2015 NDIA GROUND VEHICLE SYSTEMS ENGINEERING AND TECHNOLOGY SYMPOSIUM
SYSTEMS ENGINEERING (SE) TECHNICAL SESSION
AUGUST 4-6, 2015 - NOVI, MICHIGAN

Understanding Systems through Graph Theory and Dynamic Visualization

Troy Peterson
Technical Fellow
Chief Engineer
Booz Allen Hamilton
Troy, MI 48084

ABSTRACT

As today's Cyber Physical Systems (CPS) become more and more complex they provide both incredible opportunity and risk. In fact, rapidly growing complexity is a significant impediment to the successful development, integration, and innovation of systems. Over the years, methods to manage system complexity have taken many forms. Model Based Systems Engineering (MBSE) provides organizations a timely opportunity to address the complexities of Cyber Physical Systems. MBSE tools, languages and methods are having a very positive impact but are still in a formative stage and continue to evolve. Moreover, the Systems Modeling Language (SysML) has proven to be a significant enabler to advance MBSE methods given its flexibility and expressiveness. While the strengths of SysML provide clarity and consistency, unfortunately the number of people who know SysML well is relatively small. To bring the full power of MBSE to the larger community, system models represented in SysML can be rendered in a more intuitive form. More specifically, Graph Theory has proven to be very effective in the design, analysis, management, and integration of complex systems. Network Analysis and Design Structure Matrix, both variants of Graph Theory, enable users to model, visualize, and analyze the interactions among the entities of any system. Use of MBSE and Graph Theory together to create dynamic visualization can help teams gain insights, build intuition and ultimately help speed the innovation process.

INTRODUCTION

System complexity is rapidly growing as internal and external system interactions race to new heights. This has been driven predominately by the explosion of the Internet of Things (IoT). IoT is increasing system interactions across many traditional boundaries and enabling institutions, managers and systems to adapt more readily to contextual changes. The related intensity of interconnectedness and embedded intelligence has given rise to a new class of systems called Cyber-Physical Systems (CPS). As described by the National Science Foundation (NSF) Cyber-Physical Systems are "engineered systems that are built from, and depend upon, the seamless integration of computational algorithms and physical components". They are systems which tightly intertwine computational elements with physical entities within aerospace, automotive, energy, healthcare, manufacturing and other sectors.

The rapid increase in Cyber-Physical Systems is changing the way we develop, manage and interact with systems. While the potential benefits are clear, it also comes with many risks; some of which have not yet been fully exposed. These systems place significant demands on organizations to ensure rigor and trustworthiness of systems by improving safety, security and reliability. The NSF notes that CPS challenges and opportunities are both significant and far-reaching. To address these challenges the NSF is calling for methods to conceptualize and design for the deep interdependencies inherent in Cyber-Physical Systems.

While Cyber Physical Systems advance and transform the landscape, the Systems Engineering discipline is also experiencing a transformation to a model-based discipline. A necessary transition to handle the complexity and emergent behaviors exhibited by CPS. While Model Based Systems Engineering (MBSE) shows significant promise, it is still in a formative stage and very few subject matter experts understand the languages or have ready access to MBSE related tools. The following sections discuss key enablers to managing the complexity of Cyber-Physical Systems. More specifically (1) the role of MBSE to provide an explicit and integrated system model. (2) Expressing system models in a way that deepens our understanding and (3) the power of Graph Theory as a complementary means to reach the larger community of stakeholders in the development process. Together these enablers can help teams better understand system models to gain insights, build intuition and ultimately help speed the innovation process.

MODEL-BASED SYSTEMS ENGINEERING (MBSE)

Over the years, methods to reduce system complexity have taken many forms. Model Based Systems Engineering (MBSE) provides organizations a timely opportunity to address this complexity. INCOSE defines Model-Based Systems Engineering (MBSE) as "the formalized application of modeling to support system requirements, design, analysis, verification and validation activities beginning in the conceptual design phase and continuing throughout development and later life cycle phases…[1]" The Object Management Group's MBSE wiki notes that "Modeling has always been an important part of systems engineering to support functional, performance, and other types of engineering analysis.[2]"

The application of MBSE has increased dramatically in recent years and is becoming a standard practice. Enabled by the continued maturity of modeling languages such as SysML and significant advancements made by tools vendors, these advancements are improving communications and providing a foundation to integrate diverse system models. MBSE is often discussed as being composed of three fundamental elements – tool, language and method.

