**Information • Knowledge • Systems *Management*** 2 (2001) 325–345
IOS Press

# On the Systems Engineering and Management of Systems of Systems and Federations of Systems

Andrew P. Sage and Christopher D. Cuppan

*Department of Systems Engineering and Operations Research, George Mason University,*
*Fairfax, VA 22030-4444, USA*

**ABSTRACT:** This paper is concerned with the engineering of systems that are themselves comprised of other component systems, and where each of the component systems serves organizational and human purposes. These component purposes may be locally managed and optimized independently, or nearly so, of the objectives to be met by the composite system. There are a number of inherent characteristics of these systems, and such related terms as systems of systems (SOS) or federations of systems (FOS) or federated systems of systems (F-SOS) are often used to characterize them. It is asserted that the resultant systems generally possess the characteristics of complex adaptive systems. We provide an overview of the literature describing these engineering efforts and provide plausible strategies for systems engineering and management of SOS and FOS that are based on the principles of a "*new federalism*". Finally, the implications of these plausible SOS and FOS systems engineering and management concepts are discussed with emphasis on evolutionary acquisition in the style of DoD and Intelligence Community related programs.

## INTRODUCTION

Systems engineering and management professionals are increasingly confronted with the need to consider the integration of human, organizational, and technological issues in all facets of life cycle activities that are intended to result in the fielding and support of systems. Given the current trends in IT globalization, they are predominantly facing distributed organizational difficulties and, quite often, various amounts of dysfunctionality. These problems are being induced as canonical management concepts are extended in an attempt to address international coalition development, integration, and management of widely distributed systems. It is generally understood that these extensively distributed systems both developmentally and operationally exhibit the behaviors of complex adaptive systems [CAS] (Axelrod & Cohen, 1999; Czerwinski, 1999; Kelly & Allison, 1999; Stacey, Griffin, & Shaw, 2000). However, if the state of the practice is any indication, an equivalent understanding of the dynamical properties of the engineering organizations affiliated with these CAS is rather uncommon. This is a formidable liability for the engineering profession. It appears to be a reality today that large, distributed engineering organizations exhibit CAS characteristics throughout the life cycles of the distributed systems that they are responsible for defining, developing, and deploying (Sage, 1992, 1995; Sage & Rouse, 1999). More to the point perhaps, these engineering organizations should in fact be recognized as complex adaptive systems.

One of the major contemporary characteristics of large systems is that they are often formed from a variety of component systems: newly engineered from the *"ground-up"* custom systems, potentially tailored existing Commercial-Off-The-Shelf [COTS] systems, and existing or legacy systems. The term *"System of Systems"* has come into common usage in the past few years to describe such systems (Owens, 1995; Gompert & Isaacson, 1999). Cook (2001) provides a number of interesting comparisons between a sim-

1389-1995/01/$8.00 © 2001 – IOS Press. All rights reserved

ple monolithic system, such as a radio receiver, and a complex system of systems, such as the Internet or a complete multinational peace keeping force. He also illustrates how these differences influence system acquisition.

This paper will discuss the system of systems concept, sustainable engineering of a system of systems, and related issues concerning evolutionary acquisition. We will be particularly concerned with command and control systems and intelligence community-related systems engineering organizations and will address engineering life cycle contexts accordingly. A review of the related concept of a federation of systems is also presented. As a major part of this, we discuss a newly rediscovered organizational concept, federalism. Federalism may be wholly appropriate to apply within contemporary, distributed, and often *"virtual"* organizations (Handy, 1995) in order to obtain the necessary effectiveness properties with respect to life cycle engineering of a distributed system of systems. The notion of *"effectiveness"* versus, and as a compliment to, *"efficiency"* is treated here as one major positive characteristic of a complex adaptive system that has achieved a sustainable balance (Holland, 1995, 1998). Such a balance can equivalently be viewed as a sustainable ecology and is a necessary ingredient in a systems ecology (Sage, 1998). The tailored application of the principles and maxims of the *"New Federalism"* (Handy, 1992; Krygiel, 1999) to DoD and Intelligence Community systems engineering organizations will then be discussed.

## SYSTEM OF SYSTEMS

The above references to a *"System of Systems"* capture important realities brought about by the fact that modern systems are not monolithic. Rather, they have five characteristics (Maier, 1998) that make the system of systems designation (Krygiel, 1999; Carlock & Fenton, 2001) most appropriate:

1. **Operational Independence of the Individual Systems.** A system of systems is composed of systems that are independent and useful in their own right. If a system of systems is disassembled into the component systems, these component systems are capable of independently performing useful operations independently of one another.
2. **Managerial Independence of the Systems.** The component systems not only can operate independently, they generally do operate independently to achieve an intended purpose. The component systems are generally individually acquired and integrated and they maintain a continuing operational existence that is independent of the system of systems.
3. **Geographic Distribution.** Geographic dispersion of component systems is often large. Often, these systems can readily exchange only information and knowledge with one another, and not substantial quantities of physical mass or energy.
4. **Emergent Behavior.** The system of systems performs functions and carries out purposes that do not reside in any component system. These behaviors are emergent properties of the entire system of systems and not the behavior of any component system. The principal purposes supporting engineering of these systems are fulfilled by these emergent behaviors.
5. **Evolutionary Development.** A system of systems is never fully formed or complete. Development of these systems is evolutionary over time and with structure, function and purpose added, removed, and modified as experience with the system grows and evolves over time.

We see that the operational concepts needed for a trustworthy system of systems process are not at all simple. There are many needs that must be fulfilled in order to be able to define, develop, and deploy a trustworthy system of systems.

It is very clear that the **system of systems** (**SOS**) concept is now ubiquitous across many contemporary systems engineering and management efforts. This term is becoming increasingly used in the systems engineering literature (Manthorpe, 1996; Carlock, Decker, & Fenton, 1999; Roe, 1999). It is natural to ask *"What is a system of systems?"* Unfortunately, there is no universally accepted definition of these *"super-systems"*. What distinguishes a system of systems from other systems does not, at this point, have a definitive answer. In a formal sense almost anything could be regarded as a system of systems. A personal computer is a system. However, the monitor, microprocessor, disk drive, random access memory are also systems. Thus, should we call a personal or a mainframe computer a system of systems? Is a theatre missile defense system or the Internet a system of systems? Instinctively, they are quite different. The big difference in the last two examples is that the former system may be massive in size but monolithic in purpose while the latter is capable of supporting myriad communications and commerce purposes. There is also a distinction based on operational and managerial independence of the systems components.

Many modern systems possess the following characteristics. The component systems achieve well-substantiated purposes by themselves and continue to operate in this way and accomplish these purposes even if detached from the overall system. The components systems are managed in large part for their own purposes rather than the purposes of the whole. Yet, they function to also resolve purposes of the whole that are generally unachievable by the individual systems acting independently. In other words, these ultimate purposes *"emerge"* from the SOS. It is not the complexity or size of the component systems or their geographic distribution that makes them a *"system of systems"*, although many contemporary systems of systems will be geographically distributed. Thus, geographic distribution can be viewed as a third characteristic. Another major characteristic that is useful in distinguishing very large and very complex but monolithic systems from a true system of systems is evolutionary development. A system of systems may not appear fully formed and functional initially. Its development is evolutionary in the sense that functions and purposes are added, removed, and modified with experience in use of the system. As a consequence of this evolutionary development, the resulting system of systems will have the property of emergent behavior whereby it functions and carries out purposes that are not possible by any of the component systems. These are the five characteristics of systems of systems detailed earlier. A system will be called a SOS when all or a majority of these characteristics are present.

Often, appropriate missions exist for relatively large SOS where there is a very limited amount of centralized control and authority. Instead, there is a coalition of partners that has decentralized power and authority and potentially differing perspectives of situations. An additional useful and related concept which describes such systems is termed a "Federation of Systems" (FOS). In these systems, there is little central power and authority for *"command and control"*. The participation of the coalition of partners is based upon collaboration and coordination to meet the needs of the **federation**. FOS are generally characterized by significant autonomy, heterogeneity, and, geographic distribution or dispersion (Krygiel, 1999).

There are many statements of needs for a SOS or a FOS. The current vision statement for DoD supporting the warfighter, for example, is embodied in the Joint Vision 2010 (JV2010) guidance document and its recent replacement, the Joint Vision 2020 (JV2020) guidance document. JV2010 and JV2020 require a true revolution in war fighting doctrine enabled by massive, complex and distributed software-intensive information systems. The following are among envisioned changes in JV2010.

> "By 2010, DoD must change the conduct of the most intense joint operations. Instead of relying on massed forces and sequential operations, massed effects will be achieved in other ways. Information superiority and advances in technology will enable us to achieve the desired effects through the tailored application of joint combat power."

