Is MBSE a Tectonic Shift in the Way We Perform Systems Engineering?

Moderator: Jorgensen Raymond, Rockwell Collins, USA
Panelists: Dr. Stephen Cook, Univ S. Australia, Australia
John Watson, Lockheed Martin, USA
James Van Gaasbeek, Northrop Grumman, USA
Ralf Hartmann, EADS Astrium GmbH, Germany

Abstract

Explore the possible transformation of system engineering from a document-centric culture/ old school system engineering thinking to a model-centric culture/ information mgt thinking perspectives - and all the hurdles, hijinks, hysteria, and hype that we'll encounter along the way.

Biographies

Moderator

Raymond W. JORGENSEN is an avionic systems engineer working on flight deck applications that are used across a diverse community of aerospace customers. He actively works in requirements analysis and design, serving as a corporate knowledge management resource for system engineering processes, tools, and training, continually seeking best practices in the application of system engineering.

Panelists

Dr. Stephen COOK is the Director of the Defence and Systems Institute and a Technical Director of the Defence Systems Innovation Centre. He has had a varied career that commenced with over ten years' engineering experience in the telecommunications and aerospace industry after which he joined the Defence Science and Technology Organisation (DSTO) rising to Research Leader Military Information Networks in 1994. Since 1997 he has been with the University of South Australia as the foundation DSTO Professor of Systems Engineering. He has a wide span of research interests including systems modelling, systems engineering of C2 systems, systems approaches for defence capability development, acquisition modernisation, and theoretical frameworks to support the coherent teaching of systems engineering. Prof Cook is a Past President of the Systems Engineering Society of Australia.

John WATSON is a Principal Member Engineering Staff at Lockheed Martin MS2, in Moorestown NJ. John's principle job function is as a systems architect. He has been involved in modeling systems architecture efforts in a number of Lockheed Martin programs including the Aegis Open Architecture Program, the Aegis Modernization Program, Orion Space Program, and multiple radar programs. He has over thiry years of industry experience covering a wide spectrum of responsibilities in leading and managing systems and software architecture, design and implementation both in the DoD and the telecommunication commercial industry. Recently

John has been involved in the a corporate engineering effort to promote model-based system development across Lockheed Martin.

Jim VAN GAASBEEK has 35 years experience analyzing and developing rotary-wing and fixed-wing aircraft, launch vehicles and spacecraft, both in the United States and European defense environments. Beginning as a rotor aeroservoelastician, his career has progressed with experience in constructive and virtual simulation, accident investigation, vehicle-management system design and systems engineering, concentrating in risk management and requirements development, management and verification. He was invited to participate in this panel to provide some controversy, obviously being cast against type.

Ralf HARTMANN has been an INCOSE member since 1996 and was a founding member of GfSE, the German Chapter of INCOSE. Currently he is INCOSE Director for Strategy having taken up the role in mid 2008.

Ralf has been President of GfSE, Chair of the 2nd European Systems Engineering Conference (EuSEC) in 2000 and Co-chair of the International Symposium 2004 in Toulouse. Within INCOSE, he has held several positions such as co chair of the Standards Committee and the Modelling and Tools Technical Committee and he is the CAB representative for Astrium Satellites. Ralf was the principal INCOSE delegate for the global GEOSS initiative from the day when INCOSE joined as a participating organisation until he took over responsibility as Director for Strategy. In 2005, Ralf was selected as an INCOSE fellow and he received the INCOSE Founders Award in 2008. Among his technical contributions two best papers at symposia and his contribution to the Systems Engineering Vision 2020 may be highlighted.

Ralf Hartmann is currently a Vice President for Control Ground Systems, Engineering Tools, and Satellite Functional Verification Infrastructure at Astrium Satellites within EADS. Since 1987 he has worked for Astrium and its predecessor companies in the area of robotics, automatic control, simulation, S/W development and systems engineering. During his career he has held various technical and project management positions as well as senior management responsibilities. Today he is also a member of the EADS Systems Engineering Steering Group and the Engineering Board within Astrium Satellites.

Highlights in Ralf's career include the implementation of a comprehensive Systems Engineering Qualification Program and the implementation of the Satellite Design Office, a conceptual design centre, which he led for some years. Furthermore, Ralf was one of the key authors of the current European Space Systems Engineering Standard (ECSS-E10).

He received a Diploma in Electrical Engineering from the University of Karlsruhe and he is a certified project management professional.

Model-based Systems Engineering
The University of South Australia View
Stephen Cook, Director Defence and Systems Institute

Background

My first introduction to model-based approaches for systems engineering occurred during my PhD studies in the late 1980s (Cook, 1993). My research project involved automating the production of measuring instrument specifications. I started by taking a classical document-centric approach but the design and construction of just one prototype made me realise that natural language is a very limited medium for holding requirements. From there, I investigated mathematical linguistics as a knowledge representation paradigm and artificial intelligence for reasoning. This led me to using Prolog to build a shell that could hold a requirements specification and perform some useful reasoning. Within this tool, named *Specriter 3,* the requirements were held in Prolog clauses in a frame-based structure that provided semantic context, inheritance, and content-sensitive requirements entry and editing. The most important concept behind this work was not the functionality of the tool itself but rather that the requirements were held in a model that inherently supported reasoning and avoided the more difficult aspects of natural language processing. Multiple views were able to interrogate the requirements and also to generate a conventional MIL-STD-490A requirements document automatically. This tool was used in teaching for some years but I choose to return to a career in defence telecommunications research and engineering and did not pursue the development of the tool after 1990.