The first element, tools, are software packages used to manage a systems descriptive model and they cover the spectrum from open source, free options to tool suites that are very costly with proprietary methods and plug-ins. While there are differences between tools most are very capable and can be used to make significant improvements in managing the complexity of CPS.

Many modeling languages also exist to express system representations. MBSE practitioners often use the System Modeling Language (SysML). "The OMG systems Modeling Language (OMG SysML™) is a general-purpose graphical modeling language for specifying, analyzing, designing, and verifying complex systems that may include hardware, software, information, personnel, procedures, and facilities. In particular, the language provides graphical representations with a semantic foundation for modeling system requirements, behavior, structure, and parametrics, which is used to integrate with other engineering analysis models.[3]" SysML continues to mature providing a powerful way to graphically express the complexity of systems.

The third element, method, has not always been given proper consideration, because the language and tool are relatively method independent, it is methodology which further differentiates the effectiveness of any MBSE approach and its ability to help manage the complex and interrelated functionality of today's Cyber Physical Systems.

As a Model-Based Systems Engineering (MBSE) methodology, Pattern-Based Systems Engineering (PBSE) is tool and language neutral and offers a strong underlying ontology and metamodel. At the heart of its S*metamodel, shown in Figure 1, is a focus on interactions which are also at the heart of Cyber Physical Systems and the basis of the physical sciences.

PBSE can address 10:1 more complex systems with 10:1 reduction in modeling effort, using people from a 10:1 larger community than the "systems expert" group, producing more consistent and complete models sooner. These dramatic gains are possible because projects using PBSE get a "learning curve jumpstart" from an existing pattern and previous users, rapidly gaining the advantages of its content, and improving the pattern with what is learned, for future users.

Over several decades, PBSE has been developed and practiced across a range of domains, including carrier grade telecommunications, engines and power systems, automotive and off road heavy equipment, telecommunications, military and aerospace, medical devices, pharmaceutical manufacturing, consumer products, and advanced manufacturing systems[4,5,6].

PBSE will be leveraged as the MBSE methodology due to the impact it has had in helping teams focus on interactions, improve platform management as well as its data compression characteristics and strong underlying metamodel. To increase awareness of the PBSE approach, INCOSE has recently started a Patterns Challenge Team within the INCOSE MBSE Initiative[7]. More detail on PBSE can be found at: http://www.omgwiki.org/MBSE/doku.php?id=mbse:patterns:patterns

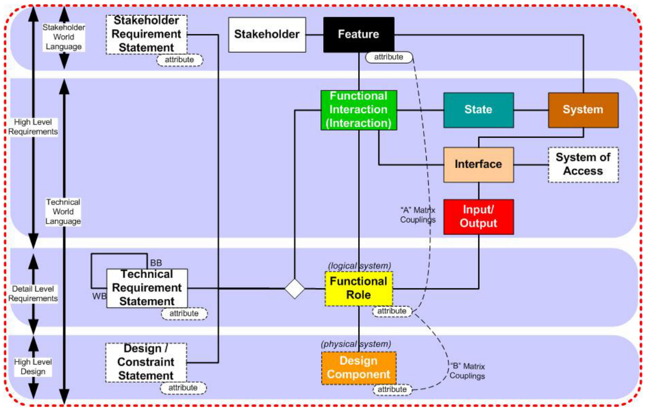


Figure 1: A summary view of the S* metamodel

While MBSE is still in a formative stage and continues to evolve, the Systems Modeling Language (SysML) has proven to be a significant enabler to advance MBSE methods given its flexibility and expressiveness. The flexibility of the language and advances in tools also permits easy construction of allocation tables and dynamic tabular representations. While these strengths provide clarity and consistency, unfortunately, the number of people who know and SysML well is still relatively small. This has led to some criticism and limited widespread acceptance.

LEARNING & MODEL EXPRESSION

To bring the full power of MBSE to the larger community system models in SysML can also be represented in a more intuitive form. Not as a replacement to the rich detail provided by the SysML but as a complementary product to conceptualize and design for the deep inter-dependencies inherent in Cyber-Physical Systems.

To ensure we can extend the power of MBSE and SysML to a much larger community it's worthwhile to first consider how we think and interact with information. The objective is to maximize our ability to translate system data and information into knowledge we can use to improve the trajectory of our programs. To achieve this we need to deepen the understanding of systems models for the larger community of development stakeholders.