Innovation was implicitly suggested as being necessary for the requisite transformation of forces. Importantly, the JV2020 document notes firmly that innovation includes but is not limited to technological innovation; it also includes organizational and conceptual innovation. It is further noted that accomplishing this requires continuous learning, a reasonable tolerance for errors, and experimental processes to accomplish the needed learning and accomplish the needed change.

The systems fielded in order to obtain these capabilities will not be monolithic structures in terms of either operations or acquisition. Rather, they will be SOS, if not FOS in most cases, which are integrated in accordance with appropriate architectural constructs in order to achieve the cooperative effects that will be required.

As previously noted, Krygiel (1999) has provided a hierarchic taxonomy of systems that shows the relationships between conventional systems, systems of systems, and federations of systems. She considers three dimensions: autonomy, heterogeneity, and dispersion. A federation of systems, or federated system of systems, will generally have greater values of these characteristics than a (non-federated) system of systems. The key points to consider relative to this hierarchy are as follows:

1. Many conventional systems are special purpose-built, as a mixture of Commercial-off-the-shelf (COTS) and custom developments of hardware and software. These are generally provided by multiple contractors who are used to supporting a specific customer base and under the leadership of a single vertical program management structure.
2. SOS integrate systems across communities of contractors, and sometimes across multiple customer bases, and are generally managed by more horizontally-organized program management structures, such as integrated Product and Process Development (IPPD) teams. When the IPPD team effort is efficacious, they are generally well able to deal with conflict issues that arise due to business, political, and other potentially competing interests.
3. FOS face the same dilemmas identified for SOS but are generally much more **heterogeneous** along trans-cultural and trans-national socio-political dimensions, are often managed in an **antonymous** manner without great central authority and direction such that they satisfy objectives and purpose of an individual unit in the federation, and there is often much greater geographic **dispersion** of organizational units and systems. Thus, the delimiters between SOS and FOS, while generally subjective, are nonetheless principled.

Clearly, the notions of autonomy, heterogeneity, and dispersion are not independent of one another. Increasing geographic dispersion will usually lead to greater autonomy and consequently also increase heterogeneity. The Internet is perhaps the best example of a system that began life under the aegis of a single sponsor, the US Department of Defense, and which has grown to become a federation of systems.

The complexities and difficulties associated with systems engineering and management programs required to field SOS or FOS are daunting. Often, they will be beyond the tractability and capability of canonical program management terms and practices. The Wentz (1998) summary of lessons derived from Operation Joint Endeavor, as discussed in Krygiel (1999), provides insight into the challenges and difficulties of achieving a SOS, much less a FOS. To succeed with the integration of communications and information systems (CIS) to support the implementation force (IFOR), Wentz states:

> “The challenge facing NATO and the nations was to build a long haul and regional CIS network out of a mixture of military and commercial equipment that would vary widely in age, standards, and technology and would be built very quickly once given the order to deploy. … Putting the pieces of the puzzle together would most likely not result in a true ‘system of systems’ for IFOR. Furthermore, there would be a need to interface systems that had not been planned or designed for interfacing. The independent national systems would be tied together, not engineered as a single system. … Given the uncertainty of the situation it would most likely be a case of integrating what you get, not necessarily what you need, and then making the best of it. … An effective process or organizational structure for engineering and integrating such system of systems was as elusive as the successful systems deployment results being sought, …and just as important.”

Questions need to be raised as to what are the appropriate systems engineering and management program organization and leadership concepts to apply within SOS and FOS domains. There is much evidence that the current focus on evolutionary acquisition is intended to allow these issues to be vigorously addressed within the DoD. It is asserted herein that we would be well-served to include the application of the concepts of *"federalism"* in these deliberations.

## FEDERALISM APPLIED TO SYSTEMS OF SYSTEMS

Handy (1992) has studied a new federalist approach, called a *"New Federalism"*, which speaks to the necessary considerations for the structuring of loosely coupled organizations in order to help them adapt to information and knowledge age changes. Many similarities can be found in Handy's (1992) New Federalism and Drucker's (1999) organizational concepts for surviving beyond the information revolution. In order to obtain a systems engineering ecology, in other words a sustainable systems engineering approach, the prospect of applying federalist political principles to SOS and FOS management makes a great deal of sense. This is predominantly because of the fact that engineering development alliances, generally taking the form of virtual organizations (Handy, 1995) or virtual teams, are becoming more and more commonplace.

The concept of federalism is particularly appropriate since it offers a well-recognized way to deal with the systems engineering management paradoxes of power and control such that the desired ecological balance is obtained. Generally, this is accomplished by:

- "making things big by keeping them small"[1];
- encouraging autonomy but within the appropriate bounds of process and architecture standards; and
- combining variety and shared purpose, individuality, and partnerships at national and global levels.

It is appropriate to note, again, that these characteristics are those of a successful CAS.

Federalism is not just an elegant word for restructuring conventional systems engineering and management lines of authority. It is based on five principles set forth by (Handy, 1992) that must be adopted, *"inside-out"*. A summary of Handy's five principles tailored to the domain of systems engineering and management follows.

1. **Subsidiarity** is the most important of federalism's principles. It means that power belongs to the lowest possible point within the FOS engineering team. Handy indicates that a higher order body should not take unto itself those responsibilities which properly belong to a lower order body. Managers are often tempted to subsume their subordinates' decision prerogatives. Subsidiarity requires, instead, that they enable those subordinates, by training, advice, and support, to make those decisions better. Only if the decision would substantially damage the FOS program itself and/or its objectives is the manager entitled to intervene. Subsidiarity is the reverse of empowerment in that it is not the FOS program manager who is giving away or delegating power. Instead, power is assumed to lie at the lowest point in the organization and should be taken away only by agreement between the engineering professional and project manager(s).
2. **Interdependence** is a reality in federalism. The autonomous development units or teams of a FOS development federation stick together because they need one another as much as they need management leadership and leadership authority. This concept is known as pluralism and it is in this sense that a federation is different than that of a confederation. Federalism encourages combination when and where appropriate but not the all too familiar centralization of services. It is only when combination and aggregation becomes excessive that combination offends the principles of federalism. Achieving this interdependence, or pluralism, is necessary in federalism because it distributes power. This avoids the risks of autocracy and over-control of the typical centralized

---

[1] In other words, expand the domain of an enterprise by instantiating multiple quasi-autonomous units. This is as opposed to acquiring mass by aggregation around a centralized base.

program management bureaucracy. Nonetheless, the SOS or FOS program management is still the focal point for action.

3. There is a **Uniform and Standardized Way of Doing Business.** Interdependence within federated SOS or FOS engineering organizations is unlikely, if not impossible without agreement on basic rules of conduct, common traditions of communicating, and common units of measurement of progress and quality.
4. **Separation of Powers** is required. Federalism requires that management, monitoring, and governing aspects of FOS engineering programs and projects be viewed as separate functions to be accomplished by separate bodies. The membership of these bodies may in fact overlap. When these three functions are combined into one body, short-term needs generally drive out long term considerations. Month-to-month management and monitoring *"issues"* take time and attention needed for governance.
5. **Dual Citizenship** exists in a federated engineering development project. Every individual is a "citizen" in two communities, the local development group/professional group/union, and the overall FOS program at large. Local citizenship seldom needs much support. The FOS program draws its strength from the strong leadership of the local groups. It is the federated *"citizenship"* that requires emphasis if the benefits of subsidiarity and interdependence are to be acquired by sponsors and customers of a FOS engineering program.

The concept of federalism is particularly appropriate for a FOS. This is simply because federalism offers a well-recognized way to deal with the systems engineering and management paradoxes of power and control that are characteristics of such systems such that the desired ecological balance is obtained. That is, the need to make things big by keeping them small; to encourage autonomy but within bounds (e.g., process and architecture standards); to combine variety and shared purpose, individuality, and partnership, at levels ranging from local to global.

Again, in order to obtain a federalized organizational ecology, these five principles must be adopted, *"inside-out"*, by the systems engineering and management organizations. The *"inside-out"*, or *"bottoms-up"*, restriction is significant. In any complex adaptive organization, such an SOS or FOS organization, it is generally the agents that really wield the control. Any attempt to drive an incompatible solution down from the top will be thwarted. As Holland (1995, 1998) observes for any CAS, agents will simply route-around any centrally imposed obstacle or problem in order to sustain their own benefits and to survive.

