MASSACHUSETTS INSTITUTE OF TECHNOLOGY
1861

Engineering and Politics in Space Systems:

Developing a more integrated view

Assignment for ESD.83:
Research Seminar in Engineering Systems

Prepared by
Annalisa L. Weigel
December 2000

> *"Here is Edward Bear, coming downstairs now, bump, bump, bump, on the back of his head behind Christopher Robin. It is, as far as he knows, the only way of coming downstairs, but sometimes he feels that there really is another way, if only he could stop bumping for a moment and think of it…"*
>
> —A. A. Milne

Introduction

The origins of systems engineering in large-scale government technical endeavors can be traced back to the post-World War II development of military systems such as missiles. In the 1940s, Bell Laboratories was the first to use the term "systems engineering." In 1950, probably the first attempt to teach systems engineering was made at MIT by G. W. Gilman. [Brill, 1999]

If we look at the progress of systems engineering texts from 1960-2000, we see that all of them contained a common set of chapters. These were usually on the topics of identifying user needs, translating needs to requirements, the trade study process, system analysis methods, integration and test, and verification and validation. As systems engineering progressed into the 1970s, 1980s and 1990s, more chapters were added in the new books, such as design for manufacturing, design for reliability, design for maintainability, and so on. These are what Prof. Joel Moses at MIT refers to as "the –ilities." There are even chapters in systems engineering books on dealing with politics, policy and legal issues. But all these added chapters appear as "afterthoughts" to the systems engineering process. They are literally tacked onto the end of books, sometimes contributed by different authors than the rest of the book, and their discussion is not integrated into the core process that is presented in the book as systems engineering. This reflects that these –ilities, including policy, have not truly become integral parts of the systems engineering process yet. But perhaps there is hope.

Politico-Technical Systems and the Phases of Systems Thinking

This situation may be reflective of the stage of systems thinking we currently find ourselves in for the area of politico-technical systems. Politico-technical systems are those that the general public funds through taxation. The public expresses its value judgements on these systems almost solely through the political process. These politico-technical systems are further characterized by the public's sharply limited connection to the design, construction and operation of the system. [Rechtin and Maier, 1997, p. 76] Government space projects for exploration and defense run by the National Aeronautics and Space Administration (NASA) and the U.S. Department of Defense (DoD) are examples of politico-technical systems.

If the history of systems engineering thinking is broken down into phases [Mindell, 2000], we are currently at the end of the second phase and the beginning of the third phase in most fields of engineering systems. The first phase, which consists of recognizing that systems thinking is applicable to a problem, likely encompassed the years between World War I and World War II for politico-technical systems. An early example of this first phase of systems thinking is offered by Mindell [in Hughes, 2000, p. 27-56] as he traces an attempt by Ivan Getting to improve gunfire control through considering its parts as a system. The second phase, which consists of applying decomposition principles and analysis techniques to systems, could be considered to have started after World War II with the introduction of what came to be called systems engineering. The NASA Apollo lunar landing program in the 1960s is a classic example of the decomposition approach to systems thinking on a complex project.

The beginning of the third phase of systems thinking is now emerging. This phase thus far appears to be characterized by expanding the boundaries of the system under consideration. Users, infrastructure, control, regulation, and so on are now being included within the boundary of the system. Accompanying this expansion of system boundaries is the need to synthesize the multitude of new components and their relationships in some meaningful way that predicts or evaluates system behavior. A good example of this third phase of systems thinking is transportation systems. Previously, a transportation system was thought of as only a single vehicle – a car, a bus, an airplane. Now, a "transportation system" is thought to encompass multiple vehicles, and also the infrastructure it requires (roads, runways, airports), its control and regulation mechanisms (traffic lights, control towers, licensing), its fuel, its passengers, its pollution, and so forth. As this third stage progresses over the coming decades, perhaps the many systems engineering "afterthoughts" described earlier will become more integrated into the mainstream thinking and approach to conceiving, designing, implementing and operating politico-technical systems.

The Need for Integration

The focus of this paper is on integrating the technical aspects of system design with the political aspects of system design. The time is now perhaps right, as we begin this third phase of systems thinking, for this idea to really take hold and achieve broad implementation. Harvey Brooks made a call for action like this in a 1972 paper where he wrote, "If engineers are to bring systems thinking to bear on social problems, they must learn how to incorporate social and political theory into their analytical framework *ab initio*" [Brooks, 1972]. As a founding member of the International Institute of Applied Systems Analysis (IIASA) that very same year [Hughes and Hughes, 2000], Brooks was a strong proponent of applying systems thinking to socio-technical and politico-technical systems, especially in an international context. One wonders if, at that moment in time in 1972, Brooks really appreciated how challenging a task it would become to address socio-technical and politico-technical systems with a rigorous systems approach.