"Vision trumps all other senses. We are incredible at remembering pictures." [8] When we hear a piece of information, we often only recall 10% of it three days later. When a picture is used 65% is recalled. Hands down pictures beat text so the richer we can make our visualizations the more rapidly we can learn and begin to understand complex systems.

Research has shown repeatedly that we are wonderful at encoding images but not especially good with arbitrary information or long strings of numbers / information. Our brains are designed for spatial information and our image recognition is very durable. A string of numbers can be very difficult to remember – however if put into a spatial format, almost perfect recall becomes easy. While it may be identical information, a spatial view is much easier to remember simply due to how our brains are wired. In fact, if information is encoded spatially our recall time is also improved dramatically. [9]

Just as a computer needs information coded properly in bits – we need images. Of course, long strings of numbers are very easy for a computer to recall but spatial information and associations can be far more challenging for a computer. This is true even with the great strides made in machine learning and artificial intelligence. As our developments become more digital we need to appropriately allocate activities. Let machines handle what they do well, for instance, storing and reproducing information and focus our attention on leveraging our amazing cognitive ability to compare, contrast, associate, integrate and synthesize information.

Use of different representations of the same information is not a new concept. Architects, engineers and others often provide multiple views of the same elements to provide users of their products as much clarity as possible. To represent form engineers often use left, right, top, bottom, exploded, isometric, cut-out and perspective drawings. To represent function we also use several views; this notion is a key tenant of SysML, UML, DODAF and other languages and frameworks. In particular, SysML defines nine diagrams, all of which are useful and depict important information about the structure and behavior of a system. However, there are many other views we can also use to improve clarity. The number and diversity of views also correlates to the number and diversity of stakeholders who have an interest in better understanding the modeled system. Many SMEs are required to engineer systems. Each SME has tools, languages and methods that they use to model and design systems. These languages are often not natural or intuitive to others outside their domain. We need to have representations which bridge over roles, domains and areas of functional expertise.

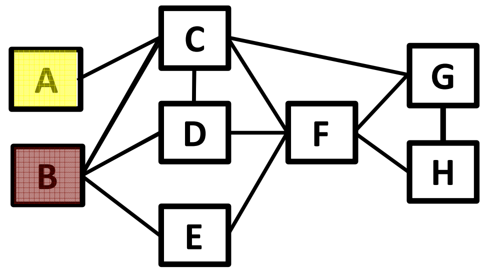
Graph Theory can provide a means to reach this larger community without significantly sacrificing the power and expressivity of SysML's semantics. It can exposes us to new ways of viewing, analyzing and understanding the complex systems we design.

GRAPH THEORY & DYNAMIC VISUALIZATION

The application of graph theory has proven very effective in the design, analysis, management, and integration of complex systems. Graph Theory provides a simple yet powerful means to analyze and manage complex systems architectures. More specifically, it enables the user to model, visualize, and analyze the interactions among the entities of any system. Derivatives of Graph Theory, such as Network Analysis and Design Structure Matrix (DSM), are enabled by and support the application of Model Based Systems Engineering (MBSE). In fact, both DSM, as a matrix-based system modeling representation, and Network Analysis, as a graphical node and line representation, can be generated from SysML models. These representations offer a complementary way to visualize and analyze systems models.

Network Analysis and DSM also provide a powerful way to query and design modular architectures, assess change propagation and provide a host of metrics to better understand system interactions. Figure 2 provides an example network and matrix view of a notional system. While containing the same information, each view can highlight different aspects of the modeled system more readily. The benefit of using nodes and lines in a network and X's in matrices is that they are generally intuitive to read and understand by a diverse set of stakeholders.

Network Analysis and Design Structure Matrix (DSM) have proven to be very effective in the analysis, management, and integration of complex systems. They enable the user to model, visualize, and analyze the dependencies among the entities of any system—and derive suggestions for system optimization. While a DSM provides this understanding in a compact and clear representation a Network Graph can vary the size and organization of nodes and lines to highlight clusters and metrics such as degree, centrality and others. Figure 3 demonstrates further how these views can aid teams in better understanding system interdependencies in a relatively intuitive manner.