As we have noted, subsidiarity is the most important of federalism's principles and associates power with the lowest possible points within the systems engineering and management team. Taken seriously, subsidiarity is an awesome responsibility because it imposes on the engineering professional or group what might be called *"type-II"* accountability, or accountability for solving the correct problem. Traditional engineering management structures generally operate on the basis of *"type-I"* accountability, that is to say solving the problem correctly by making sure that no mistakes are made. The potential for a *type II* error are not considered. Under Subsidiarity in an engineering management context, people are judged against their *"type-II"* accountability through seeking answers to such questions as: was the correct problem or issue defined? Did they seize every opportunity? Did they make all the possible improvements? To be effective, subsidiarity has to be formalized by systems management, through such mechanisms as contracts. It has to be clear who can do what, how power is to be balanced, and whose authority counts where. Termination guidelines for people and organizations and procedures must also be formally instantiated and, above all, followed! This issue will be fully elaborated in the discussion of Managing Empty Spaces that follows. It is particularly worthwhile to point out here that the success of a complex adaptive systems engineering organization is based on the concept of survival of the fittest.

Interdependence is achieved partly through the *"reserve powers"* of the FOS project leadership, partly through the shrewd organizational planning of the FOS project leadership. Federalism encourages combination when and where appropriate but not the all too familiar centralization of services. It is only when combination and aggregation becomes excessive, such as when all program services are located in one place, that combination offends against the principles of federalism.

Achieving the needed interdependence, or pluralism, is a key to federalism because it distributes power. This avoids the risks of autocracy and over-control of the typical centralized program management bureaucracy. The FOS program management team/office still has to be the focal point; to talk and share. Telephones and videoconferences are no substitutes for real *"arm-to-arm"* meetings. Significant logging of airplane miles and *"red-eyes"* become inevitable. However, the exhaustion is clearly worth it (Handy, 1992). Paradoxically, this type of dispersion bonds the FOS groups and teams together. FOS engineering-related units who use each other need each other. Such is the generic nature of complex adaptive systems.

Management is the executive function, responsible for delivering the discrete *"engineering goods"* (e.g., engineering configuration items or CI's). Monitoring is the quality/judicial function, responsible for seeing that the *"engineering goods"* are delivered in accordance with the *"laws of the federation"* (e.g., architectural standards, build-to requirements, quality objectives). Governance (e.g., Senior Program Manager, Program Executive Officer) is responsible for overseeing management and the monitoring. The governance function is also responsible for establishing the a priori strategy, policy, and direction to be followed by the SOS or FOS development program. When these three functions are combined into one body, the short-term tends to drive out the long. Month-to-month management and monitoring *"issues"* steal the time and attention needed for governance. The big decisions often go wrong as a result. Thus, separation of powers is a major need.

In a federated engineering development project, strong leadership of the local groups results in the dual citizenship needed. This is another of federalism's paradoxes but the one that ensures a strong local identity, similar to the tagging mechanism of Holland (1995). This also results in the willingness to reach out and avoid committing the *"type-II"* (wrong problem) errors previously discussed relative to subsidiarity. Appleton (1994) indirectly reaches the same conclusions in his analysis of successful business forms and business process reengineering. It is the federated *"citizenship"* that requires emphasis if the benefits of subsidiarity and interdependence are to be reaped by the sponsors and customers of an SOS or FOS engineering program. This directly leads into the psycho-social notion of shared values as initially demonstrated by successful Pacific-rim corporations (i.e., what the Japanese would recognize as the *"spiritual fabric"* of their federations). It must be recognized that accomplishing this requires more than the groups need to improve the bottom line. According to Handy (1992), it must be some modern-day equivalent of the citizens of Elizabethan England collectively acting for *"the Queens Great Matter"* (i.e., world stability and strategic domination over competitors).

Once the five key principles of federalism have been ingrained, *"bottoms-up"*, within a distributed engineering organization, five additional guidelines, or maxims, must be brought into play. These maxims must be adhered to by the leadership, dispersed throughout the distributed organization, for the benefits of federalism to begin to be accrued. In actuality, the previous statements are not entirely correct. To wit, the necessary principles of federalism will never be ingrained into an engineering organization unless the five leadership maxims are demonstrated, in an apriori sense, as being standard operating procedure by the organizational leadership. A brief description of Handy's (1992) leadership maxims, within a systems engineering and management context, suggests the potential benefits to federalism.

1. **Authority Must Be Earned From Those Over Whom It Is Exercised**. At the risk of overstating the very obvious, quality engineering professionals require management by consent if they are to give their best. It must be acknowledged by SOS or FOS development project leadership that the consent is the engineering professional's to give or withhold.
2. **Engineering Professionals Have Both The Right And The Duty/Responsibility To Sign Their Work.** Subsidiarity requires that people take responsibility for their decisions by signing their work (e.g., CI's, document artifacts) both literally and metaphorically. This is a key aspect of per-

formance in other professions (e.g., doctors, lawyers, professors). A signature on one's work may be the best single recipe for quality. For reasons of personal pride, as well as fear of recrimination, few will want to forever visibly sign their names to a *"dud"* product or deliverable.

3. **Autonomy Means Managing Empty Spaces.** In a federated SOS or FOS development program, groups and individuals live within two concentric circles of responsibility (note: engineers would understand this concept in terms of a limit cycle metaphor). The inner circle represents their minimally acceptable baseline — everything they have to do or risk failure. The larger circle marks the limits of their authority. The in-between area is their area of discretion (again, reference the type-II responsibilities under subsidiarity as previously discussed). This area is the space in which they have both the freedom and the responsibility to initiate action. Engineering professionals within a SOS or FOS project must fill this space — it is their *"type-II"* accountability. Implicit in this maxim is the notion that those higher up in the SOS or FOS program management structure may not know better in many cases. That assumption requires a lot of (warranted) trust and a necessary *"forgiveness"* when things turn out wrong. Where no mistakes are tolerated (conventional *"type-I"* project management ideology), no professional initiative will be risked. *"Forgiveness providing an individual/the group learns"* is a necessary part of federalist thinking in an engineering context. It is just as important to note that if someone can no longer be trusted, they cannot be given such an *"empty space"*. To keep the spirit of subsidiarity intact, those who do not merit trust must go / be removed to elsewhere — and quickly! This corrective mechanism is also advocated by Austin (1996). Summarizing, leaders will simultaneously need to be tough as well as trusting and forgiving. This is another paradox that exists in federated programs. Recall that success within a CAS is based on the underlying premise of the survival of the fittest (necessary inverse: extinction of the unfit!).
4. **Twin Hierarchies Are Both Necessary And Useful.** Twin hierarchies demonstrate federalism's (as well as the CAS) principle of interdependence at work. There is, in every organization, a clear status hierarchy. Some people are justifiably senior to others because of their proven knowledge, experience, and ability. However, in the engineering task hierarchy of a federated SOS or FOS program, the role dictates who is who (i.e., the junior CI leader may direct the proven senior test engineer). As skills become more specialized, successful federated ventures realize that there are temporary alliances of expertise (and hierarchies not based on status) that must be established in order to make best use of one another and, more over, get the job done in the spirit of avoiding the *"type-II"* errors previously discussed. Distinguishing between status and task hierarchies allows organizations to be much flatter without losing effectiveness. As a positive side effect, this approach allows the young specialists to demonstrate their expertise to the rest of the organization and provides great encouragement for making that expertise as good as it can be. Drucker (1999) clearly echoes these observations. Each of these side effects directly enhances the ability of a federated SOS or FOS program to reap the synergistic benefits attained through application of subsidiarity and interdependence, respectively.
5. **What Is Good For Me Should Be Good For The Organization.** This is the twin citizenship principle brought down to the level of the individual. Successful professionals believe the necessity of self-enlightenment. They know that if they do not continually invest in their own learning and development (i.e., adaptation), they will *"waste away"* as an asset. What they ask of federated SOS or FOS program management is that it facilitates and encourages the process of improvement. In turn, the professional owes a loyalty to the federated entity. Federated SOS or FOS program management must never take this loyalty for granted; it must be carefully nurtured without incident (lest the professional feel released from any sense of obligation). Both Handy (1992) and Drucker (1999) speak in complete unison on this requirement.

Thus, we indeed see that federalism is not just a classy word for restructuring conventional systems engineering management lines of authority. The ecology-based thinking behind it is simply that: autonomy releases energy; energy fuels innovation which is required for survival in adaptive environments; people have the right to do things their own way as long as it is in the common interest of the project; people need to be well-informed, well-intentioned, and well-educated to interpret that common interest, and; individuals prefer to be led rather than centrally managed.

Summarizing our discussions on federalism to this point, there are at least three paradoxes of power and control associated with these observations. The first paradox is that FOS development-related organizations, such as virtual corporations and alliances, need to be both big and small at the same time. Federalism responds to the pressures of this paradox by balancing power among:

- Those in the center of the organization;
- Those in the centers of expertise; and
- Those in the center of the action, such as the line engineers and developers.

