Implementing Model Based Systems Engineering In South Africa

Dr. Rudolph Oosthuizen
CSIR
roosthuizen@csir.co.za

Abstract. In Systems Engineering (SE), modelling is applied to capture and represent the mental models of the systems' stakeholders. These models, consisting of various viewpoints on the system structure and behavior, can be used to derive requirements for system concept development. The INCOSE SE Vision 2020 sees Model Based Systems Engineering (MBSE) as part of a long-term trend toward a model-centric approach which is expected to replace the document-centric approach. MBSE is expected to provide significant benefits such as enhancing productivity and quality, reducing risk, and providing improved communications among the system development team. Implementing MBSE for the South African environment requires a process suited for the local environment and scale of projects. This paper builds on research and reports of MBSE implementations by the international SE community to provide a guideline for MBSE implementation in South Africa.

Introduction

Systems engineering aims to solve problems by bringing systems into being through systems thinking for understanding the part in the context of the whole. Hitchins (2008) also defines systems engineering as "… the art and science of creating whole solutions to complex problems …". Systems engineering consists of interdisciplinary activities to ensure that the stakeholders' needs are met in a cost-effective and timely manner. The basic systems engineering process distils the stakeholder needs to develop concepts, define requirements, and design and development solutions through interdisciplinary activities to ensure stakeholders' needs are met in a cost-effective and timely manner (Walden et al. 2015, Oliver et al. 1997).

The input to the systems engineering process is a need (or problem) from a stakeholder which needs to be analyzed to discover requirements for defining the purpose, objective and high-level functions of the required solution (Buede 2000, Ramos et al. 2012). The problem is analyzed through interacting with the stakeholders to ensure that the goals and objectives of the design are well understood (Walden et al. 2015, Oliver et al. 1997). However, today's problems tend to be ill-defined, with a wider impact, which may result in complexities and other environmental effects (Rittel & Webber 1973).

Holt & Perry (2008) list the three evils of systems engineering as complexity, communication, and understanding. Level of system complexity depends on the number of system elements and their interaction. An improper understanding of the problem and user needs leads to inaccurate requirements and the improper application of systems engineering. Communication problems between engineers, the development team and the stakeholders lead to interpretations of the meaning of requirements and associated models. This can be further exacerbated by poor communication between the design team and manufacturing teams. The demands on systems engineering to produce systems capable of effective operation within complex environments are ever-increasing. Modelling is one way to address the complexities and enable effective understanding and communication.

Modelling provides an iterative process to develop, use and update models to obtain insight into a complex system's behavior (Maria 1997). A model is defined as an explicit and incomplete representation or abstraction of reality to aid its description and understanding (Ramos et al. 2012). Capturing mental models of all the stakeholders in suitable models and communicate it amongst team members will improve successful problem analysis, solution concept development and solution implementation (De Weck et al. 2011).

In systems engineering, models are used to describe and represent selected aspects of the structure, behavior, operation and interfacing with other systems. Different viewpoints of the system may be structured and organized into an architecture. The conceptual models may assist design by understanding the relationship between the system as a whole and its parts to enable deriving possible emergent properties (Buede 2000, Ramos et al. 2012, Maria 1997, Walden et al. 2015).

The INCOSE Systems Engineering (SE) Vision 2020 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".

Model-Based Systems Engineering (MBSE) employs modelling languages, such as Systems Modelling Language (SysML) or Unified Modelling Language (UML), to capture, analyze, and specify the system behavior, structure, requirements, and capabilities through various consistent system views. These can be used to model complex systems and their architectures through consistent diagrams (views) of the model.

Models are also more useful than text-based systems engineering documents to develop system concepts and requirements as they support experimentation with knowledge on the problem and develop an understanding of the implications of different solutions (Estefan 2007, Walden et al. 2015).

Transforming from traditional document centric SE processes to MBSE is not trivial. Since the international INCOSE community is already performing this transition, the local transformation of South Africa also needs to be discussed. The focus of the paper is not "Why" to implement MBSE, but rather "How" in our environment.

Implementing MBSE for the South African environment requires a process suited for the local environment and scale of projects. In general, the published MBSE processes and frameworks were developed in and for first world countries with large economies, such as United States and the United Kingdom. There, projects tend to have much larger budgets and systems engineering teams. This paper discusses pitfalls experienced in the international SE community as well as guidelines on how to implement MBSE in an organization. The identified guidelines will form the basis for a MBSE Initiative (workgroup) by the South African Chapter of INCOSE. This paper forms part of exploratory research to identify topics and questions to be researched in support of MBSE implementation.

The paper will firstly provide a background on modelling and MBSE before presenting the pitfalls and guidelines derived from literature. These will be used to propose a roadmap for MBSE implementation in South Africa as well as a number of research topics.