Part of my role in the defence communications research laboratory in which I worked, was to provide advice on a range of technical issues to do with major defence communications projects. At that time, the early 1990s, it was further reinforced, that the document-centric approach was struggling to deal with complex Information and Communications Technology (ICT) projects (project definition studies ran to thousands of pages of documentation.) During that time, one of the major contributions made by my team was to produce mathematical network performance and behaviour models that informed the function and performance specifications and verification methods. In more than one project, these models became part of the contract and replaced certain textual parts of the project documentation.

Some years later I accepted a position in systems engineering research and again became active in research in requirements engineering and support tools. This started with a project that commenced as a tools interoperability research task and progressed to a model-based approach to requirements engineering (Scott and Cook, 2008). In the early stages of this project, it was determined that the AP-233 Application Protocol for systems engineering data that was part of ISO-10303 (Version 5.1) was suitable for transfer of textual requirements between tools, however, it was considered no more suitable to act as a knowledge representation for artificial reasoning than the existing representations utilised by contemporary requirement management tools. To overcome this limitation, a model-based approach was adopted whereby a formal grammar was designed as the underpinning knowledge representation for the requirements clauses. The utility of this approach was investigated through the construction of software tools. An evaluation of these prototypes was very encouraging both from the perspective of the elicitation of new requirements and the high success rate in parsing and evaluating existing requirements. This research strengthened the argument that model-based approaches not only offer significant potential but that this potential can be realised. The fruits of this research are currently the subject of commercialisation efforts.

Contemporary MBSE activities: an Australian defence perspective

In addition to this research, the Defence and Systems Institute, has been applying SysML-based approaches to capture architectural descriptions for a range of software-intensive defence systems. We have found that utilisation of the tools assists in capturing design descriptions providing they are used on a regular basis. We found that one of the major benefits is the automatic configurations management and the propagation of model changes. Also it was found that the use of the tools facilitates the production of a more comprehensive system description that also combines some inherent behaviour modelling.

Two limitations were, however, identified. Firstly, the tools we used captured a sequential representation of the activities of the system without any depiction of the time relationships of and between activities. We found that this can make it hard to understand the temporal performance of the system. Secondly, the diagrams within the model were delivered within documents rather than as modelling entities. We found this negated the some of the benefits of the model-based approach, specifically inherent model and version consistency and automatic change propagation.

The Australian Department of Defence, through groups such as the Defence Science and Technology Organisation (DSTO) and the Defence Systems Integration – Technical Advisory, is pursuing substantial research programs in MBSE (DSI-TA, 2010). This work is focussed on understanding the current penetration of MBSE into the department and the implications of forthcoming MBSE practice on major systems integration projects that are currently being initiated.

Campbell (2010) surveyed MBSE use within the Australian Department of Defence and has identified extensive utilisation of MBSE across a number of projects. He mentions that there is substantial use of CORE for modelling joint and maritime systems and this has been found very useful at the front end of projects and far more efficient that document-centric approaches. He also cites that the Architecture Analysis Design language (AADL) has been used in the aerospace domain. The AADL users surveyed think that it is a good tool for smaller projects.

Of particular note is Whole-of-Systems Analysis Framework (WSAF) that has been developed by DSTO to support capability analysis and for the generation of capability development documentation (Robinson and Graham, 2010). WSAF employs MBSE principles within the guidelines of the Defence Architecture Framework to address the issues of knowledge management, requirements elicitation and documentation traceability in the capability development process. WASF has been used for several projects with success and there are plans to expand its application to new projects.

Campbell's paper, which very effectively articulates the University of South Australia view on MBSE for Australian Defence, has findings consistent with the general stance of the INCOSE MBSE WG. It also identifies the following points in relation to the introduction of MBSE into defence projects in Australia.

- There is rising interest in the use of MBSE in Australian defence projects and some early adopters.
- The success of the DSTO WSAF toolset in producing excellent documentation has been clearly demonstrated; this success is acknowledged by the customer.
- Reported benefits of MBSE include:
    - more accurate architectural designs
    - better traceability

    - reduced work effort
    - more easily produced traceable documentation.
- MBSE and the tools that have been developed to employ it are compatible with the traditional SE approach. However, successful use of the MBSE approach requires a degree of formalism and rigor that is often missing in early project activities.
- Architectural design and specification is a key element of MBSE and should be the starting point for all design work.

Campbell found that the main inhibitors to more widespread adoption of MBSE are entrenched culture, general resistance to change, and the steep learning curve. He notes that the cultural effects arise as much from the need for management to commit to a significant upfront effort as a project gets underway as from any other cultural trait and while the steep learning curve was the barrier most often mentioned, the need for training was not evaluated as being particularly necessary.

Given that project problems have frequently been traced to inadequate attention to systems engineering fundamentals, it would appear that MBSE, with it inherent early capture of operational needs, systems requirements, and the evolving architectural design, has the potential to drive a cultural change that could make a significant difference to project outcomes.