The second SOS or FOS paradox lies in declared preferences for the equivalent to *"free and open markets",* which necessarily requires some level of competition among the members of the federation, as the best guarantee of efficiency, even as its managers instinctively organize their own operations for centralized control. The centralized *"engineering management shop"* context has a negative added value. To state it another way, the transaction costs of central planning and control exceed the contribution that they undoubtedly make. However, Handy (1992) observes that open markets, on their own, do not work any better than central planning. A bit of both is needed, and this circumscribes the federalist compromise.

The third paradox is best summed up by the phrase: *"what you do not own you cannot command"*. More directly, the third paradox is the desire by engineering leadership to run a SOS or FOS program as if it were yours when in fact you cannot afford, or may not want, to make it yours. This leads to the notion of alliances, which are notoriously difficult to manage or lead. Each alliance is unique, to be *"lived with"* rather than managed, better built on mutual respect and shared interests than on administrative or legal documents and tight controls. In these circumstances, power has to be shared, autonomy granted (i.e., in the hands of the *"adaptive agents"* ←→ *"the engineers"*, or at least the agent aggregate), and the *"marriage"* held together by trust and common goals, two of federalism's chief ingredients.

These paradoxes must trigger consideration of traditional management changes in SOS or FOS programs. Handy (1992) notes that at the same time that these paradoxes are triggering consideration of such federalized organizational structure changes, another force is pushing SOS or FOS projects towards federalism. That is, the *"pull of the professionals"*. Once again, this point is almost identically raised by (Drucker, 1999). SOS and FOS managers soon realize that their people are their chief assets. Often this realization becomes apparent only when a project is in trouble or being dismantled due to problems. For example, while a project disaster may be viewed by its sponsors as being directly due to the synoptic development team, its individual workers are often more highly valued separately, such as through migration to another project, than in the ensemble of the project being dismantled.

Engineering professionals prefer small, autonomous work groups based on reciprocal trust between leadership and those being led, groups that are responsible, as far as possible, for their own destiny. They would, of course, like to have it both ways, preferring autonomous groups to be part of a larger organization that can provide resources, career opportunities, and the leverage that comes with size. Thus, federalism applied to SOS or FOS engineering management is, to these engineering professionals, a way to make it big while keeping it small and independent.

Once these five key principles of federalism have been deep-rooted within an engineering organization, there are doubtlessly additional guidelines that must be bought into play. Handy (1992, 1995) suggests that the necessary principles of federalism will never be ingrained into an engineering organization unless the five leadership maxims are demonstrated as being standard operating procedure by organizational leadership.

## IMPLICATIONS OF THE SOS, FOS, AND FEDERALISM CONCEPTS FOR SYSTEMS ENGINEERING AND MANAGEMENT

Despite evidence that would suggest alternative plans, it is not atypical for contemporary organizations to treat the engineering of SOS or FOS with protocols that are, at best, just suitable for monolithic systems. The archetype of such ill-advised protocols is the DoD *"grand design"* life cycle that is based on the waterfall model[2] described in (Royce, 1970) that came into prominence circa-1970. A large number of problems have been encountered with *"grand design"* efforts to engineer a system. Thus, there have been a number of efforts to extend developmental approaches beyond the classic waterfall approach. Today, the classic waterfall approach is suggested only in those rare cases where user and system level requirements are crystal clear and unlikely to change, and where funding for the grand design is essentially guaranteed. This is rarely the case for major systems, especially those that are software intensive. This would be the rarest of all cases for a SOS or a FOS. Changing user needs and changing technology virtually guarantees that major systems cannot be developed using the grand design approach. Some of the difficulties are as follows.

- There is negligible participation of the full set of stakeholders to the tasks associated with engineering the system. Thus, the needed transdisciplinarity (Sommerville & Rapport, 2000) across disciplines is absent.
- There is limited attention devoted to making requirements consistent, understandable, and testable.
- No, or very inadequate, trade-off studies are conducted in order to understand and ideally to reduce the hazards of high-risk system requirements.
- Systems engineering processes and methods are selected indirectly and abductively by default through use of familiar methods and tools, such as by choosing a systems engineering (CASE) tool, instead of by first engineering the systems engineering process itself based on the nature of the application, and then selecting the methods and tools that best implement the chosen process. Thus, there is a failure to engineer an appropriate process for evolution of the system.
- There is insufficient modeling and simulation of the system architecture to verify that the architecture will support system requirements, especially those that relate to such deployment needs as fault tolerance, reliability, safety, and usability.
- Interoperability with external systems and compliance with existing architectures and frameworks, such as the Federal Enterprise Architecture Framework [F-EAF][3], is not considered to the extent that it should be in choice of the system architecture.
- There is a failure to properly plan for integration of a new system with existing legacy systems that the new system must support and communicate with.
- Little or no attention is paid to the fact that system evolution and emergence over time will necessarily be constrained, in terms of both cost and effectiveness, by the characteristics of earlier versions of the system.

As a follow on to the last point, it appears that evolutionary development of system architecture that considers the needs of emerging SOS and FOS will maintain the potential to greatly decrease the risk and costs of excessive rework of systems after they are initially engineered.

Two leading alternative approaches to the *"grand design"* approach for the engineering of systems were initially termed incremental and evolutionary, although the term evolutionary is now generally used to characterize both of these. Incremental development has as a plan to deliver the system in pre-planned phases or increments, in which each delivered module is functionally useful. In such an approach, the

---

[2] While Royce's waterfall model does in fact explicitly allow for cyclic iteration, it continues to be interpreted by engineering practitioners as being a pure serial, that is to say "grand design", process.

[3] Much additional information about the Federal Enterprise Architecture Framework may be found at http://www.itpolicy.gsa.gov/mke/archplus/cmodel.htm.

overall system capability improves with the addition of successive modules. In such an approach, the desired system capability is planned to change from the beginning, as the result of *"Build N"* being augmented and enhanced through the phased increment of *"Build N+1"*. This approach enables a well-functioning implementation to be delivered and fielded within a relatively short time and augmented through additional builds. This approach also allows time for system users to thoroughly implement and evaluate an initial system with limited functionality compared to the ultimately desired system. Generally, the notion of preplanning of future builds is strong in incremental development. As experience with the system at build N is gained, requirements changes for module N+1 may be more easily incorporated into this, and subsequent, builds.

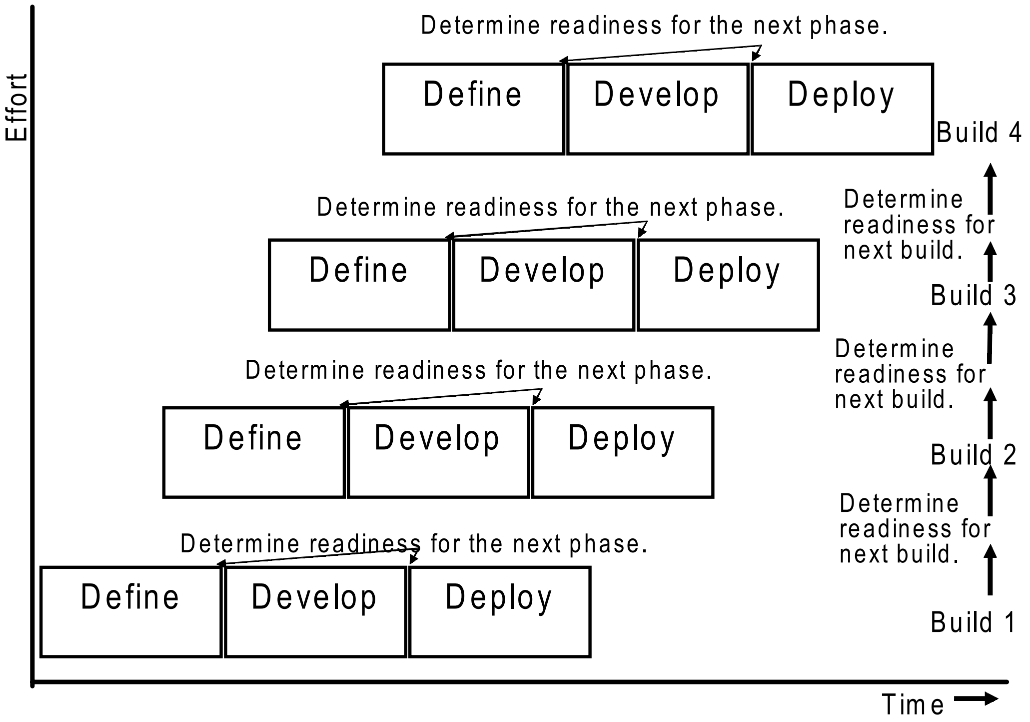


Fig. 1. Iterative lifecycles in evolutionary acquisition.