This notion of closely integrating political aspects and technical aspects is particularly important for politico-technical systems, as their name implies, which have heavy involvement with the political process. Previously, policy and politics were viewed as constraints on the politico-technical system design. Those constraints flowed one way – from the politicians down to the engineers. The problem with viewing policy as constraints is that constraints are typically considered to be static, whereas politics and policy are dynamic. Trying to solve a problem whose constraints are constantly changing is worse than trying to hit a moving target. Thus, a dynamic component to a project should be handled in a way that allows the program to successfully adapt to the changes when they occur. This misconstrual of political aspects as static has been the downfall of many projects. The Space Exploration Initiative (SEI), for example, was hard hit by a change in politics. In 1990, President Bush announced that the nation was embarking on a journey back to the moon and on to Mars. SEI barely made a dent in engineering plans before a change in Presidential policy cancelled the program entirely. President Clinton was elected in 1992 and opted for a cooperative foreign policy using the space station over a competitive foreign policy based on beating the Russians to Mars. The old program could not fit into this changed constraint.

Political Choices and Technical Choices Colliding

A more integrated view of technical and political aspects of system design is important because policy choices and technical choices are related. The example in Figure 1 illustrates this for a recently cancelled U.S. military program for space-based radar called Discoverer II (DII), jointly run by the U.S. Air Force, the National Reconnaissance Office (NRO) and the Defense Advanced Research Projects Agency (DARPA).

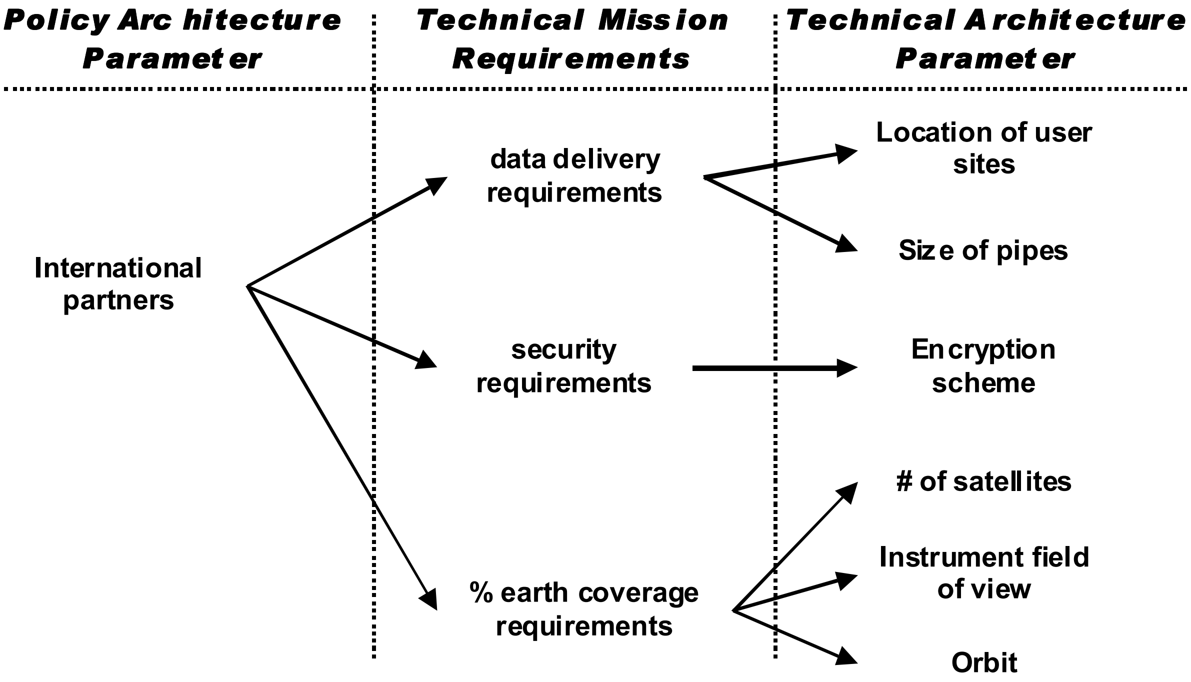

Figure 1: Discoverer II Example of the Impact of Political Choices on Technical Choices.

