain from disorder. New York: Random House, 2012.

[2] Y. Bar-Yam, "Complexity rising: From human beings to human civilization, a complexity profile," Encyclopedia of Life Support Systems (EOLSS). Oxford: UNESCO, EOLSS Publishers, 2002.

[3] D. D. Siljak, Decentralized control of complex systems. Boston: Academic Press, 1991.

[4] Y. Bar-Yam, "A mathematical theory of strong emergence using multiscale variety," Complexity, vol. 9.6, pp. 15–24, 2004.

[5] Y. Bar-Yam, Making things work. Cambridge, MA: Knowledge Press, 2004.