Network View

Lines indicate connectivity between elements

	A	B	C	D	E	F	G	H
A			X					
B			X	X	X			
C	X	X		X		X	X	
D		X	X			X		
E		X				X		
F			X	X	X		X	X
G			X			X		X
H							X	

Matrix View

X's indicate connectivity between elements

Figure 2: Simple network and equivalent matrix views of a simple system

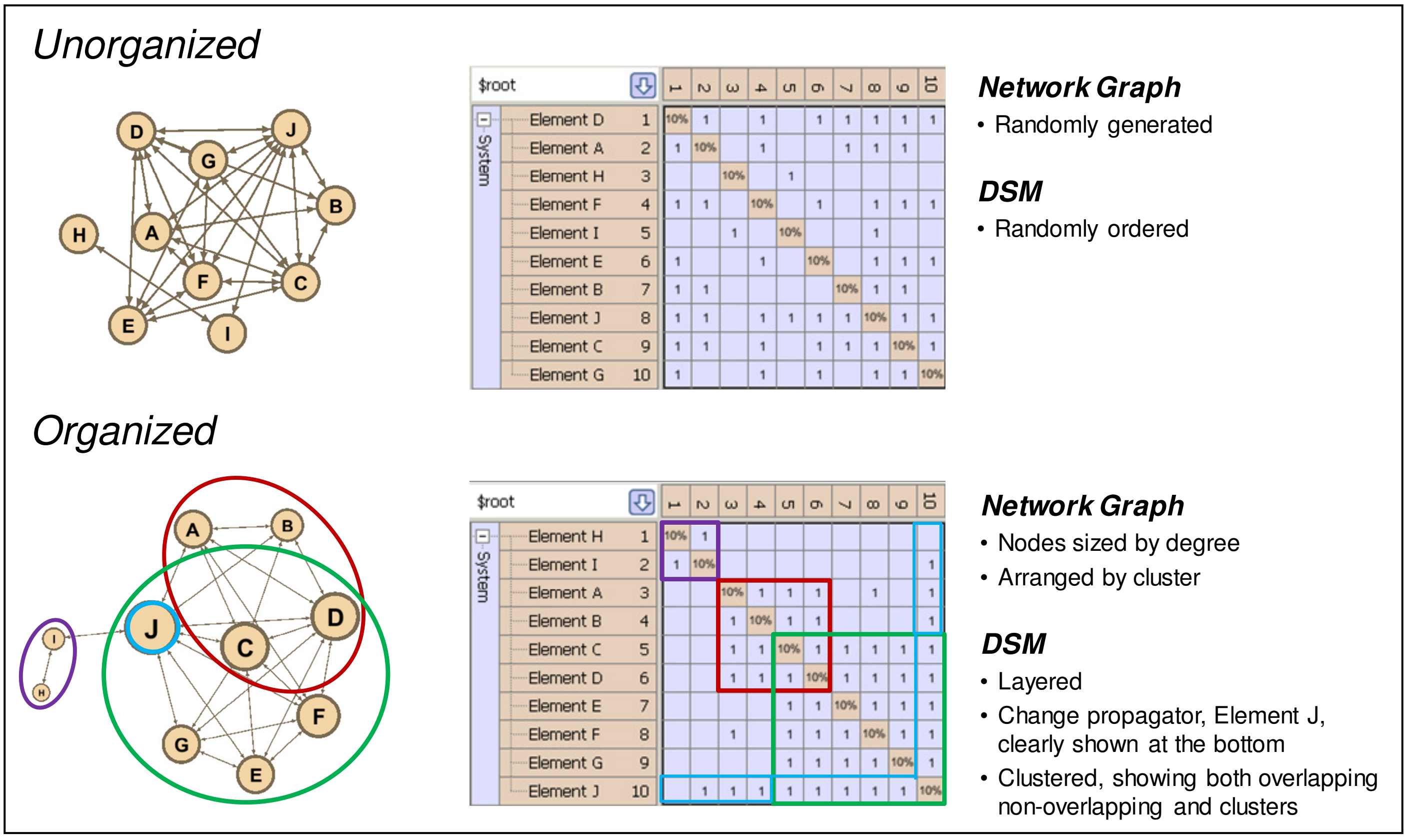


Figure 3: Unorganized and Organized Network and Matrix views highlighting clusters and degree.

These views and methods are simple and insightful yet powerful for managing, developing and improving our understanding of complex systems. Moreover, they have been successfully applied in the automotive, aerospace, construction, microprocessors, electronics and other industries as well as in the U.S. Air Force, U.S. Navy and NASA.