At one point in the DII program, policy makers decided that DII should be an allied program, not solely a U.S. program, and other nations should be invited to join. As the figure above shows, the impact of this policy choice results in changes to the technical mission requirements, which manifest themselves in various measurable technical architecture parameters. With international partners involved, the requirement of how much earth coverage to provide changes because, obviously, international partners are interested in coverage of their countries and not just U.S. territory. This results in changes to the number of satellites that will be needed for the program, the orbits those satellites will be placed in, and various instrument characteristics. Similarly, the desire to have international participation increases the security requirements placed on the system, resulting in complicated encryption schemes and software. Lastly, international participation dictates a change in data delivery requirements that show up in the size of the pipes and the number of ground sites necessary to receive downlinked data from the new orbits that provide the new fraction of earth coverage necessary.

But what is perhaps most important to take away from this example is that there is a feedback occurring here. The policy choices impact technical requirements. Technical requirements impact technical parameters, and technical parameters impact cost, schedule, performance and risk of a program. It is these things – cost, schedule, performance and risk – that policy makers are most concerned about. And it is these things that they are affecting directly through their choice of policies. The problem is that, arguably, many policy makers don't fully understand the technical implications of their choices.

Toward a New Way of Doing Business

It is time to consider politics and policy as an integral part of any politico-technical system design process. But why has such little progress been made on this to date? The quote from Milne at the beginning of this paper may illustrate the reasons. Most of the people who have paid much attention to politico-technical system design have been its practitioners in industry and government. Their schedules are already overburdened with the tasks absolutely necessary to get their system delivered. They report having little to no time to step back and contemplate their process and how they might do it differently, let alone better. In fact, the interview with a researcher is often a welcome break from their barrage of firefights on near-term critical items that normally occupy their day [Weigel, 2000]. At least it is so for those that can devote an hour of their day to speaking with an outsider.

Only recently have MIT and other academic institutions begun to examine issues relating to system design of large-scale politico-technical engineered systems. This paper presents the beginnings of a way to frame the political and technical issues involved in politico-technical system design. First, a model of the domains that interact with the political and technical aspects of system design is proposed, along with some terminology. This model is by no means fully complete or exhaustive, and is still being refined. The paper then explores various "lenses" that might become the basis for a doctoral research project examining how to better understand the integration of the political and technical aspects of system design.

The Domains and Their Interactions

A conceptual framework of the environment in which political aspects and technical aspects of politico-technical systems design interact is useful for this discussion. No such explicit framework has been found in the literature, so an original concept has been created. (Hereafter, all references to systems, systems design, systems engineering, systems architecture, etc. refer to politico-technical systems unless otherwise noted.)

The politico-technical environment can be considered to have three layers or domains in which organizations and groups work to bring forth a system design. These are:

- Technical Domain
- Operational Domain
- Political Domain

They are represented graphically in Figure 2. Each domain utilizes a particular process and produces a particular outcome from that process. In the following sections, each domain will be discussed, including definitions of their processes and outcomes, what kind of work takes place in the domain, and what groups and organizations lie in each domain.

Domain	**Process**	**Output**
Technical	Systems Engineering	Physical System
Operational	System Architecting	System Architecture
Political	Politics	Policy Architecture

Figure 2: Model of Domains in Politico-Technical Systems Design.

Political Domain

The work of the political domain is accomplished through the process of politics, and the outcome of the political process is a policy architecture (as regards a politico-technical system). Politics and policy mean many things to many people. For the purposes of this research work, we will define them here as:

> *Politics*: "the properly constituted and legal mechanism by which the general public expresses its judgments on the value to it of the goods and services that it needs" [Rechtin & Maier, 1997, p. 197]
>
> *Policy Architecture*: a plan or course of action adopted by a government or organization designed to influence and determine decisions, actions, and other matters [Morris, 1969]

The definition of politics is very important. The public, as the ultimate buyer and client (though not end user) of politico-technical systems, needs to decide what systems to buy. Politics, as it is described here, is the mechanism through which that value judgement process occurs. The public expresses its values to its elected representatives who then work in the Congress to determine what kinds of systems to buy. In essence, the political domain represents the client or customer of politico-technical systems, and controls the purse strings for all such projects. This is an important and necessary role – the role of the customer – in system design.

The primary political domain actors are the Congress, the White House, the media and the general public. The action in the political domain formally takes place in Washington, DC in the offices and the hearing rooms of the Congressional buildings. Actors in the political domain are guided by the logic of politics. For elected officials, this focuses around benefits, jobs and revenue to their respective districts. The logic of politics is further built on personal experiences, relationships, negotiations and compromise. Proof in the logic of politics involves having the majority of votes, and little more. This is in marked contrast with the technical domain, which will be discussed in the next section. [Forman, in Rechtin and Maier, 1997]

