# Collaborating on MBSE Usability

Archer, Lempia, Lyells, Workinger…

May, 2011

# Overview

This document proposes a way for the MBSE Usability Team to finesse resource issues to achieve its mission by:

* Facilitating the development of an MBSE Community of Practice
* Developing an Internet-Based collaboration site for the MBSE-CoP
* Leading in the identification, development and publication of exemplars of best practice for MBSE-based development

Core activities of this effort include detailed use case development, as previously planned; however, this use case effort is viewed from the perspective of using the initial detailed use cases as exemplars for the larger community. Fundamentally, the proposed approach is about working smarter to make the most of our limited resources and about creating an environment where an extended MBSE Community of Practice creates value in unanticipated ways.

# The Context

The INCOSE Model Based Systems Engineering Initiative is a collaborative effort to facilitate development of MBSEs. Meeting usability challenges is a key aspect of the initiative. More specifically, the mission of the MBSE usability team is to facilitate the growth of MBSE environments that are 1) easy to learn, 2) efficient to use, 3) have structures and processes that are easy to remember, 4) are easy to use accurately, and 5) produce high satisfaction among their users. These five aspects of usability imply the presence of a broad range of capabilities that make an MBSE environment ‘fit for intended use’.

# The Problem

From a usability perspective, the development of fully capable MBSE environments presents unique challenges to engineering organizations, in part, due to the wide array of capabilities needed for by MBSE environments, including:[[1]](#footnote--1)

* Represent engineered systems with accurate descriptive models
* Simulate system behavior accurately
* Support analysis of systems and alternatives
* Represent complex systems, accurately
* Support the collaboration of distributed teams with an integrated modeling environment that enables:
	+ Rapid and effective teaming
	+ Efficient use of engineering labor
	+ Rapid, affordable, and accurate model development
* Support engineering process integration throughout the entire system lifecycle including long term capability engineering/re-engineering and reuse.
	+ Make sure that processes undertaken in each phase of the lifecycle of a system can use data generated by processes in the previous phases.
	+ Make design and development information available to system operators without necessarily being able to anticipate all of the operational needs for such information.
* Support integration across all activities and functions - The work is in the network.
* Support integration across tasks with seamless data flow. - Data is created once and used wherever needed.
* Support lossless integration across organizational and tool boundaries
* Integrate the systems engineering modeling environment and specialized domain engineering modeling and simulation environments
* Describe and navigate large systems, accurately
* Support integration with testing and evaluation that allows efficient and accurate verification and validation
* Support visibility of underlying assumptions, range of validity and accuracy, status of validation, stochastic ‘reality’
* Capture design rationale

***The above points have two key themes:***

* ***Over half of the above bullet points refer to aspects of integration.***
* ***Over half of the above bullet points refer to collaboration.***

This emphasis is consistent with the analysis of the high value use cases that were developed during the 2011 INCOSE International Workshop and the challenges for MBSEs posed by Michael Ryschkewitsch, NASA’s Chief Engineer. It’s particularly notable that the content of the high priority use cases suggests that most users expect integration and collaboration support that are largely unavailable in today’s MBSE environments.

The three dimensions of MBSE implementations are described as tool, model, and process.[[2]](#footnote-0) MBSE users rely upon competent implementation of all three aspects. However, individual stakeholders tend to have different roles and access for MBSE development and use and tend to contribute to different degrees along the three dimensions:

* Tool vendors create and enhance a host of proprietary modeling and simulation tools with varying degrees of openness to integration.[[3]](#footnote-1) Integration typically requires significant resources from tool vendors, even when a vendor is basing integration efforts on open standards.
* To the extent that usability evaluation and testing are occurring now, tool vendors conduct these activities on their own products.[[4]](#footnote-2) Even relatively small tests of usability with seriously limited scope need significant resources.[[5]](#footnote-3)
* Various groups of language developers have created a wide array of modeling languages, each language having groups of devoted users that depend upon the language’s distinct functionality, representational and analytical capabilities. SysML is a notable offering in this area, but so are MODELICA, AADL and SIMULINK.[[6]](#footnote-4) Today’s MBSE environments are typically fragmented.
* The model objects and design patterns developed and applied by various engineering organizations are often uniquely tailored to the types of systems under development. The specific character of such objects and patterns cover a wide range of systems from buildings to avionics to chemical process plants, to supply chains.[[7]](#footnote-5)
* Processes differ from organization to organization. There is no ‘one size fits all’ systems engineering lifecycle and individual processes are routinely tailored to the situation for development programs. Today, systems engineering is under considerable pressure to handle systems of greater size, complexity, and flexibility. A variety of systems engineering processes have been applied that go beyond the classic ‘top-down’ and V-Models. Examples include spiral development, agile systems engineering, lean systems engineering, capability engineering and systems of systems engineering. Bottom-up and hybrid processes such as design for emergence are increasingly common. Systems engineers are responsible for choosing processes that best support their situation.[[8]](#footnote-6)
* Systems engineering processes are inherently collaborative. Collaboration support for Model Based Systems Engineering is idiosyncratic to individual organizations, typically limited in scope, and not well supported in most modeling tools.

Given the scope of MBSE application and the scope of the challenges, it appears that:

* There will be no single “one size fits all’ Model Based Systems Engineering Environment. There will be many MBSEs of varying configurations, depending upon the needs of the organizations where they are used.
* No single organization, government, industry, or academic, has all of the material and intellectual resources required to develop all the necessary capabilities for MBSEs.
* No single individual or organization will develop all of the insights necessary to develop the MBSEs of the future.
* Collaboration support is crucial to effective use of MBSEs within distributed engineering organizations, but it is still largely missing from existing MBSE environments.

Overall, the vision of developing MBSE environments that meet the needs of today’s engineering organizations transcends the material and intellectual resources of any single organization. No small team of volunteers can reasonably be expected to supply and wield the resources to address these challenges, directly.

# Proposed Solution:

As a volunteer organization with modest resources, the MBSE usability team needs to finesse resource issues by serving as a catalyst and a facilitator for MBSE development on a large scale using Internet-based collaboration processes.

The key issue is ‘How can the MBSE usability team support such large scale collaboration?’ It is suggested that the team can support the efforts of the MBSE community in four ways by:

1. Facilitating an MBSE community of practice. (MBSE-CoP)
2. Providing a virtual “meeting place[[9]](#footnote-7)” for the MBSE community of practice (MBSE CoP)
3. Managing a set of shared MBSE resources for systems engineers and MBSE developers.
4. Focusing attention on exemplars of MBSE tools, models, and processes.

For the virtual meeting place to work it is necessary that potential participants understand how to:

* Benefit – Education, access to resources, problem solving with colleagues in the CoP
* Contribute – Participants will need to know how to share insights and artifacts that they create
* Communicate – Dialog

Ideally, the virtual meeting place would, itself be an exemplar of an MBSE that supports the natural synergies of collaboration, provides a minimum of necessary structure and allows that structure to evolve agilely as insights emerge and are applied to the project.[[10]](#footnote-8)

It is proposed that the collaborative framework (wiki) contain artifacts such as:

* Tool Information
	+ Pointers to websites for tool companies
	+ Examples that are constructed to demonstrate tool features
	+ Freebees from tool vendors
	+ User reported experiences with tools and associated dialog
* Engineering Examples[[11]](#footnote-9)
	+ Pointers to the work of the MBSE challenge teams.
	+ Use Cases – The 2011 INCOSE International Workshop produced a set of preliminary use cases. However, these preliminary use cases were high level scenarios or what an agile developer might call “user stories.”[[12]](#footnote-10)
		- High Level Scenarios – Some of the use cases from the workshop contained high level usage scenarios.
		- User Stories – Many of the use cases from the workshop consisted of user stories without a step-by-step sequence of operations.
		- Detailed Use Cases – Detailed use cases contain a step-by-step exposition of a use case with specific references to objects/features/attributes that are manipulated at each step in the use case.[[13]](#footnote-11)
	+ Models will include:
		- Model examples that are complementary to specific use cases.
		- Model Exemplars – These are outstanding examples contributed by individuals engineers for the purpose of sharing with colleagues.[[14]](#footnote-12)
		- Architectural and Design Patterns – These are patterns in the tradition of Chris Alexander, Erich Gamma, and Robert Cloutier. Patterns are re-used by architects and design engineers to solve recurring architecture and design problems.
	+ Views and Viewpoints will include:[[15]](#footnote-13)
		- Standard views that are used in particular architectural and design contexts. Examples include:
			* The nine diagram types from SysML represent nine types of views.
			* The views developed by Forrester for System Dynamics are often used for systems analysis.
			* Domain specific views are common in engineering practice for specific types of systems. Domain-specific views often contain a specific scope, representational approach and perspective. For example, in a power plant, there is a standard scope for a boiler feedwater system.
		- Lifecycle View
			* Ex: Organizing the use cases from the workshop according to the INCOSE lifecycle model. (Or the DoD 5000 model, or the NASA lifecycle model…)
		- Innovative views that have been found to be useful and are candidates for wider usage.
	+ Recordings of MBSE usage:
		- Videos
		- Recordings that are generated within an MBSE tool.
* Virtual Whiteboards – As in an agile development environment, virtual whiteboards will be used for collaboration. Contributors will be able to add an item to a virtual whiteboard, select an item to work on, discuss various items, and volunteer to work on an item from the whiteboard. Two virtual whiteboards are envisioned.
	+ Development Agenda - Contributors will be able to select items from the agenda, volunteer to work on them, and implement them. Teaming with others is encouraged.
	+ Issues List – Contributors will be able to pose and discuss MBSE issues. Many issues will be explicitly tied to artifacts.
* Analyses
	+ Formal – These are expected to include:
		- Reports on usability tests for specific MBSEs implemented with particular Tools/Languages/Processes.[[16]](#footnote-14)
		- Survey Data[[17]](#footnote-15)
		- Exemplars of engineering analysis using an MBSE.
	+ Informal –
		- Lessons Learned – These will be based upon experiences contributed by MBSE users and developers.
		- Ratings – There will be a place for contributors to comment on experience with artifacts posted on the site and to rate the success of using the artifact for particular purposes.
* Processes –
	+ MBSE Development Process Exemplars, including:
		- Ron’s epiphany – Define all of the objects and then connect them.
	+ Systems Engineering Processes from an MBSE perspective
		- INCOSE standard
	+ Usability Evaluation Processes for MBSE Tool/Model/Process
		- Ex: The AMBR process, generalized.
		- Usability Inspection techniques
		- …
* Dialog & Discussion – Contributors will be able to have ongoing dialog about specific artifacts such as use cases and model examples. There will also be support for discussing higher-level issues such as integration strategy. In general, each artifact will have an associated discussion thread.

Role of the Core MBSE-Usability Team

If we use a conventional approach, the overall scope of activity envisioned is considerably beyond the resource-time available from the MBSE usability team, as it exists, today and for any time in the foreseeable future. This can be finessed by relying upon collaboration with the larger MBSE development community and focusing on two roles for the team:

* Developing and encouraging the development of exemplars
* Creating and managing the collaboration environment

For example, it would be useful to have detailed use cases covering all the key MBSE usage situations. However, it takes significant effort to create even one detailed use case and many variations are possible. Solution: Select a highly leveraged use case and develop it as an exemplar then encourage development of other use cases with the original exemplar providing guidance. For instance:

* The most commonly cited and highly rated use case from the workshop was developing system architecture using object libraries. Choosing this as an examplar:
	+ Lead development of a detailed use case for architecture using a library of objects. (Make sure that the detailed example, as developed, is sufficiently excellent to serve as an exemplar for other similar efforts.)
	+ Lead development of the model that is complementary to the use case.

Exemplars that would have high leverage include:

* Key use cases
* Key model examples
* Usability Measurement Process Examples
* Architectural Patterns

Hence, the role of the team will be to facilitate and lead by example. Exemplars will play a key role.