The use of matrices in system modeling, as done with Design Structure Matrix, can be traced back to the 1960s and ’70s with Donald Stewart and John Warfield. However, it wasn’t until the 1990s that the method received widespread attention. Much of the credit in its current popularity is accredited to MIT’s research in the design process modeling arena. Network Analysis has recently grown dramatically due to the growth of social networks. Many advanced algorithms have been, and continue to be, developed to better understand how these networks change and adapt. Also to better understand which individuals may have power as a broker or may be more central in the network etc. The views and metrics useful in social networks are also very useful in analyzing the highly interconnected Cyber Physical Systems we see today.

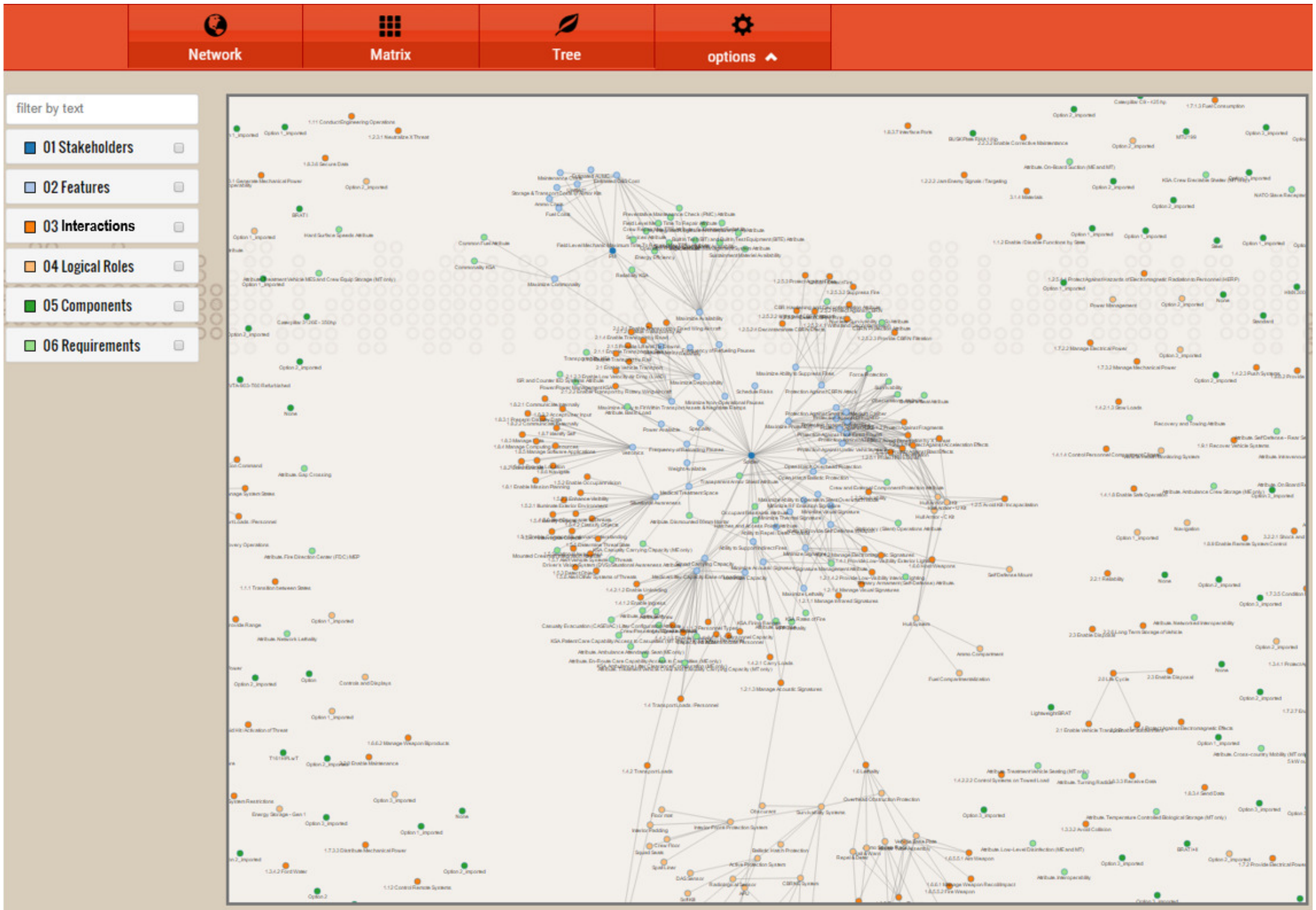


Figure 4: Dynamic visualization browser rendering a system model constructed using SysML in COTS MBSE tool