Technical Domain

The work of the technical domain is accomplished through the process of systems engineering, and the outcome of the technical process is a finished and complete physical system. There are instances when the systems engineering effort does not reach completion, for example when a program gets cancelled before the physical system is finished being designed and built. But when permitted to carry through to the end of the systems engineering process, the outcome will be a finished and complete system. There is ongoing debate within the field of systems engineering on the correct definition of the systems engineering process as well as its outcome. For further definitions of systems engineering, see Sage, Blanchard & Fabrycky, Martin, and Rechtin & Maier. For the purposes of this research work, we will define them here as:

> *Systems Engineering*: The "identification and quantification of system goals, creation of alternative system design concepts, performance of design trades, selection and implementation of the best design, verification that the design is properly built and integrated, and post-implementation assessment of how well the system meets (or met) the goals." [Shishko, 1995, p. 4]
>
> *Physical System*: The physical instantiation of the system designed through the systems engineering process. "The system is a collection of things or elements which, working together, produce a result not achievable by the things alone" [Rechtin and Maier, 1997, p. 253]

The primary technical domain actors are the aerospace companies contracted to build systems (currently Boeing, Lockheed Martin, TRW, for example) and the government system program offices (SPOs) that oversee the technical system development. The individual people in the companies come from scientific and engineering backgrounds, and most are engaged in performing actual engineering. The exceptions would be the various levels of program managers necessary to coordinate the project, but most of them have worked their way up through the engineering ranks to become managers [Weigel, 2000]. The individual people in the SPOs are military officers or government employees, depending on whether the program is a DoD or NASA program, and are of varying backgrounds (not all are technically trained).

The action in the technical domain takes place at company facilities, where teams of engineers execute the quantitative, reductionist systems engineering process described above by Sishko. It also takes place at the SPOs, where periodic reviews and evaluations of the technical progress of the contractor are held. Actors in the technical domain are guided by the logic of engineering and science. The logic of engineering and science is built upon mathematics and physical principles. Engineers prove their points with equations and numbers, oftentimes these artifacts speak for themselves. This is in marked contrast with the political domain, which does not speak the language of math and physics and instead opts for rhetoric and negotiation.

Operational Domain

There is a fundamental communications gap between the technical and political domains. Without someone or some group to translate between the two, it is unlikely that an integration of political and technical aspects of a system would ever take place. This is where the "operational domain" comes in and plays the roles of translator and go-between for the political and technical domains.

The work of the operational domain is accomplished through the process of architecting, and the outcome of the architecting process is a system architecture. "System architecture" is even a newer concept to the aerospace world than systems engineering, and thus we can expect considerable debate on the precise meaning of the term. Rechtin and Maier authored a popular work in this area, and we turn to it for insights. For the purposes of this research work, we will define system architecting and system architecture as:

> *System Architecting*: "The process of creating and building architectures; those aspects of systems development most concerned with conceptualization, objective definition, and certification for use" [Rechtin and Maier, 1997, p. 251].
>
> *System Architecture*: "The structure – in terms of components, connections, and constraints – of a product, process, or element" [Rechtin and Maier, 1997, p. 251]

The primary actors in the operational domain are a more diverse set of groups than for the technical or political domain, as might be expected given their translation or "ambassadorial" role between the other two domains. They are acquisition offices, legislative liaison offices, systems architecture offices, long-range planning offices, local support contractors and lobbyists from companies and other advocacy groups. Sometimes the activities of the SPOs put them into the operational domain from time to time, for example, if they are called upon to testify to Congress. But the majority of SPO activity and their primary job responsibility still occurs in the technical domain.

The action in the operational domain seems to be centered in Washington, DC despite most of the technical domain work taking place in other locations around the country. All of the offices listed above are based in the Washington area with the exception of the SPOs. The work of the operational domain, creating system architectures, takes places within these offices, at visits to SPOs and system contractor sites, and in the Congressional buildings in meetings with staffers. The language of the operational domain is a mix of political and technical jargon, combined with the distinctive lexicon associated with procuring systems for the government and letting contracts. The logic of the operational domain revolves around user needs, legacy systems, budgets, risk, personalities, agency rivalries, and power. Actors in this domain seem to resemble the political actors more than the technical actors.

The Interaction of the Domains, or How Systems are Created

The operational domain is responsible for the interaction between the technical and the political domains. The operational domain reaches up and down to other two, and brings in the information necessary to do its system architecting job. We can liken its middle position to purgatory, sandwiched between heaven and hell. Determining which direction is up, however, we suppose largely depends on the organization to which you belong.

The operational domain also filters information to the political domain from the technical domain, and from the technical domain to the political domain. In this latter capacity, the operational domain acts like a mediator and translator. This translation function, while necessary given the organizational structure of the government and the nature of politico-technical systems, adds delays and time lags into the system. These delays foster misunderstandings between the political and technical domains, and may be the root of program failures.