Evolutionary lifecycle development is similar in approach to its incremental complement; however, future changes are not necessarily pre-planned. In this approach, we recognize that we are unable to initially predict and set forth engineering plans for the exact nature of these changes. The system is engineered at Build N+1 through reengineering the system that existed at build N. In this approach, a new functional system is delivered at each build, rather than obtaining Build N+1 from Build N by adding a new module. The enhancements to be made to obtain a future system are not determined in advance, as in the case of incremental builds. Evolutionary development approaches can be very effective in cases where user requirements are expected to shift dramatically over time, and where emerging and innovative technologies allow for major future improvements. It is especially useful for the engineering of unprecedented systems that involve substantial risk and allows potentially enhanced risk management. Evolutionary development may help program managers adjust to changing requirements and funding priority shifts over time since new functionality introductions can be advanced or delayed in time in order to accommodate user requirements and funding changes. Open, flexible, and adaptable system architecture is central to the notion of evolutionary development. Thus, it appears that evolutionary development of a system architecture has the potential to greatly decrease the risk and costs of excessive rework of a SOS or a FOS after it has been initially engineered. Figure 1 indicates the general nature of the evolutionary lifecycle. This can be represented as a continuing waterfall, with feedback across the lifecycle, or as a spiral. Much of what has come to be known as evolutionary acquisition is based upon an equivalent spiral lifecycle.

There are a number of follow-on evolutionary acquisition efforts. Evolutionary acquisition strategies define, develop, and deploy an initial, militarily useful capability and a plan for subsequent definition, development, test and production/deployment of increments beyond the initial capability over time. The scope, performance capabilities, and timing of subsequent increments are based on continuous communications among the requirements, acquisition, intelligence, logistics, and budget communities. **A reasonable definition of evolutionary acquisition is that "evolutionary acquisition is a strategy for use when it is anticipated that achieving the desired overall capability will require the system to evolve, emerge, and adapt to changing user requirements and technologies during definition, development, or deployment."** This definition provides a linkage between the concepts of evolutionary acquisition and complex adaptive systems through use of the terms "emerge" and "adapt". Although the notion of simulation-based acquisition is not directly associated with this definition, it is clear that modeling and simulation are invaluable tools to use in any program and program management effort where there is continued change and the need to adapt.

The Department of Defense has not been unmindful of these needs and the need for evolutionary and incremental lifecycles was recognized a decade ago and made a part of the DoD 498 standard, which is no longer operational due to the decision to use commercial standards whenever feasible. Acquisition reform is a major effort now, and has been for much of the past decade. In the effort to reduce acquisition response time, the rewrite of DoD 5000 series regulations (DOD 2000) calls for evolutionary acquisition to be the preferred method for future defense acquisition programs. Unfortunately, there is often considerable mystification over the meaning of the term and lifecycle development methods that should be used for application of the various evolutionary acquisition approaches. Some of this mystification is evident in use of expressions such as evolutionary development, spiral development, spiral acquisition, evolutionary spiral development and a host of other expressions where the meanings are not well understood or accepted across the many stakeholders involved in these efforts.

The new approaches to defense systems acquisition are intended to support:

- Multiple process pathways, rather than a single rigid process, for system acquisition,
- Concentration on needed technology development and risk reduction efforts prior to acquisition program commitment,
- Timing of funding commitment and program initiation that depends upon the maturity of technology development and the involved in acquisition, and
- Use of evolutionary acquisition approaches as the preferred approach.

Only three potential milestone points are contained in the current Defense Acquisition System Documents, DoD 5000.1 and 5000.2 that relate, in large part, to evolutionary acquisition.

1. **Milestone A — Exploration.** At Milestone A, the Milestone Decision Authority (MDA) approves initiation of concept studies, designation of a concept development team, concept exploration exit criteria, and an Acquisition Decision Memorandum (ADM). Entrance into System Development and Demonstration is dependent on three things: technology (including software) maturity, validated requirements, and funding. Unless some other factor is overriding in its impact, the maturity of the technology will determine the path to be followed.
2. **Milestone B — System Development and Demonstration**. Prior to approving entry into System Development and Demonstration at Milestone B, the MDA considers such factors as threats and opportunities, technology assessments, early operational assessments or test and evaluation results. Programs that enter the acquisition process at Milestone B need a system architecture and an operational architecture for their relevant mission area. Transition into System Development and

Demonstration at Milestone B requires: full funding, a program manager (PM) in place, an approved operational requirements document ORD, and readiness to undertake system-level development.

3. **Milestone C — Commitment**. The purpose of this milestone is to authorize entry into low-rate initial production for systems that require this, into production or procurement for non-major systems that do not require low-rate production, or into limited deployment for a Major Automated Information Systems (MAIS) or a software-intensive system with no production components. Prior to making a Milestone C decision, the MDA considers such factors as independent cost estimates, component cost analysis and economic analysis for a MAIS, peoplepower estimates, training, and operational support. Following completion of a Full-Rate Production Decision Review by the MDA, the program enters Full-Rate Production, or procurement, and Deployment.

There are also a number of post-systems acquisition activities, the objectives of which include execution of a support program that meets operational support performance requirements to insure sustainability of systems in the most cost-effective manner for the anticipated life cycle of the system itself. When the system has reached the end of its useful life, it must be disposed of in an appropriate manner. The sustainability (sustainment) program includes all elements necessary to maintain the readiness and operational capability of deployed systems. The scope of support varies among programs but generally includes supply, maintenance, transportation, sustaining engineering, data management, configuration management, manpower, personnel, training, habitability, survivability, safety, occupational health, C4I, and environmental management functions. Follow-on operational test and evaluation programs evaluate operational effectiveness, survivability, suitability, interoperability, and deficiencies. These shall also be conducted, as appropriate. Figure 2 illustrates this milestone concept and the various components discussed here.

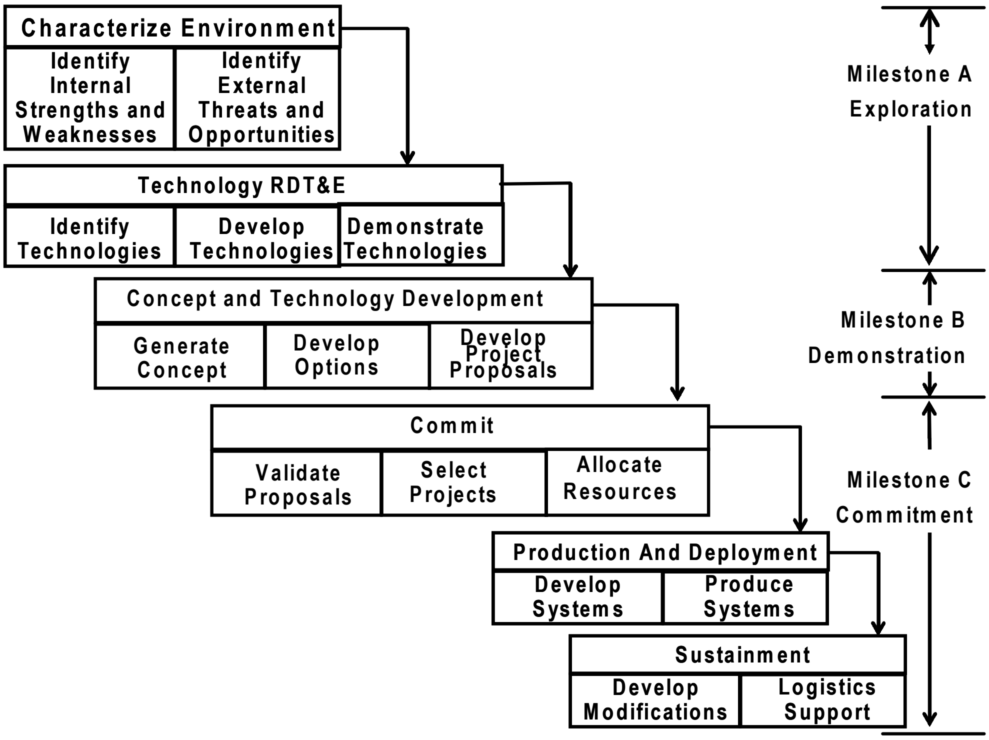
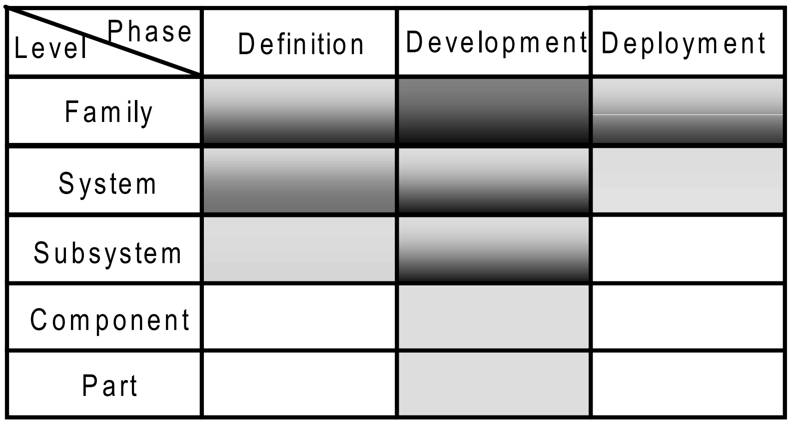


Fig. 2. Illustration of three milestones in defense system acquisition.