Conclusions

Some 20 years after coming to appreciate the limitations of document-centric approaches, I am now more convinced than ever that contemporary MBSE approaches can increase the rate of knowledge acquisition by project participants, improve the quality of system designs, improve knowledge management, improve project communication, provide automated reasoning, and improve the application of systems engineering practices.

The Defence and Systems Institute at the University of South Australia has identified MBSE as a major focus area for research and teaching. We see MBSE as the vehicle that can create the paradigm shift needed to direct more attention onto the tenants of systems engineering at the early stages of major projects and as such it has the potential to make a valuable contribution to successful project outcomes.

References

Cook S.C., "A Knowledge-Based Specification Generation System for Computer-Aided Production of Measuring Instrument Specifications", *Measurement*, Vol 11, pp 235-255, 1993.

DSI-TA, *Annual Report 2009*, Defence Systems Integration Technical Advisory, 25 March 2010

Scott, W and Cook, S.C., "Automated Requirements Quality Analysis through Requirements Comparisons", *Proceedings of 18th Annual International INCOSE Symposium*, Utrecht, The Netherlands, 15 to 19 June, 2008.

Robinson K. and Graham D., "An improved methodology for analysis of complex capability", *Proceedings of SETE 2010*, May, SESA.

Campbell P., *Initial Model Based System Engineering Study For DSI-TA*, University of South Australia, 2 April 2010.

Is MBSE a Tectonic Shift?

Table of Contents

Abstract

Abstract: Is Model-based System Engineering (MBSE) a Tectonic Shift in the Way We Perform Systems Engineering? Explore the possible transformation of system engineering from a document-centric culture/ old school system engineering thinking to model-centric culture/ information mgt thinking perspectives - and all the hurdles, hijinks, hysteria, and hype that we'll encounter along the way.
Purpose: To explore some of the controversy between system engineering standard practice and model based system engineering, shedding some light on “what’s new, what’s different, and what’s really the same thing packaged with a different ribbon.”

Introduction

To answer the abstract above there are a number of dimensions that MBSE can impact and need to be examined. The first section will provide a foundation of why we need to change. The following sections will identify some of those dimensions that need to change. Finally the conclusion will state if this is or is not a “Tectonic” change to the way we build and support our systems.

Why Change Anything?

Simple, our System Engineering environment is changing and creating new challenges that need to be addressed. These changes include:

System Size and Complexity

The systems we are being asked to build today are continually expanding in size and complexity. These systems are no longer just complex software running on a few computers. They consist of many computer sets with complex software interconnected via a series of switches, routers and other specialty hardware items. With this type of system reliability, survivability, security, maintainability and performance across the system are now even more important issues.

System Engineers are the focal point of this system effort. Their responsibilities include (1) derive and maintain the overall system architecture; (2) provide the necessary analysis and simulation to minimize development risks and determine the best solutions; (3) ensure the system "illities" are being satisfied; and (4) to verify the pieces of the system when interconnected will work.

Cheaper, Better, Faster

In addition to this complexity we are being asked to make these systems in less time, to cost less, be state of the art, be open, and be easily adaptable to future needs.

Collaboration of Dispersed Teams

To address these large complex systems it takes multiple teams each providing expertise in a particular technology and domain. It is not enough to simply bring these domain experts together, but these teams need to collaborate and exchange their knowledge, experience and ideas. To derive the "cheapest, fastest, best" solution, to make the best trade-offs and to re-use where appropriate, these domain teams need to consider alternative solutions that span across their technology domains.

Dimensions of Change

Standards

To help guide us so we all so we all change in the same direction and with the same target in mind we must utilize industry standards. If not, we could all transition to MBSE and still have difficulty communicating. We might as well continue to use PowerPoint and Visio as our modeling tools.

OMG has provided our foundational industry standard modeling language to help us specify, analyze, design and verify our systems called SysML. We can use this language to capture the results of most of the existing SE tasks we perform in the document-centric culture.

SysML is not the only OMG standard we need to utilize. Other standards, such as UML, UPDM, XMI and MARTE, are also needed to complete our ability to fully specify and interact with other tools and domains.

These standards must be driven from our industry needs. It is not just an academic exercise; they must provide practical solutions and techniques for capturing, describing, managing and viewing vast amounts of information. That means we, the industry, must help mold these standards by funding and supporting the efforts of these standards bodies with the right people.

Tool Vendors

A modeling tool provides the mechanisms to implement these standards and capture the system information. Selection of a tool should not be based on if they are 100% compliant with these standards. That is a given, if they don't the tool is just not acceptable. So how will the vendors differentiate themselves?

These large complex system models will contain tens of thousands, if not hundreds of thousands of model elements. Entering, viewing selected information, and maintaining the integrity of this data cannot be done just with human hands.

The differentiator between vendor's tools will therefore be their ability to automate modeling tasks to increase our productivity, measure completeness and accuracy, maintain the integrity of these artifacts, minimize maintenance and seamlessly integrating our modeling tools with other SE analysis, simulation, configuration management, implementation and test tools.

If we look at the evolution of some of our everyday products like cell phones, the initial cuts were basic and crude compared to today's evolved products. Cell phones filled your trunk, had limited range, had limited coverage and had very basic calling features. Today's devices, to name some, fit in your pocket, last a week or more on a battery, include internet features like emailing and searching, take still pictures and movies, record audio, play videos, play music, provide games, text message, includes a phone directory, includes a calculator, uses voice recognition, includes GPS and have local apps that integrate these features and internet capabilities.