The success of the operational domain in creating a system architceture should determine the overall success of the system, as a sound architecture usually results in a successful system [Crawley, 1999]. For success in politico-technical systems, information from both the technical side and the political side is critical. Thus, the degree to which the operational domain is able to extract accurate and needed information from the technical and political domains should indicate its potential for success.

Lenses for Learning and Insight

As described in the three-domain model above, the interactions in architecting a system can be very confusing. We might wish for a crystal ball to unravel the complexities of the technical and political interactions. Indeed, given the ambiguous and qualitative nature of politics and the explicit and quantitative nature of technical issues, we may come to the conclusion that a crystal ball is perhaps the only thing that can marry the two and truly provide any insight into this messy situation. But before we give up hope all together, there are several concepts from fields engaged in systems thinking that may shed some light on the issue. The following sections introduce five "lenses" for scrutinizing the political and technical interactions on system design. They briefly examine how each might contribute to a doctoral research project on political and technical interactions in space systems architecting.

Design Structure Matrices

The design structure matrix (DSM) concept was first introduced by Steward in 1981 (see references). "A DSM is a compact, matrix representation of a system. The matrix contains a list of all constituent subsystems/activities and the corresponding information exchange and dependency patterns" [MIT DSM Research Team, 2000]. Dependency or direction of information flow is indicated by marks in the matrix. Marks above the diagonal indicate feedback from a later activity to an earlier upstream activity, meaning the earlier activity has to be repeated because of new information. Marks below the diagonal indicate a forward flow of information to later activities. [MIT DSM Research Team, 2000]

Applying DSMs to the technical and political interactions of space system architecting could yield many different insights. A DSM provides a visual image of key relationships, information flow, patterns of interaction and feedback, and downstream impacts of decisions. For space systems architecting, a DSM could capture the dependencies of necessary political and technical information to create system architectures. A second DSM could capture the organizational dependencies necessary among the three domains modeled in the previous pages to achieve the information flow required in the first DSM. A third DSM could capture the "as-is" current state of organizational interaction in creating space system architectures. The comparison of the second (ideal) and the third (as-is) DSM will indicate the location of deficiencies in the organizational structure for system architecting. These organizational deficiencies indicate where technical and political information is likely not being exchanged, and thus the success of the system architecture may be compromised. If case studies on the root cause of previous space system architecture programmatic "failures" correlated with these organizational deficiencies identified in the DSM, that would be a very powerful finding. If it did not, that would be valuable information as well, and would lead us to reexamine the DSM to understand if key components were overlooked.

The DSM would also point out feedback loops in the current architecting process. When feedback loops cross organizational boundaries, that could indicate a potential weak link. A more robust organization for space system architecting could be achieved by reordering the sequence of activities in the architecting process to minimize the number of times a feedback loop crosses organizations.

As MIT is a leader in research on DSM techniques, there is a wealth of experience and tacit knowledge to draw from for this approach to understanding the technical and political aspects of space systems architecting. In addition, the DSM techniques can be combined with other techniques discussed below. Although DSM techniques are very powerful when used to understand dependencies in a system qualitatively, they do not provide a quantification of the impact of the dependencies. As an example, a DSM may show four main feedback loops occurring in a system, but it doesn't readily shed light on which of those feedback loops is costing the project the most money, is the most risky, or provides the most critical information to the system. Several researchers have expanded on DSM methods to evaluate such things (Browning, Whitney, Eppinger for example), but in reality this requires linking the DSM information to another analytical tool.

System Dynamics

System dynamics is a tool for analyzing complex systems. It provides a means for generating computer models that simulate real world systems. The approach of system dynamics incorporates the theories of nonlinear dynamics and feedback control, which result in a more sophisticated understanding of the impacts of changes in highly interconnected dependent systems than would be possible through human intuition.

The interactions between the three domains are interconnected and dependent, and would thus make for an interesting yet challenging candidate to simulate in a system dynamics model. One of the particular strengths of the systems dynamics application to space system architecting is its ability to capture delays and lags in the system which occur when information is filtered and translated through the operational domain. Since these delays may be critical to determining the degree of success of the architecture, the ability to model them is key. In addition, system dynamics models continuous-time processes rather than discrete processes, and as such can capture ongoing processes and simultaneous procedures that influence each other. Since there is activity going on in all domains all the time in the system design process, this is well-suited to a system dynamics approach.

A simplified feedback model of the organizational interactions between groups in the three domains could be created from the DSMs described above and from supplemental interviews with key players. The model could then be simulated and evaluated for equilibrium behavior and response to changing initial conditions. Since many of the model parameters would likely be based on assumptions, it will also be important to understand how those assumptions built into the model affect the simulation results. After the behavior of the model is well understood, scenario analysis can be performed to investigate specific hypotheses.