Evolutionary acquisition strategies define, develop, and produce/deploy an initial, militarily useful capability based on proven technology, time-phased requirements, projected threat assessments, and demonstrated manufacturing capabilities, and plan for subsequent development and production/deployment of increments beyond the initial capability over time. To facilitate evolutionary acquisition, Program Managers use appropriate enabling tools, including a modular open systems approach to ensure access to the latest technologies and products, and facilitate affordable and supportable modernization of fielded assets. Sustainability strategies evolve and are refined throughout the life cycle, particularly during development of subsequent blocks in an evolutionary strategy. Thus, we see the conceptual evolution of the evolutionary acquisition effort to include sustainability notions.

An excellent overview of evolutionary acquisition may be found in a Defense Systems Management College document (DSMC, 1998). There, it is simply indicated that evolutionary acquisition is a strategy for use when it is anticipated that achieving the desired overall capability will require the system to evolve during development, manufacturing, or deployment. This appropriate definition provides a suitable linkage between the concepts of evolutionary acquisition and complex adaptive systems through use of the term *"emergence"*.

When we consider the engineering of a system, we also often find ourselves also considering architectural views or perspectives. Many discussions of systems architectures focus on three primary architectural views: functional, physical and implementation. Development of the implementation or technical architecture is the process during which the entire design is integrated. This process also provides the raw materials for definition of the system's external and internal interfaces. Each of these activities in the design process is first completed at a high level of abstraction and correspondingly low level of detail. This results in an initial implementation or technical architecture for the system at a high level of abstraction. Then the entire process is repeated at a lower level of abstraction associated with greater detail for the next level of components. This repetition at lower and lower levels of abstraction, and greater and greater detail, is continued until ultimately the detailed implementation architecture is realized. The associated decisions and designs are reviewed and changes are implemented at the higher levels of abstraction to the extent needed (Sage & Lynch, 1998).



| Phase<br>Level | Definition | Development | Deployment |
|---|---|---|---|
| Family | ▓ | ▓ | ▓ |
| System | ▓ | ▓ | ░ |
| Subsystem | ░ | ▓ |  |
| Component |  | ░ |  |
| Part |  | ░ |  |

Fig. 3. Framework for activities by level and phase.

We have emphasized here that systems engineering is a multi-phase process. Each of these phases can be viewed at a number of levels: *"family of systems"* (i.e. SOS or FOS), system, subsystem, component, and part. These are generally defined in a functional block diagram structure for the system-based constructs being engineered. At each of these levels, the various phases of the systems engineering process need to be enabled through identification of appropriate various work efforts and a work breakdown structure (WBS) is one appropriate way to display this information. As can be inferred from the engi

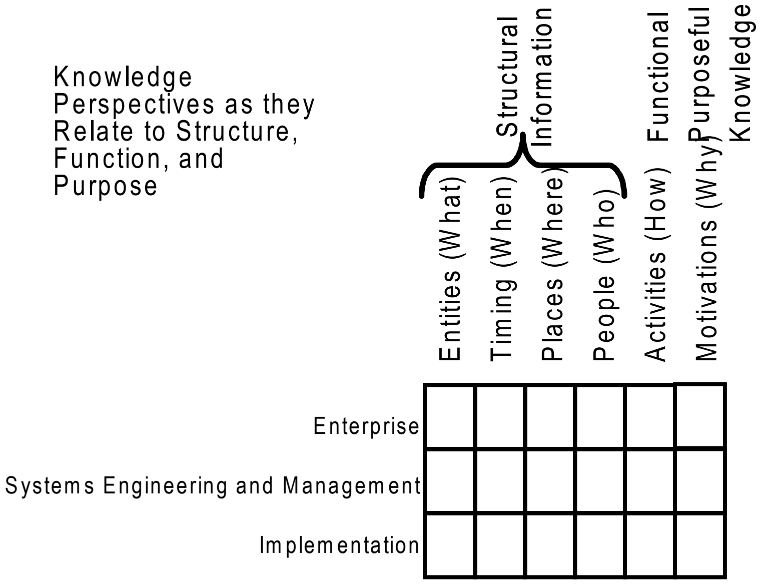


Fig. 4. Knowledge perspectives of acquisition stakeholders.

neering management principles of federalism, it may be the case that the delimiters of these processes and corresponding WBS descriptors are substantially different from one level to the next. We may identify a two-dimensional matrix framework representation of the phases and levels, such as shown in Figure 3. In this figure, the darker shades of activity cells are intended to represent greater intensity of effort and activity. While there is activity at the system and family of system (and enterprise level if we were to show this) across each of these phases, activity at the component and part levels are to be found primarily during portions of the development phase of effort. When we recall that this framework needs to extend across each of the three major systems engineering lifecycles, and that the family of systems may be comprised of a large number of *"systems"* as in SOS or FOS, the complexity of the effort to engineer these systems becomes apparent. This provides major encouragement for modeling and simulation as a part of the effort to successfully engineer a system.

Instead of thinking of the columns of this activity framework in terms of the phases of the systems engineering lifecycle, we might think of the questions that need be asked during the process of engineering a system. These include the information related questions that concern structure: what, where, when, and who. They also include knowledge related questions that concern function and purpose: how, and why. When we associate these questions used to fully define a factor, we obtain the representation shown in Figure 4. This representation may be recognized as an adaptation of the Zachman Framework for Enterprise Architecture (Zachman, 1987, 1997).

Relative to systems and program management, it is also important to address federating the engineering and acquisition cultures in order to be able to accept the broad sharing of tools, data, and responsibility/authority that are implicit in successfully engineering of entities up to and including SOS and FOS. This concept is encapsulated by Handy's 1st and 2nd principles of federalism. The need for cultural change focuses on the need to share data across different domains of the acquisition process, so that players focusing on different modalities of the same design are using the same, or consistent data. This is directly a restatement of Handy's 3rd principle of federalism. A fundamental objective is the appropriate and early involvement of all stakeholders in the acquisition process. This includes players involved in the training and maintenance of a system as well as other systems with which the primary system must interoperate. Figure 4 has illustrated a number of questions that should be answered by the major groups

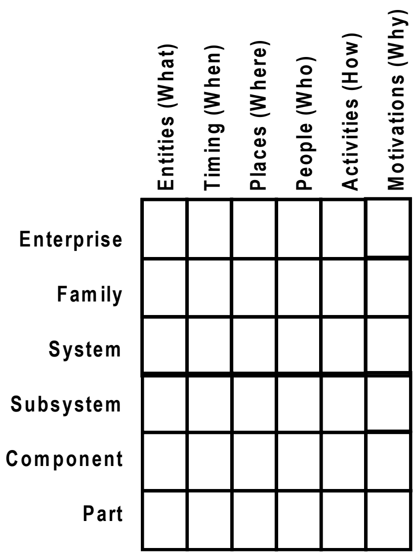


Fig. 5. Perspectives associated with system breakdown structure.