If we look at this spectrum of evolution of the cell phone and compare it to where our SE tools are today, I truly believe they are closer to that cell phone in your trunk then today's cell phone. Although this is a tremendous opportunity for the tool vendors, some of these techniques and needs are only beginning to surface, but the winners will be those that manage change the best.

Customer

We are just starting to see some of our customers providing a model based product description. Since they have to overcome the same obstacles as the rest of the industry, it will probably be some time before we see all our proposals being delivered as model based.

However, when it does occur, the request for a proposal will not just be a document containing textual requirements, but will be a model with improved rigor, completeness and provide a more visual understanding of the need. It could ultimately include an executable model where both sides could benefit from an enhanced dynamic behavioral understanding with the addition of the time dimension.

Company

Each company that embarks on MBSE must be willing to make an investment in training, tools and their personnel. Companies must accept the fact that as they transition and develop their employee's MBSE skills each employee that transitions will have a momentary slowdown in productivity. This is the same slowdown that takes place with the adoption of any new technology.

The motivation to transition cannot just be a grassroots movement from small pockets of engineers within the company; it must also come from the leadership team. Both have to believe it is a good investment in the company's future and will move them to another level of engineering capability.

This investment is needed to stay competitive and to be compatible with our customers. But even if our customers can't transition immediately, the company must see this investment as an improvement in productivity and an improvement in our abilities to engineer complex systems and reduce support costs.

People Changes

Change is hard for most of us. Once we learn the ins and outs of how to perform a task, we become proficient and comfortable at executing that task. Now we're asking our SEs to learn a new language and a new way to think about a problem, to change the foundational way they communicate their ideas and thoughts. For some with backgrounds in UML, object oriented design and software development the transition is not as bad, but for many it is a first time exposure.

So we can universally read, understand and communicate, all SEs must become fluent in the language. Learning this new language is the same as learning any other verbal or software language and should not be taken lightly. It is not a matter of simply attending a class. It must be practiced for an extended period to become fluent. Until then it is going to be a source of frustration by becoming an obstacle in expressing their ideas and thoughts.

In addition to a new language they will also have to learn a series of new tools to capture and manage this information.

The good news is the tasks they perform today are essentially the same tasks they will continue to do after the transition, so that their vast domain knowledge will continue to be leveraged. Therefore, the training they receive should be relevant to each engineer's domain expertise, and the practice tasks as they hone their skill should also be in a familiar environment.

Culture Changes in Process

Once we establish a new language there are two additional changes necessary. The first is the way we approach a development problem and the second has to do with our focus on textual requirements.

First we must embrace iterative development. In essence that means "do a little, test a little". Focus first on those characteristics of the system that are architecturally significant or introduce the most risk. An important element of iteration is not to be fixated on waiting for the specification to be 100% done before collaborating with those that are the recipient of the specification. Waiting undermines our creativity and ability to determine the "cheaper, faster, best" solution. The specification needs to be a record of what both sides agreed upon, not a one-way communication tool. Specifications still have to be controlled and managed, but there must be more collaboration and understanding of the problem before we determine the best solution. This is more of a workflow problem not a modeling problem. But since the models enable our ability to share information it provides part of the solution.

Second we can no longer depend on describing our systems with just textual statements, i.e. requirements. Textual statements are proven to be vague, incomplete and easy to misinterpret. The modeling language provides the rigor and precession to specify a system much more precisely. In addition the language includes graphical notation to allow us to visualize and understand concepts quicker. Therefore we must allow the full capabilities of the modeling language to provide the specification and not just textual statements.

Conclusion

When you look across all the dimensions of change, including our engineering processes, the way people think, our companies, the evolution of tools and our customers, and that we are changing and leaning together, i.e. pulling the industry up by our bootstraps, it is a significant change.

It is not that we can't do it, others have. It is a longer road than most would wish. To fully recycle the old ways with the new, it is probably a generation away.

Therefore we have to be mindful this evolution is taking place and leverage the lessons learned from others as they break new ice. Not all new MBSE ventures in this evolution are going to be successful, but all will provide new insight that will help us steer the course.

The system engineering tasks that we do for MBSE are not much different than those we do in a document centric approach. What does change is how we execute those tasks.

The key element to remember is what MBSE provides. MBSE provides more structure in how we derive, gather and capture system related information. In a document based world things are specified using free text. So the relationships between shared pieces of information are loosely coupled and it is up to the diligence of the individual to ensure consistency across multiple documents.

In an MBSE world an informational structure, including the relationships between information, is created and maintained as part of the normal modeling process. This informational structure allows us to **automate** our ability to;

- View our systems from many perspectives, such as domains of interest, capabilities and abstraction layers
- Defined Information once and reuse it across these multiple perspectives and views
- Measure the impact of change more effectively
- Measure the integrity of information ensuring it is complete, consistent and accurate across all system views

Therefore I say it is a tectonic change, probably of magnitude 5 or 6. If we plan ahead, stay the course and manage the change we can easily ride it through. We humans have observed complexity before and have solved it by finding ways to automate the mundane and repeatable tasks, and by abstracting away unnecessary information to view only the relevant information thus allowing us to see and understand the problem at hand. After all, the first step in solving a problem is first being able to clearly visualize it.