Since the behavior and value of many systems is largely determined by interactions between its elements, these methods have become increasingly useful and important in recent years not only with engineered systems but also within the natural and social sciences. Both Network and Matrix views are easily created with many MBSE tools in use today. For network tools Gephi, Pajek, NodeXL and others provide ample analytical power. For the Matrix view a list of tools can be found at www.dsmweb.org. Some of these matrix tools, such as Lattix, integrate with well-known MBSE / SysML software packages.

Not only do we want to visually display information in a rich way - we should also be able to dynamically explore it. This provides a very powerful way to learn the model and learn about the systems we develop to deliver value. As notes, this ability should be extended to users outside the limited set of skilled individuals who create system models. Dynamic visualization provides us the ability to navigate the model, to query and learn, to see what is present, and just as importantly, what is missing. It permits active hypothesis testing, experimentation and gap identification. Navigation is essential when we want to explore how contextual dynamics might impact system functionality – if a condition or attribute changes what is the potential ripple effect through a system, and what if several attributes change simultaneously? Figure 4 above is a screen shot of an MBSE model built using SysML translated into a graph. The graph can be dynamically explored, filtered, searched and translated into other complementary views such as a matrix.

In "Thinking in Systems[10]" Donella Meadows stated ".. Words and sentences must, by necessity, come only one at a time in linear, logical order. Systems happen all at once. They are connected not just in one direction, but in many directions simultaneously. To discuss them properly, it is necessary somehow to use a language that share some of the same properties as the phenomena under discussion. Pictures work for this language better than words, because you can see all the parts of a picture at once." While these comments are directed toward views more aligned with Systems Dynamics they are also very applicable to how we view systems in other languages and representations. Often we parse, filter and hide aspects of systems which hinders our understanding and we forfeit the opportunity to fully use our cognitive ability recognize key relationships. The use of graphs permits us to explore the entire model and "…see all the parts at once."

CONCLUSIONS

The days of developing and operating complex yet discrete systems is rapidly coming to an end. The proliferation of mobile, sensor and network technologies has dramatically increased technology adoption and integration into higher level systems. Every day complex systems are more interconnected and self-aware. As these Cyber Physical Systems become more and more complex they provide both incredible opportunity and risk. In brief, the increase in system complexity is outpacing our development capabilities to fully capitalize on opportunities and more importantly to identify and mitigate critical risks.

The benefits of Model Based Systems Engineering approaches are a powerful way to model, understand and manage the evolution of these highly complex systems. Translating these detailed models into dynamic visualizations can extend the full power of model based methods to the larger engineering community and deepen our understanding of these highly complex systems. When coupled with an understanding of how we learn these models have an excellent opportunity to help development teams and leadership gain insights, build intuition and speed the innovation of complex systems.

REFERENCES

[1] INCOSE SE Vision 2020

[2] http://www.omgwiki.org/MBSE/doku.php?id=start

[3] http://www.omgsysml.org/

[4] W. Schindel, and V. Smith, "Results of Applying a Families-of-Systems Approach to Systems Engineering of Product Line Families", SAE International, Technical Report 2002-01-3086 (2002).

[5] J. Bradley, M. Hughes, and W. Schindel, "Optimizing Delivery of Global Pharmaceutical Packaging Solutions, Using Systems Engineering Patterns", in Proc. of the INCOSE 2010 International Symposium (2010).

[6] W. Schindel, "Integrating Materials, Process & Product Portfolios: Lessons from Pattern-Based Systems Engineering", in Proc. of 2012 Conference of Society for the Advancement of Material and Process Engineering, 2012.

[7] http://www.omgwiki.org/MBSE/doku.php?id=mbse:patterns:patterns

[8] http://theweek.com/articles/460769/12-things-know-about-how-brain-works

[9] Vishton, Peter M., Scientific Secrets for a Powerful Memory, The Great Courses, Course No 1965, 2012

[10] Meadows, Donella, "Thinking in Systems – A Primer", Chelsea Green Publishing, 2008