What we can do now

The proposed solution includes the effort we’ve been discussing for some time, including:

* Develop a detailed use case exemplar for architecture with libraries, to include:
	+ The detailed use case
	+ The library needed for the use case
	+ The model that the use case manipulates
* Develop other high value use cases as exemplars.

The proposed solution also includes additional efforts, including:

* Develop a shared vision. How close is what is described to a path forward that we can agree on? Let’s work to get on the same page.
* Presentation to the overall team. – Once we’re on the same page, we need to present whatever we come up with and get the input and involvement of the larger team.
* Design the collaboration environment. – **David**, is this your meat?
	+ Core Capabilities that include a critical mass for initial use
	+ A path toward full capabilities
* Start implementation of the collaboration environment
* Improve this document
	+ Add the lifecycle process for use cases. – **Ron**, is this your meat?
	+ ***Make sure that a future revision of this document reflects a shared vision.***

Issues

There will definitely be issues in making this work. Several come to mind:

* Getting the collaboration infrastructure to critical mass. (What would constitute critical mass in this case?)
* Attracting participation and building momentum for the collaboration
* Making the entry bar for participation low enough so that people do not balk at getting involved.
* How to design all this in such a way that people naturally add their creativity and it grows in a healthy way to go beyond the initial vision. (Design for emergence)
* Incentives for participation:
	+ The joy of participation
	+ Social
	+ Recognition – “Usability Heroes”
	+ Awards
	+ Prizes
	+ …
1. This list is partially drawn from a set of MBSE challenges presented by Dr. Michael Ryschkewitsch, NASA Chief Engineer, in his keynote speech at the 2011 Conference on Systems Engineering Research. [↑](#footnote-ref--1)
2. “Model” implies representation in one or more modeling languages. [↑](#footnote-ref-0)
3. Many widely used modeling and simulation tools still have proprietary data formats. [↑](#footnote-ref-1)
4. Usability cannot be “tested in.” Good sound design is required. However, in the MBSE area, tools are new and design is still experimental. Although usability testing is essential, practical development and experimentation cannot wait for usability testing. [↑](#footnote-ref-2)
5. See Agent-Based Modeling and Behavior Representation (AMBR), Gluck 2004 for a usability testing exemplar. [↑](#footnote-ref-3)
6. SIMULINK is a popular proprietary modeling tool and language from Mathworks. It is integrated with MATLAB, but not well integrated with SysML and many other significant components of MBSE environments. [↑](#footnote-ref-4)
7. The typical supply chain is a systems of systems. [↑](#footnote-ref-5)
8. It has been said that systems engineering needs to go beyond “the tyranny of the ‘shall’.” [↑](#footnote-ref-6)
9. A wiki seems the natural choice. Wikipedia is a natural choice as an exemplar. [↑](#footnote-ref-7)
10. Perhaps the most challenging part of this is that systems engineers are good at framing and reframing systems, including work environments. The virtual meeting place needs to have the flexibility to support multiple perspectives and refactoring. [↑](#footnote-ref-8)
11. Contributors will have the ability to modify existing artifacts, create variations of existing artifacts, define new artifacts and new types of artifacts. [↑](#footnote-ref-9)
12. A user story is essentially a promise to talk about a use case in the future with just sufficient detail to remind the team of the topic of discussion. The term “user story” came from agile programming. In an agile programming environment, user stories are typically written on index cards. [↑](#footnote-ref-10)
13. It is important to be specific about objects manipulated in an MBSE use case because of the complexity of the processes and the objects being manipulated. [↑](#footnote-ref-11)
14. Use Cases / Models – Each use case manipulates objects. Each point of manipulation in the sequence of the use case has binding(s) to specific model objects / features of objects. [↑](#footnote-ref-12)
15. Views are designed to support a set of engineering tasks. They can be extremely important for usability. A good view that displays the key relationships for an engineering problem can significantly improve both engineering efficiency and effectiveness. New types of views are often difficult to develop. It is a major achievement to develop a new view for a type of engineering activity that becomes widely used by one’s colleagues. [↑](#footnote-ref-13)
16. See the AMBR report. [↑](#footnote-ref-14)
17. See, for example the MBSE survey by Cloutier [↑](#footnote-ref-15)