Model-Based System Engineering (MBSE) Panel Discussion
International Council on Systems Engineering
International Symposium
Chicago, Illinois
11-15 July 2010

Panel Topic:

Each panelist is to address both the potential (and/or current) benefits that may be realized by Model Base System Engineering, and address the challenges that organizations will or are experiencing with the application of MBSE methods and tools (and the frustrations/ disappointments that accompany such usage).

Position:

Much has recently been written about the emergence of Model-Based System Engineering (MBSE) as a powerful tool in the system engineer's toolbox. It is often linked to the release of SysML and the utility of that diagramming language in facilitating the rapid spread of MBSE. However, MBSE has been used with success for many years, and doesn't depend upon a particular notation for its deployment.

Jim Long [Long 2003] has described the evolution of an MBSE technique from early work in software development. He has documented the progression from the Software Requirements Engineering Methodology (SREM - 1974) through the Systems Requirements Engineering Methodology (SysREM) to the Distributed Computing Design System (DCDS – 1977) which was then extended into commercial products (Requirements Driven Design-100 and CORE). The methods, and their attendant tools, used the functional flow block diagram (FFBD), extended functional flow block diagram (EFFBD), behavior diagram (BD) and N-squared diagrams (N2) as the graphic representations and "modeling language".

In the same time frame, Wymore was developing a mathematical theory supporting model-based systems engineering, published in his book in 1993 [Wymore 1993]. Both the work reported by Long, and Wymore's theoretical foundation, tend to see MBSE used for behavior modeling.

Grady has written about modeling and a Unified Architectural Description Framework (UADF) [Grady 2009]. Grady is generally method- and notation-agnostic. He does, however, address modeling of more than system behavior, recommending modeling of all aspects of the system and presenting the results in a unified manner. His approach includes models of such non-behavioral aspects of the system as reliability, maintainability, supportability and availability (RMS&A).

Model-Based Systems Engineering (MBSE) can use either static models or dynamic, executable models, in the form of modeling and simulation, or both. RMS&A models will typically be large, static, mathematical models expressed in spreadsheets, for example. Behavior models can be represented as static diagrams, such as FFBDs or Activity Diagrams. However, the models are much more valuable when brought to life through execution. This latter use of models is incorporated in the modeling and simulation activity.

M&S has been used to perform MBSE with success in the military-aerospace industry for decades. Defense ministries/departments have used modeling and simulation in planning exercises, often referred to as war games. Such games have been played at the campaign level and above to explore different aspects of **D**octrine, **O**rganization, **T**raining, **M**aterial, **L**ogistics, **P**ersonnel and **F**acilities (DOTMLPF) in representative threat and cooperative environments to determine potential outcomes and to determine what changes to DOTMLPF would be necessary to assure better outcomes. The contractor community participates in these exercises from time to time, and provides models of existing and feasible potential material products and services for use in the exercises to ensure feasibility.

MBSE is also performed by contractors to refine their products as the deployment environments change, and to support acquisition and design, development, test and evaluation (DDTE) activities. Modeling and simulation is performed over a wide range of levels of abstraction, from the battle-space level to the subsystem level. See Figure 1.

Figure 1. Modeling and Simulation is Performed at Various Levels of Abstraction.

Modeling and simulation activities and tools fall into three categories – constructive, virtual and live. Constructive M&S consists of executable models that, typically, run faster than real time and have no direct human interaction. Virtual modeling and simulation makes use of executable models to run in real time with one or more humans interacting with them (human-in-the-loop simulation). Live modeling and simulation combines the simulated, human-in-the-loop, system model operating in conjunction with live assets.

Campaign and mission constructive modeling is frequently performed using extensive computer codes to investigate the interaction of friendly and threat forces over the course of days or weeks, and across at least a theater-wide geographical region.

Mission and operational modeling is often performed using virtual simulation. Northrop Grumman developed the five-degree-of-freedom Large Amplitude Multi-mode Aerospace Research Simulator (LAMARS) simulator for both mission and operations modeling and for engineering development work [Linklater and Slutz 2007] (see Figure 2).

Using this simulator, other high-fidelity simulators, and lower-fidelity simulators, missions with “m x n” engagements could be modeled with up to 4 “blue” aircraft and up to 20 “red” aircraft. These engagements were used for such system-level tradeoffs as determining the number of engines needed, or the number of flight crew required to fly the mission. Basic survivability information could be extracted and used in the RMS&A analyses to determine such parameters as necessary fleet size.

Constructive executable models are also used at the operational and functional level. Northrop Grumman has had success building operational and functional models of systems and executing them using the Vitech CORE /

COREsim tool[1]. In a recent case, we built a model of the system with more than 20 interrelated vignettes covering, among other activities, system startup, integrated combat turns and aerial refueling. All model information was stored in the single tool database, and all linked with relationships.