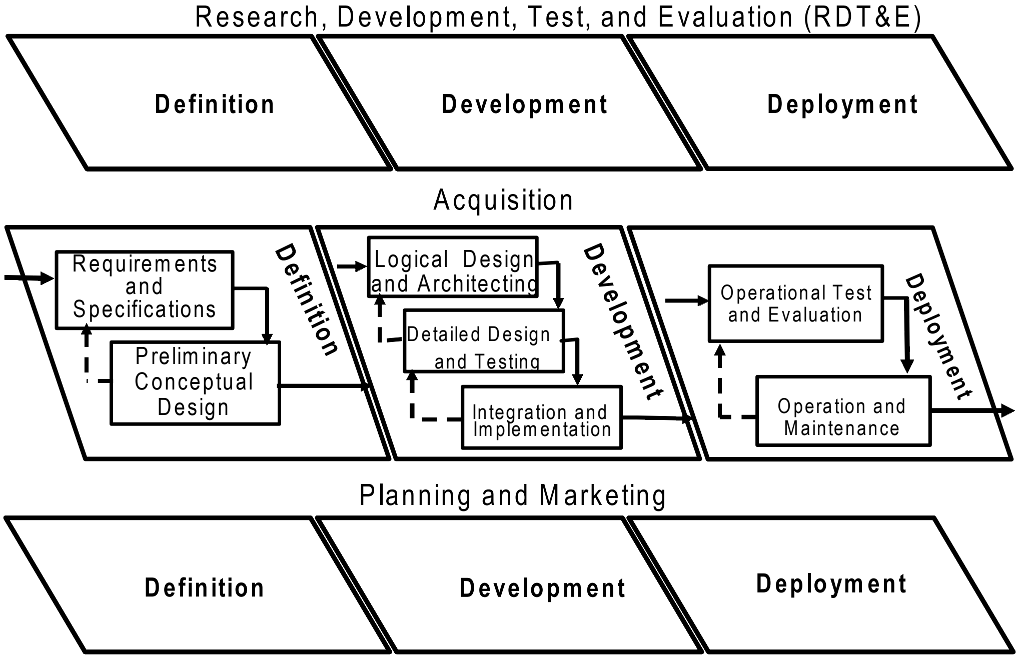


Fig. 6. Expansion of the individual build model of Figure 1 for greater realism.

concerned with engineering a system. Figure 3 represented activities that need to be associated with various phases of the effort at the level of elements in the system breakdown structure. We can also identify a framework comprised of the six interrogatories on the column dimension and elements in the system breakdown structure along the row dimension of a matrix, such as shown in Figure 5. Alternately, we could attempt a multidimensional representation of stakeholders, interrogatories, phases, and levels in the systems breakdown structure.

Descriptions of program management and acquisition strategy need necessarily to accommodate these evolutionary acquisition notions. Planning, and marketing, or the equivalent in DoD terminology, is one of the three major lifecycles associated with successful engineering a system. The intent of Figure 6 is to capture this notion of three major systems engineering lifecycles: planning and marketing, RDT&E, and acquisition (Sage, 1995; Sage & Rouse, 1999). In the DoD defense systems acquisition lifecycle, shown in Figure 2, milestone A is much concerned with planning, and with use of RDT&E to bring about mature technologies such that one can enter the commitment and evolutionary acquisition milestone with minimum risk and with knowledge of the "correct problem". Much associated with the notion of planning a program or project is that of managing the resulting program or project. The three essential phases associated with efforts associated with **Program Planning and Marketing** are:

1. **Definition**, interpreted more specifically as defining and organizing the program;
2. **Development**, interpreted more specifically as analysis and preparation of the program for actual set up and operation; and
3. **Deployment,** interpreted more specifically as explicit implementation of the program plan and associated tracking and management of the resulting operational program, including identification of activities to cease effort for the particular build in question and initiate a new build (or potentially close down the program).

There are a number of particular detailed activities that need necessarily to be associated with each of these phases. It may be helpful to describe some of these here. With respect to these phased efforts of definition, development, and deployment, we can identify several associated activities.

In **Program Definition**, we are specifically concerned with defining and organizing the program. We define Program Objectives, generally in the form of a Program Objective Statement (POS), to insure that the right program is being accomplished. We define Program Organization, such that all will know who is doing what, when, and why. Generally, this will include selection of a program team and a team leader. Define a Program Framework to describe how the program team will operate such that the program is being accomplished right. We also define a Program Definition Document (PDD) as a document that encompasses the definition and organization of the program in terms of objectives, organization, and framework. This will include such efforts as a Systems Engineering Management Plan (SEMP).

In **Program Development**, we are specifically concerned with analysis and preparation of the program for actual set up and operation. We analyze resource availability in terms of resource needs. We optimize tradeoffs across available and needed resources. We determine such program management documents as a Work Breakdown Structure (WBS), Cost Breakdown Structure (CBS), and Schedule Breakdown Structure (SBS), and a Risk Management Plan.

In **Program Deployment**, we are specifically concerned with implementation of the program plan. We implement the developed program management plan as an operational program for evolutionary acquisition. We identify the status of the evolving program over time and use this as a basis for configuration control and configuration management, including risk management as a part of configuration management. We establish a Configuration Control Board (CCB) to ensure successful adaptation, evolution and emergence of the program. Finally, we identify activities to cease effort for the particular build in question and initiate a new build or potentially close down the program if it appears that its useful life has ended. These activities, and potentially more are accomplished over time for each of the successive evolutionary builds. They generally need to be accomplished across the levels of the System Breakdown Structure, as identified by the rows in Figures 4 and 5.

In their work addressing SOS enterprise systems engineering, Carlock & Fenton (2001) identify three major roles of levels for systems engineering and management effort. At the top level, the enterprise wide SOS architecture and a strategic development plan are organized and maintained. At the middle level, the organization identifies processes that allow for trade-offs to take place across alternative implementation solutions such as to implement options that are best for the entire SOS such as to result in the program planning and management activities just described in terms of appropriate baselines. At the third level, there are implementation processes that bring about the engineering of a SOS in accordance with these baselines. This is, of course, an overall process of definition, development, and deployment to bring about planning for a SOS. The three role levels also directly map to the separation of powers concepts identified in Handy's 4th principle of federalism.

These authors also identify three contractual options to bring about an integrated SOS: contract with a single prime SOS engineering and integration contractor, assign separate contracts for each component system of the SOS and also contract with an independent SOS integrator, and to assign separate contracts for the component systems of the SOS and have the user organization (the Government in this paper), act as the SOS integrator. They describe the relevant features of each approach and discuss some of the costs and benefits of each approach in terms of the cases studied by these authors. The associated concerns are becoming widely recognized as evidenced by the recent effort by (Hammer, 2001) which describes how cross-company business process streamlining can lead to reduced costs, more effective operations and enhanced product and process quality. A four stage effort is recommended, comprised of scoping, organizing, redesigning, and implementing.

In the Krygiel (1999) work, two relatively detailed case studies are discussed and studied: the National Imagery and Mapping Agency's Digital Production System, and the U.S. Army's Task Force XXI. The integration experiences and lessons learned are discussed, and there are important discussions concerning the SOS integration environment of the future. These case studies, as well as those conducted by Carlock & Fenton (2001) were more of the nature of a complicated system of systems, as contrasted with a FOS and the correspondingly greater autonomy, heterogeneity, and dispersion. Nine important lessons were learned in the Krygiel (1999) study of SOS integration.

1. Precursor activities to a successful integration include: SOS architecture definition; development and tests of individual systems that comprise the SOS; development and tests of internal and external interfaces associated with the individual systems of the SOS; independent certification of compliance with the SOS architecture.
2. Use early, incremental, and iterative integration to achieve a SOS.
3. Develop a testing strategy for integration of the SOS will require: an agreed-to plan and process for testing; and a set of activities, characteristic of the operational activities that support the mission the SOS to use in testing the actual component systems of the SOS.
4. One should plan for substantial difficulties and expect to consume significant time and resources in integrating all the component systems of a SOS.
5. The use of a single integration facility with an agreed upon internal infrastructure will substantially facilitate integration of a SOS.
6. The process for SOS integration is dependent upon adequate support. Required will be an on-site acquisition leader empowered to integrate the SOS, and a SOS team with sufficient resources and authority to develop and manage the component systems of the SOS.
7. Common processes and a common infrastructure are needed in the integration environment in order to manage a SOS integration effort successfully. These include: an Engineering Board with responsibility and authority for identification and resolution of SOS issues and discrepancies; establishment of processes for identification of SOS issues and their disposition, tracking, and resolution; automated support for tracking SOS operational requirements; configuration management and control of the baselines of the component systems of the SOS; a formal build, verification, and re-integration process for changes; a robust communications infrastructure; and an office automation environment that supports these processes and their management.
8. Daily planning and scheduling of resources for integration events, contingency plans and schedules, timely dissemination of information pertinent to integration events, and daily status meetings with results immediately are activities that promote effectiveness and efficiencies in integrating a SOS.

9. Prototyping a SOS is desirable and can provide early insight into operational requirements and SOS systems architecture.

While these lessons were primarily for integration of a SOS, they seem to be capable of reasonable extension to the integration of a FOS.

## SUMMARY

The observations and recommendations described in this paper, while coarse-grained, are nonetheless empirically supported by and supportable through multiple case studies, analogies, and metaphors. There is a need to reconsider our canonical approach to engineering and management of SOS and FOS, both of which are instances of CAS. Most poignantly, there is a need to treat the engineering structures supporting SOS or FOS as CAS. Federated systems engineering and management concepts are plausibly valid approaches to achieving success in the evolutionary fielding of complex architectures up to and including SOS and FOS. The combined application of federated system engineering principles and evolutionary acquisition life cycles are particularly well-suited to address the risk areas of these CAS. It may even be argued that such federalistic and evolutionary principles are necessary concepts.