Both system dynamics and DSM approaches permit the investigation of feedback loops in a system. Without a coherent theory about how political and technical interactions work in space system architecting, a DSM approach can be used to help formulate a theory. This theory can then be formalized in a system dynamics feedback model, which would be a new and original extension of the DSM methodology. The major drawbacks of system dynamics as applied to investigating technical and political aspects of system design are in the lack of quantifiable causal relationships. In physics, gravity acts on an object in a widely accepted and quantifiable manner. In the interactions of politics and technology in system architecture creation, little research has been done on quantifying the relationships. Thus, most of the relationships used in the system dynamics model will be based on assumptions. And the overall quality of the model will depend directly on the quality of the assumptions.

Axiomatic Design

Nam Suh of MIT developed the concept of axiomatic design in his book *The Principles of Design* published in 1990 (see references). Axiomatic design focuses on the initial phase of the system design and stresses the separation of *what* (objectives) is to be achieved and *how* (means) it will be achieved. In axiomatic design terminology, the *what* is expressed as functional requirements (FRs) and the *how* is expressed as design parameters (DPs). The separation of the *what* from the *how* is achieved through Suh's independence axiom, which requires that the independence of the functional requirements be maintained. A design where the FRs are uncoupled from the DPs (in other words, there is a 1-to-1 correspondence between FRs and DPs) has satisfied the independence axiom. These uncoupled states are preferable because they permit easy changes to the FRs, because each FR only affects one DP. The principle of decoupling to eliminate dependencies also appeared in the DSM methodology.

Translating the principles of axiomatic design into the space system architecting process seems possible. The political domain, representing the public as the customer, would express the FRs (the objectives) of the system. The technical domain, charged with the systems engineering tasks, would generate DPs (the means) for achieving the FRs laid out by the customer. It then becomes the job of the operational domain to strive for independence between the FRs and DPs, which seems to fit well into the description of the operational domain tasks presented earlier in this paper. Achieving independence creates flexibility in the architecture, and minimizes the impact that changes in the FRs have on the architecture. Since the political environment is constantly changing and the political environment determines (and changes from time to time) the FRs of the architecture, working to minimize the impact of those changes before they even happen makes a lot of strategic sense. Of course this must be what Suh knew all along.

Practitioners of axiomatic design acknowledge that there are drawbacks. Arriving at an uncoupled or even decoupled design is a big challenge. Many times, the independence goals are not achievable at all, especially with very dependent and interrelated systems that don't reduce to neat packages. This may turn out to be the case with large and complex politico-technical systems. So while axiomatic design has been used successfully in traditional small-scale engineered systems, it remains to be seen how successfully it can be applied when political factors are a large driver in the process. An interesting experiment is being conducted in the Production System Design Lab at MIT, applying axiomatic design to the product development process. Results from this research are due out in March 2001 and may offer useful insights on applying axiomatic design to the space systems architecting process.

Ecological Resilience

Engineering of physical systems (such as space systems) has not traditionally adopted biological metaphors and approaches. However, there are a growing number of attempts to bring in new knowledge and methods into engineering sciences from the biological sciences. One area where the biological sciences and the engineering sciences have already met is in ecological engineering. In describing resilience of ecological systems, Holling [in Schulze, 1996, p. 32] writes:

> "Ecosystems do not have single equilibria with functions controlled to remain near them. Rather, destabilizing forces far from equilibria, multiple equilibria, and disappearance of equilibria define functionally different states, and movement between states maintains structure and diversity."

It is interesting to note how much the political domain is like this description of an ecosystem. Politics has many equilibria, and when the balance of power shifts, the political environment settles into a new equilibria. Politics, like ecosystems, is not truly controllable and the result of attempts to control it rarely produces the expected outcomes.

In applying some of the ecological resilience thinking, it seems the most valuable concept is that of multiple equilibria for a system architecture. If the system architecture were being designed to adaptively function well in many states instead of being optimized for a single state, the process of creating the architecture would surely be different. Interactions between the technical aspects and the political aspects would be different as well. It seems that this ultimately gets at the concept of system flexibility in being able to adapt to many different situations equally well. While ecological resilience offers a theory on why flexibility is important, it doesn't provide a method for quantitatively valuing the multiple equilibria or flexibility in the system. Real options theory, discussed next, does provide a framework for quantitatively evaluating the value of flexibility in engineered systems.