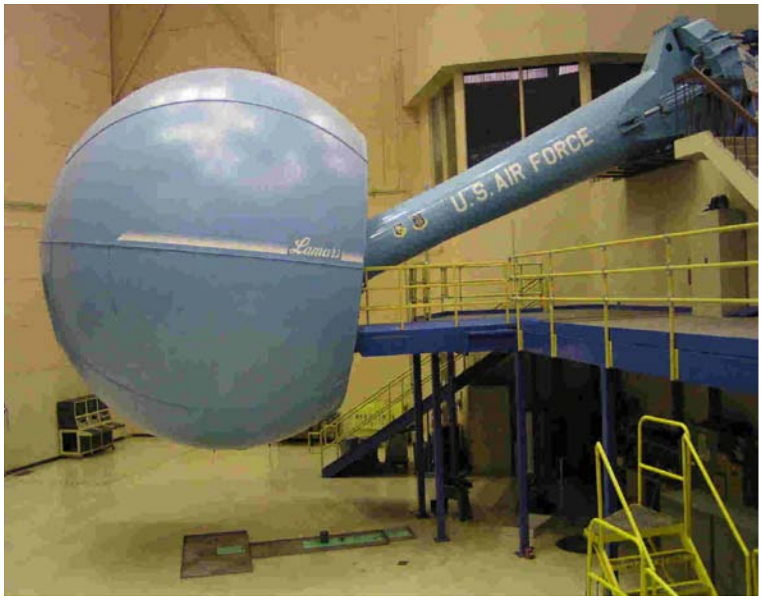

Figure 2. Large-Amplitude Multi-Mode Aerospace Research Simulator.

MBSE Benefits:

There are at least four benefits to using MBSE in the form of modeling and simulation.

Common Understanding

It is a best practice to ensure that all stakeholders share a common understanding of the problem to be solved and the material or service system to be developed. Dependence solely upon mental models of how systems are used, and how systems and subsystems interoperate, are dangerous because peoples' mental models rarely match. Explicit models, which define the relationships of customer use, functionality, and physical design, help to avoid "a million little models" all locked in individual's heads. It forces all stakeholders to work to the same model. Building a unified model, in a single database, as discussed above, or related set of databases, with a "write once, use many" philosophy, forces common definitions of systems and their components, and their uses.

As an example, assume that one is developing a theater-wide model of a maritime defense capability in which multiple systems will be used in solving the problem (see Figure 3). Existing models of various overhead, airborne and surface assets are woven into one executable model and the engagement is run multiple times. This can be executed at a single site, with all the models running on computers locally linked. Alternatively, if the models aren't all under unitary control, a network-centric approach can be taken, with the models running on computers at various locations with a single network control of the game, such as is done by Northrop Grumman using its Cyber-Warfare Integration Network (CWIN). See Figure 4 for a notional description of the CWIN[2].

Note on the figure that the CWIN has nodes at various Northrop Grumman sites. With customer permission and supervision, the CWIN can also connect to selected Government and other contractor sites to expand the models that can be included in the simulation.

---

[1] The Northrop Grumman Corporation does not endorse any particular vendor's tools or methods.
[2] It is believed that both Lockheed Martin Corporation and The Boeing Company have similar networked modeling and simulation capabilities.

Figure 3. Notional Maritime Defense Scenario.

Figure 4. Distributed Modeling and Simulation Supports Expanded Modeling.

Refine ConOps and Develop OpsCon

The modeling and simulation of potential solutions in the proposed operating environment allows the stakeholders to examine how the environment will respond to the introduction of the solution, and how they may wish to modify the enterprise-level Concept of Operations to ensure that there is a better match of the solution to the environment to be expected at, and after, deployment. Additionally, as potential operators, or operator surrogates, are exposed to potential solution operations, in the intended environment, the operators can develop the Operational Concept for the solution in advance of its deployment.

Support of Trade Studies

MBSE in the form of modeling and simulation, as practiced for decades, has allowed us to do high-level problem-space trade-offs on such things as the appropriate composition of a strike package, on fleet size and needed availability, and on such solution-space trade-offs as number of flight crew or engines and data link needs. The ability to perform the trade-offs with a representative operational environment (friendly and threat) and, for virtual and real simulations, to include representative human operators, has allowed rapid convergence to balanced solutions.

Support of Verification and Validation

MBSE also provides value in both verification and validation. First, by ensuring widespread use and understanding of the models across the stakeholder community, the operational needs and the solution requirements may be validated. Additionally, the operational scenarios implicit in the models can be used as the basis for planning of product (material or service) verification and validation.

The MBSE approach requires that one define one or more candidate architectures at each level (campaign architecture, mission architecture, operational architecture or functional architecture) to define the entities that are interoperating and their various relationships. This is often an iterative process as the MBSE analysis allows the stakeholders to refine their understanding of the needs and to refine the architectural definition. At any level, the desired activities are assigned to the entities. This relationship is shown, at the lower level of the M&S pyramid, in Figure 5.

Figure 5. Interrelationship of Operational, Functional and Physical Architectures.

The figure represents the linkages between the work done in each of the three domains, the three architectures and the feedback that drives the iterative process.

The work products that are produced by the M&S work in each of the three domains is also useable in preparation of the various DoDAF documents typically required during the DDT&E Program. Given that the three architectures and their artifacts are developed and maintained in a tool, a significant portion of each DoDAF document can be automatically generated from the tool.