Ideally, behavioral analyses through simulation would be carried out to further explore the issues and implications of such federated engineering organizations. Holland (1995) has suggested simulation analysis using genetic algorithms as valuable way of gaining deeper understanding of such organizational relationships. In Holland's approach, the classic prisoner's dilemma game would be extended to a multi-player federated environment where adaptive agents and their meta-structures would play iterated versions of the game. The prisoner's dilemma, cooperate vs. defect, would be faced by interacting agents within aggregates as well as between aggregates. By calibrating the rewards and fitness algorithms applied by the agents to address the *"type I"* vs. *"type II"* error rewards system required under the New Federalism (i.e., OK to make mistakes, not OK to not take advantages of opportunities with long-term benefits to the federation), some understanding of effective vs. ineffective interaction (i.e., *"breeding"*) strategies between federated agents would begin to be obtained.

Modeling, simulations and analyses might well be conducted in support of developing SOS and FOS in the IT-centric 21st Century within which we have all been thrust. Our canonical approaches to engineering and management are rapidly becoming anachronisms. Taking a cue from complex adaptive systems theory, we must seek to adapt and cope, abandoning myths of *"total control"*, to attain sustainable, ecologically-balanced, effectiveness in systems engineering and management.

## REFERENCES

Appleton, D.S. (1994). *PROBE: Principles of Business Engineering — A Management Guide for High-Involvement Change.* TALON Press.

Axelrod, R., & Cohen, M.D. (1999). *Harnessing Complexity: Organizational Implications of a Scientific Frontier.* New York: Free Press.

Carlock, P.G., Decker, S.C., & Fenton, R.E. (1999). Agency-Level Systems Engineering for 'Systems of Systems'. *Systems and Information Technology Review Journal.* Spring/Summer, pp. 99–109.

Carlock, P.G., & Fenton, R.E. (2001). Systems of Systems (SoS) Enterprise Systems Engineering for Information-Intensive Organizations. *Systems Engineering. 4*(4), pp. 242–261.

Cook, S.C. (2001). On the Acquisition of Systems of Systems. *Proceedings of the 2001 INCOSE International Symposium, Melbourne AU.*

Czerwinski, T. (Ed.) (1998). *Coping with the Bounds: Speculations on Nonlinearity in Military Affairs.* Washington DC: National Defense University Press.

Department of Defense (1996). *Joint Vision 2010, American's Military — Preparing for Tomorrow.* Office of the Chairman, Joint Chiefs of Staff, Washington DC. Available at http://www.dtic.mil/jv2020/history.htm.

Department of Defense (2000). *Joint Vision 2020, American's Military — Preparing for Tomorrow.* Office of the Chairman, Joint Chiefs of Staff, Washington DC, 2000. Available at http://www.dtic.mil/jv2020/jvpub2.htm.

Department of Defense (2000). *Directive Number 5000.1 — The Defense Acquisition System*, *Instruction Number 5000.2 — Operation of the Defense Acquisition System*, October 23, 2000. Available at http://ric.crane.navy.mil.

Defense Systems Management College (1998). *Joint Logistics Commanders Guidance for Use of EVOLUTIONARY ACQUISITION STRATEGY to Acquire Weapons Systems.* DSMC Press, Fort Belvoir VA, 1998. Available at http://www.dsmc.dsm.mil/pubs/gdbks/jlc_evol_acqstrat.htm.

Drucker, P.F. (1999). Beyond the Information Revolution. *The Atlantic Monthly.* October 1999. Available at URL: http://www.theatlantic.com/issues/99oct/9910drucker.htm.

Gompert, D.C., & Isaacson, J.A. (1999). Planning a Ballistic Missile Defense System of Systems. Santa Monica CA: Rand Institute Issue Paper IP-18.

Hammond, M. (2001). The Superefficient Company. *Harvard Business Review. 79*(8), September, pp. 82–91.

Handy, C. (1992). Balancing Corporate Power: A New Federalist Paper. *Harvard Business Review. 70*(6), November–December.

Handy, C. (1995). Trust and the Virtual Organization. *Harvard Business Review.* May/June, pp. 8–15.

Holland, J.H. (1995). *Hidden Order — How Adaptation Builds Complexity*. Reading MA: Addison Wesley.

Holland, J.H. (1998). *Emergence — From Chaos to Order*. Reading MA: Addison Wesley.

Kelly, S., & Allison, M.A. (1999). *The Complexity Advantage: How the Science of Complexity Can Help Your Business Achieve Peak Performance.* New York: McGraw Hill.

Krygiel, A.J. (1999). *Behind the Wizard's Curtain: An Integration Environment for a System of Systems*. Vienna VA: CCRP Publication Series.

Maier, M.W. (1998). Architecting Principles for Systems-of-Systems. *Systems Engineering. 1*(4), pp. 267–284.

Manthorpe, Jr., W.H.J. (1996). The Emerging Joint System of Systems: A Systems Engineering Challenge and Opportunity for APL. *Johns Hopkins APL Technical Digest. 17*(3), pp. 305–310.

Owens, W.A. (1995). The Emerging U.S. System of Systems. In S. Johnson, & M. Libicki (Eds.), *Dominant Battlespace Knowledge* (pp. 1–14) Washington DC: National Defense University Press.

Roe, C.L. (1999). A Systems Engineering Profess for a System of Systems. *Proceedings 9th Annual International INCOSE Symposium*, Brighton UK, June, pp. 27–33.

Royce, W. (1970). Managing the Development of Large Scale Software Systems. *IEEE WESCON.*

Sage, A.P. (1998). Towards Systems Ecology. *IEEE Computer. 31*(2), February, pp. 107–110.

Sage, A.P. (1992). *Systems Engineering.* New York: John Wiley & Sons.

Sage, A.P. (1995). *Systems Management for Information Technology and Software Engineering.* New York: John Wiley & Sons.

Sage, A.P., & Lynch, C.L. (1998). Systems Integration and Architecting: An Overview of Principles, Practices, and Perspectives. *Systems Engineering. 1*(3), November, pp. 176–227.

Sage, A.P., & Rouse, W.B. (Eds.) (1999). *Handbook of Systems Engineering and Management.* New York: John Wiley and Sons.

Somerville, M.A., & Rapport, D. (Eds.) (2000). *Transdisciplinarity: Recreating Integrated Knowledge.* Oxford UK: EOLSS Publishers Ltd.

Stacey, R.D., Griffin, D., & Shaw, P. (2000). *Complexity and Management: Fad or Radical Challenge to Systems Thinking*. London: Routledge.

Wentz, L. (1998). *Lessons from Bosnia: The IFOR Experience.* Washington, DC: National Defense University Press.

Zachman, J.A. (1987). A Framework for Information Systems Architecture. *IBM Systems Journal. 26*(3), pp. 84–92.

Zachman, J.A. (1997). Enterprise Architecture: The Issue of the Century. *Database Programming and Design.* March, pp. 2–5.

Andrew P. Sage received the BSEE degree from the Citadel, the SMEE degree from MIT and the Ph.D. from Purdue, the latter in 1960. He received honorary Doctor of Engineering degrees from the University of Waterloo in 1987 and from Dalhousie University in 1997. He has been a faculty member at several universities and, in 1984 he became First American Bank Professor of Information Technology and Engineering at George Mason University and the first Dean of the School of Information Technology and Engineering. In May 1996, he was elected as *Founding Dean Emeritus* of the School and also was appointed a University Professor. He is an elected Fellow of the Institute of Electrical and Electronics Engineers, the American Association for the Advancement of Science, and the International Council on Systems Engineering. He is editor of the John Wiley textbook series on Systems Engineering and Management, the INCOSE Wiley journal *Systems Engineering* and is coeditor of *Information, Knowledge, and Systems Management.* In 1994 he received the Donald G. Fink Prize from the IEEE, and a Superior Public Service Award for his service on the CNA Corporation Board of Trustees from the US Secretary of the Navy. In 2000, he received the Simon Ramo Medal from the IEEE in recognition of his contributions to systems engineering and an IEEE Third Millennium Medal. His interests include systems engineering and management efforts in a variety of application areas including systems integration and architecting, reengineering, and industrial ecology and sustainable development.

Christopher D. Cuppan is The Scientific Advisor for Systems Engineering at the National Imagery and Mapping Agency (NIMA), and is currently on sabbatical at George Mason University's School of Information Technology and Engineering in order to complete his Ph.D. requirements and Dissertation. Mr. Cuppan has BS and MS degrees in Cartography and Computer Science from both the University of Wisconsin-Madison and The Ohio State University. Mr. Cuppan is also a semi-professional guitarist and luthier (URL http://hometown.aol.com/cuppancd/myhomepage/profile.html).