Real Options Theory

An option is defined as a right, not an obligation, to take a future action. This creates flexibility, and provides asymmetric returns for the option holder because the option only needs to be exercised when it is advantageous to do so. Options theory has flourished and is widely used in the financial field, and there is growing interest in applying it to engineering systems. Some (including analysts at RAND and de Neufville at MIT) believe that options theory will be the next big wave of engineering systems analysis. The flexibility to adapt quickly and easily to changing environments and changing requirements will separate the "winning" systems from the "losing" systems in today's dynamic world. This applies equally well when discussing politico-technical systems as when discussing commercial systems.

As we look to apply options thinking to the space system architecture process, we find its value primarily in creating flexibility in a process filled with time-lagged information. Given the delays in information transfer from one domain into another, the concept of keeping options open while waiting on the needed information to make a decision may have large payoffs.

However, a critical assumption in options analysis is that the option's risk can be associated with an exogenous market risk with a known statistical distribution. In other words, there is a history in that market area that can be presumed to continue. Applying options analysis to completely new projects may thus be problematic, for there is nothing in history to compare the market risk against. While space system architectures are rarely duplicated, there are some similarities between architectures that could establish a history. But are these enough?

Another sticky question in applying options to space system architectures is, "What is the equivalent of *market* risk for government projects?" While the literature offers no good insights, a suitable answer may be to combine the principles of options theory with decision analysis and utility analysis, where utility analysis is used to valuate the options. But even the principle of evaluating utility is difficult for the government, as profit and loss are not the bottom line. What is the utility of a space-based image of the Mississippi delta? That's a very hard question to answer, and it varies among all the stakeholders in the government. This leads to another potential difficulty in evaluating the utility preferences of a group of stakeholders when each stakeholder has a different utility preference. Utility theory essentially breaks down under such conditions.

Lastly, it is unclear how the notion of building in options into a system architecture can be accomplished without subjecting the system to the detrimental effect of requirements creep, or a gradual growth in the requirements a system must satisfy. Requirements creep usually results in cost and schedule overruns for a system. There may be a turning point in a system development effort after which retaining options is not worth the benefits they bring. The point of options analysis is to identify when an option is valuable and when it is not, and thus it can help answer questions like these.

Conclusions

This paper began with a look into the history of systems engineering and systems thinking, and found that we are on the verge of a new phase of learning in systems analysis that focuses on expansion of system boundaries. Thus, it seems like a good time to examine in greater depth the broadening of the systems development process itself to include the political as well as the technical aspects of politico-technical systems. A model of three interacting domains – political, technical and operational – described the people and work in each layer, and how they come together to create a space system architecture. A key part of the model is information: how it is exchanged, how it is used, how it relates to system architecture success, and how its delays are inherent in the design and construction of the domains.

The five lenses of DSM, system dynamics, axiomatic design, ecological resilience, and real options theory were compared and contrasted in their applicability for examining the political and technical interactions in the system architecting process for space systems. Each of these lenses would make for an interesting doctoral research project, but more exploratory work needs to be done before a final approach can be formulated. Regardless of the choice of lens and method, the timing is right to begin such a project.

Acknowledgements

I would like to acknowledge my doctoral committee, Prof. Daniel Hastings, Prof. Sheila Widnall, and Prof. Tom Allen, for their review of my research which has contributed to this paper. I would also like to acknowledge my colleagues in ESD.83 and Prof. Mindell for their lively dialog every week and the intellectual insights that have found their way into my research.

Bibliography and References

Blanchard, Benjamin S. and Wolter J. Fabrycky (1981). *Systems Engineering and Analysis*. Englewood Cliffs, NJ: Prentice-Hall, Inc.

Brill, James H. (1999). “Systems Engineering – A Retrospective View.” *Systems Engineering*, Vol. 1, No. 4. pp. 258-266.

Brooks, Harvey (1972) “A Framework for Science and Technology Policy” *IEEE Transactions on Man, Systems, Man and Cybernetics*. Vol. SMC-2, pp. 584-588.

Crawley, Edward F. (1999) “16.882J: Systems Architecture” Class notes, MIT graduate course.

Hughes, Agatha C. and Thomas P. Hughes, editors. (2000) *Systems, Experts, and Computers: The Systems Approach in Management and Engineering, World War II and After*. Cambridge, Massachusetts: MIT Press.

Martin, James N. (1997). *Systems Engineering Handbook: A Process for Developing Systems and Products*. Boca Raton: CRC Press.

Mindell, David A. (2000) “ESD.83: Research Seminar in Engineering Systems” Class notes, MIT graduate course.

MIT DSM Research Team (2000) http://web.mit.edu/dsm/DSM_def.htm Last accessed 12/13/00.

Morris, William, editor (1969) *American Heritage Dictionary of the English Language*. Boston: Houghton Mifflin Company

Rechtin, Eberhardt and Mark W. Maier. (1997). *The Art of Systems Architecting*. Boca Raton: CRC Press.