An additional benefit of developing architectures at various levels of abstraction in the MBSE is promotion of Modular Open Systems Architecture (MOSA). Typically, the "modular" and "open" parts of MOSA are perceived as aspects of the physical architecture. This can be achieved by using non-proprietary standards (e.g., open bus protocol standards). But one of the principal thrusts of MOSA is that systems incorporating MOSA allow the customer to recompete portions of the product to implement select enhancements into the overall architecture. By using the approach seen in Figure 5, one explicitly shows any would-be supplier who wants to integrate a new capability into the "open physical architecture" how everything is related to provide operational capability.

MBSE Challenges:

Implementing MBSE and modeling and simulation in an organization poses several challenges:

Philosophical:

Building the models for MBSE typically requires a rigor that may not be desired by members of the stakeholder community. Additionally, the time spent on building and proofing models, and in their execution to support the development effort, is seen by some to be expensive and a waste of time. These perceptions are difficult to overcome, and require executive management with the will to enforce the discipline necessary to use MBSE effectively.

Tools

Full MBSE, with models spanning all the aspects of the problem and solution space, will, typically require use of many tools. Campaign modeling tends to require purpose-built tools that are specific to the problem domain and its environment. Mission modeling may also require purpose-built tools, owing to the specific nature of the problem and the solution. Operational and Functional modeling may be amenable to use of a general-purpose modeling tool. But if one is to apply MBSE at each level across the entire problem/solution domain, then one is going to need multiple tools. For example, at the operational or functional level, one can use a general-purpose simulation tool for behavior modeling, but one will need separate tools for modeling RMS&A or safety.

The need for multiple tools can cause concern over the cost of the tools, and their maintenance, and user training. Additionally, a risk may arise if data need to be transferred between tools, as the transfer may introduce errors.

Summary:

MBSE, using modeling and simulation, has been in use for decades with great success supporting the development of advanced material solutions for the defense departments/ministries of the world. The approach provides a basis for common understanding across the stakeholder community; a mechanism for refinement of the concept of operations and operational concept; support for trade studies; and support for verification and validation planning. There are challenges. MBSE increases front-end program costs, and takes time. Additionally, it can increase the need for multiple tools.

However, it is a proven approach to doing business – a best practice.

References:

Long 2003	Long, James E, Model Based System Engineering (MBSE): Early History of a Proven Systems and Software Development Process, presentation, June 2003
Grady 2009	Grady Jeffrey O, "Universal Architecture Description Framework", Systems Engineering, Vol. 12, No. 2, 2009
Linklater, Amy and Slutz, Jeffrey	Exploring the Large Amplitude Multi-mode Aerospace Research Simulator's Motion Drive Algorithms, AIAA Modeling and Simulation Technologies Conference and Exhibit, AIAA 2007-6470, 20 - 23 August 2007, Hilton Head, South Carolina
Wymore 1993	Wymore, A Wayne, Model-Based Systems Engineering, An Introduction to the Mathematical Theory of Discrete Systems and to the Tricotyledon Theory of System Design, CRC Press, Boca Raton, Florida, 1993

Acknowledgments:

The panelist wishes to thank Messrs Ray Hudson and Doug Huntley for many enjoyable hours of discussion on the topic of MBSE and architectural development.

Biography:

Jim van Gaasbeek has 35 years experience analyzing and developing rotary-wing and fixed-wing aircraft, launch vehicles and spacecraft, both in the United States and European defense environments. Beginning as a rotor aeroservoelastician, his career has progressed with experience in constructive and virtual simulation, accident investigation, vehicle-management system design and systems engineering, concentrating in risk management and requirements development, management and verification.

Is MBSE a Tectonic Shift in the Way We Perform Systems Engineering
European Space Systems Engineering Perspective
Ralf.Hartmann@astrium.eads.net
Harald.Eisenmann@astrium.eads.net

Background

Systems-engineering methods, practices, and tools have been successfully used in European space programmers over the last decades. The common reference for the systems-engineering process is the European Cooperation for Space Standardization's E-10 series of standards (see, for example, ECSS 2009b). Although many kinds of models are used to support the development and operation of space systems, the process still very much relies on documents to capture all project information, in particular for the major reviews. Based on the current projects the following can be observed:

- Compared to systems engineering the different domain engineering activities are far better supported in terms of engineering tools covering analysis and design activities. In particular the mechanical thermal engineering tasks seem supported the best. A cross discipline exchange of models hardly takes place.
- The systems engineering support currently is limited to requirements engineering tasks. An integration of requirements management tools to engineering tools is not supported

Therefore an integrated multi-disciplinary model-based representation can currently not be reported.

The Vision

In line with INCOSE's *Systems Engineering Vision 2020*, there is a commonly shared vision between agencies and industry that a common virtual (computer) model-based representation, which is shared between the different systems engineering and the engineering disciplines will enhance the effectiveness and efficiency of space-system development. In particular MBSE is expected to facilitate and improve early and continuous validation and verification, to enhance data consistency, to help develop increasingly demanding and complex systems, and to enable the successful development of systems of systems.

This multi-disciplinary representation is a virtual model which can be considered with the different H/W-based models, which are used in particular for integration and verification purposes. The main difference is that the virtual model, follows the evolving design along the system development life-cycle, and will be provided based on the model-based data. The virtual model will consist at least of the following:

- A multi-disciplinary functional system simulator, allowing the operation according to the real spacecraft. This can be considered as the enhancement of the functional system simulation facilities, which currently are typically in use for verification purposes.
- A multi-purpose viewing facility, which provides the different engineering views (e.g. SysML views, MCAD), which is fully integrated to the simulation facility and allows inspection, or injection. The model data and views would be obtained from the design tools and re-organized for the viewing use case automatically. This representation could also be used to conduct reviews on static or dynamic data – connected to the functional system simulator or offline.

- An integrated model repository, providing besides basic model management features (such e.g. as configuration control), also the (system) model (data) verification, validation.and completeness checking. It is not assumed that there will be a single model repository covering all the data. Rather a flexible, open cascaded model repository architecture is required. This allows a sharing of the required information between the different parties involved in a project. Effectively this can be considered as an extended enterprise.

In this vision documents would be just a particular report generated automatically from the model repository. All quality assurance related data such e.g. as non-conformances or problems, would be fully formalized and linked directly to the related items.

The benefits of this virtual representation for the overall process would comprise e.g. the following:

- Eased re-use of models for similar projects: delta engineering
- Formalized and continuing model verification, completeness and consistency checking
- Improved design consolidation through the extended enterprise where model data can be efficiently shared and exchanged.
- Elaboration of design options and system level trades effectively taking place

The Current Limits and Blocking Points:

In the last decade MBSE evolved significantly and INCOSE played an important role in coordinating and disseminating the effort. The major items currently being followed are the following:

- Developing SysML as a language for system engineering purposes
- Developing AP233 as data exchange standard for systems engineering data

SysML is gaining an increased awareness in industry (on technical and management level), with agencies and also academia. Many different activities are currently being performed to demonstrate and validate SysML for different use cases. While on one hand this is quite valueable, on the other hand in the in the discussion SysML and MBSE are mixed up.

It has been the right decision to build the modeling language for systems engineering purposes on UML – for many reasons. Nevertheless the shorting comings of SysML tools, which are in fact UML specifically profiled UML tools. The deployment scenario of a modeling tool for systems engineering and a software engineering tool is very different. Therefore the current SysML tools do have deficiencies among others, in terms of usability (developed from software engineers for software engineers), interfaces to other tools and the efficient model management. In particular the tool integration and model sharing capabilities seem by far not adequate for systems engineering operational deployments.

A comparison between a typical software engineering and a system engineering operational scenario shows, that the system engineering scenario is much more complex in terms of involved users, model complexity, number of interfaces and number of different tools. The further the concept of MBSE succeeds, the more important is a seamless integration of the different tool (instances). For a successful application this eventually exceeds the scope of one particular tool. Rather the assembly of the different tools, need to be considered as a system on its own.

It is a widely spread misconception with UML tool, that the languages, respectively the tools come along with the required semantics. In fact, a successful integration of a UML tool requires a specific UML profile for the particular purpose. Here the end-user have to care for the process and refine the tools according to the modeling problem. This typically requires a profound understanding of the modeling task and the tools being used. It falls into the responsibility of the expert of the application domain to care for the customization. A key element in this is the user driven formaling modeling of the process, comprising the engineering tasks as well as in particular the modeling concepts in terms of an conceptual model or ontology.

Way forward

A number of initiatives are under way today to evolve and promote the use of MBSE methods and tools in early project phases and during verification activities. MBSE is the focus of a multidisciplinary research-and-development initiative called *Virtual Spacecraft Design* under ESA contract in the frame of its Technology Research Programmed and General Support Technology Programmed. It is part of a three-year plan (2008–2010), which is expected to yield tangible results in the near future.

Profile

Ralf Hartmann has been an INCOSE member since 1996 and was a founding member of GfSE, the German Chapter of INCOSE. Currently he is INCOSE Director for Strategy having taken up the role in mid 2008.

Ralf has been President of GfSE, Chair of the 2nd European Systems Engineering Conference (EuSEC) in 2000 and Co-chair of the International Symposium 2004 in Toulouse. Within INCOSE he has held several positions such as co chair of the Standards Committee and the Modelling and Tools Technical Committee and he is the CAB representative for Astrium Satellites. Ralf was the principal INCOSE delegate for the global GEOSS initiative from the day when INCOSE joined as a participating organisation until he took over responsibility as Director for Strategy. In 2005 Ralf was selected as an INCOSE fellow and he received the INCOSE Founders Award in 2008. Among his technical contributions two best papers at symposia and his contribution to the Systems Engineering Vision 2020 may be highlighted.

Ralf Hartmann is currently a Vice President for Control Ground Systems, Engineering Tools, and Satellite Functional Verification Infrastructure at Astrium Satellites within EADS. Since 1987 he has worked for Astrium and its predecessor companies in the area of robotics, automatic control, simulation, S/W development and systems engineering. During his career he has held various technical and project management positions as well as senior management responsibilities. Today he is also a member of the EADS Systems Engineering Steering Group and the Engineering Board within Astrium Satellites.

Highlights in Ralf's career include the implementation of a comprehensive Systems Engineering Qualification Program and the implementation of the Satellite Design Office, a conceptual design centre, which he led for some years. Furthermore Ralf was one of the key authors of the current European Space Systems Engineering Standard (ECSS-E10).

He received a Diploma in Electrical Engineering from the University of Karlsruhe and he is a certified project management professional.