Sage, Andrew P. (1992). *Systems Engineering*. New York: Wiley.

Sage, Andrew P. and Charles L. Lynch (1998). “Systems Integration and Architecting: An Overview of Principles, Practices, and Perspectives.” *Systems Engineering*, 1(3). pp. 176-226.

Sastry, M. Anjali (1997) “Problems and Paradoxes in a Model of Punctuated Organizational Change.” *Administrative Science Quarterly*, Vol. 42. pp. 237-275.

Schulze, Peter C., editor (1996) *Engineering Within Ecological Constraints*. Washington, DC: National Academy Press.

Steward, Donald V. (1981) "The Design Structure System: A Method for Managing the Design of Complex Systems" *IEEE Transactions on Engineering Management*, Vol. 28, pp. 71-74.

Suh, Nam P. (1990) *The Principles of Design*. New York: Oxford University Press.

Trigeorgis, Lenos (1996). *Real Options*. Cambridge, MA: MIT Press.

Weigel, Annalisa L. (2000) *Understanding the Enterprise Value of Test: Characterizing System Test Discrepancies in the Spacecraft Industry*. MIT Lean Aerospace Initiative Monograph, Cambridge, MA

Integrated Development Environment. For most programming languages, an IDE can be used. An IDE provides a variety of tools for the programmer, and usually includesÂ Many organizations are moving away from developing a specific app for a mobile device and are instead making their websites more functional on mobile devices. Using a web-design framework called responsive design, a website can be made highly functional no matter what type of device is browsing it. With a responsive website, images resize themselves based on the size of the deviceâ€™s screen, and text flows and sizes itself properly for optimal viewing. You can find out more about responsive design here. Implementation Methodologies. Politics of the International Space Station have been affected by superpower rivalries, international treaties and funding arrangements. The Cold War was an early factor, overtaken in recent years by United States distrust of China. The station has an international crew, with the use of their time, and that of equipment on the station, being governed by treaties between participant nations. ROCKET-SPACE DEVICE ENGINEERING AND INFORMATION SYSTEMS Vol. 3, Iss. 2, 2016, pp. 66â€“71.Â Development of Microwave Monolithic Integrated Circuits of 5 mm Wavelength Range for Application in Promising Space Systems. Yu.V. Fedorov, P.P. Malâ€™tsev1, D.L. Gnatyuk2, O.S. Matveenko3, D.V. Krapukhin, S.A. Gamkrelidze4 1,4doctor of engineering science, professor, 2,3candidate of engineering science.Â Federal State Budgetary Scientific Establishment Institute of Super High-Frequency Semiconductor Electronics of the Russian Academy of Sciences (IUHFSE RAS) has developed a frameless MIC set of the 5 mm wavelength range (Fig. 1).Â Not more than. 57 64. 6.5 16. Probably more important was the significant leap in System Engineering (SE) execution that would be required to build and operate a multi-national space station. In a short period of time, NASA and its partners had to work out how to integrate culturally different SE approaches, designs, languages and operational perspectives on risk and safety.Â NASA basically developed a new version of spiral construction theory. Â§ NASA and its IPs had to develop innovative methods to test and verify interfaces.Â 6 Case Studies of Systems Engineering and Management in Systems Acquisition, George Friedman and Andrew Sage, Systems Engineering, Vol. 7, No. 1, 2004, Â© 2003 Wiley Periodicals, Inc.

The current practice of regional development requires the development of new theoretical and methodological approaches to their study. There is a wide variety of different integration forms in modern world. There is an overlapping membership of a large number of states in various regional structures, and a supplement of formal interstate interactions at the regional level with expanding stable informal and â€œprivateâ€ ties. Besides the formation of regional cooperation in various fields take place, and, finally, the intensification of direct relations between interstate unions of different regio Strong traditions, scientific schools, engineering and design teams emerged in the field of space research. Now Russia possesses a unique experience and advanced technologies for placing spacecraft into orbit and the implementation of long-term manned spaceflights.Â â€¢ a new generation of space and traffic engineering; â€¢ information technology, management, navigation systems; â€¢ electronic components and energy-efficient lighting devices; â€¢ nanodevices and microsystems engineering; â€¢ structural nanomaterials and functional nanomaterials.Â The innovation development programs have become more effective instruments for Climate engineering or climate intervention, commonly referred to as geoengineering, is the deliberate and large-scale intervention in the Earth's climate system. The most prominent subcategories of climate engineering are solar radiation management and carbon dioxide removal. Solar radiation management refers to offsetting the warming effect of greenhouse gases by reflecting more solar radiation (sunlight) back into space. Carbon dioxide removal refers to removing carbon dioxide gas ($CO2$) from the