Modelling

The purpose of modelling is to understand and describe the problem and its perceived solution sufficiently to gain insight into complex systems. A model needs to be generated for a purpose and describes the essential nature, characteristics or pattern of a process or thing without being the thing itself. Models are utilized to communicate ideas (shared vision) and experiment with knowledge on the problem and to develop an understanding of the implications of different solutions. A model requires a language to represent the perceived reality (Oliver et al. 1997, Walden et al. 2015).

A model is generally defined as an abstract description of a system based on a formal language. These descriptions are captured in a model database, which is the repository that stores all the modelling information in a structured manner so that a tool can retrieve, create or update the information (le Roux 2016). The model should be a sufficient approximation of the system it represents to incorporate most of its salient features. The model must achieve a balance between realism and simplicity, to enable understanding and simulation (Maria 1997, Gau Pagnanelli et al. 2012).

Modelling is an iterative process to develop, use and update models, from their conceptualization, using a standard, rigorous, structured methodology to obtain insight into a system's behavior as a basis for making decisions (Maria 1997). The model is constructed through schematics and network diagrams of the system to indicate how entities flow through the system. Simulations may address system functions or the detailed structure through identified scenarios (Walden et al. 2015, Maria 1997).

Various forms of models and constructs can be used to capture and represent the information and knowledge on the problem, which can be derived from the stakeholders' mental models. A mental model can be viewed as humans' interpretation of a system via their senses. The operational context and expected scenarios are defined from the initial high-level requirements to derive the required roles, tasks and functions of the system (De Weck et al. 2011).

Modelling for systems engineering has evolved from functional flow block diagrams, structure analysis and design techniques to an object oriented representation. Different views of the system may be organized in an architecture, to structure the model. Depending on the level of abstraction, and the focus of concern, there will be different viewpoints with its own unique elements that can be applied to the system (Piaszczyk 2011).

Model Based Systems Engineering

General Concepts

MBSE is defined as the formalized application of modeling to support system requirements, design, analysis, verification and validation activities throughout the system life cycle phases (Gau Pagnanelli et al. 2012). MBSE focuses on the use of information rich models instead of a document-based text documents, to complement systems engineering methodologies. Many authors suggested that models should be more useful than text-based systems engineering documents to develop system concepts and requirements. The reason being that common notations depict many system characteristics and attributes in a consistent way (Buede 2000, Ramos et al. 2012, Maria 1997, Walden et al. 2015).

MBSE employs a concurrent and incremental process to develop models to enable communication and understanding between stakeholders (Estefan 2007, Walden et al. 2015). MBSE utilizes the modelling languages of UML and SysML to develop consistent views on complex system structure, parameters, attributes, requirements, behavior and relationships. UML provides a consistent visual modelling language for specifying, visualizing, constructing, and documenting the software development artefacts. UML consists of class, use case, component, deployment, state machine, sequence, timing, activity, package, communication, composite structure, interaction overview and instance diagrams (Fowler 2004, OMG 2013, Hause 2011).

SysML is an a subset as well as an extension of UML suitable for systems engineering and include concepts such as interface and flow specifications, system concepts, parametric, and integrated requirements. SysML aims to provide a standard modelling language for systems engineering to analyze, specify, design and verify complex systems (Hause 2014). The structural diagrams represent the parts of a situation with their logical relationships. The behavioral diagrams represent the parts of a situation and their causal interactions. The requirement views specify desired structural and behavioral properties as derived from stakeholder needs. Parametric views provide the critical engineering parameters of the system for evaluating performance, reliability and physical characteristics (Friedenthal et al. 2012, Hause 2014).

The system's behavior is modelled with use case, activity, sequence and state machine diagrams. The use-case diagram provides a high-level description of the system functionality. The flow of data and control between activities are captured in an activity diagram. A sequence diagram represents the interaction between collaborating parts of a system. Actions and state transitions of a system in response to events are modelled in a state machine diagram (Hause 2014). Requirements diagrams ensure that hierarchies and the derivation, satisfaction, verification and refinement relationships are clear. System parameter constraints and other parameters such as performance, reliability and physical properties are captured in the parametric diagram (Hause 2014).

Block diagrams model the structure of the system while block definition diagrams describe the system hierarchy and classification. Internal block diagrams describe the internal structure with parts, ports, and connectors, of a system. The blocks in the block diagrams can represent any level of the system hierarchy to describe a system as a collection of parts and connections between them that enable communication and other forms of interaction. Ports provide access to the internal structure of a block for use when the object is used within the context of a larger structure (Hause 2014).

The modeler should ensure that the system stakeholders are involved from the start, and should apply methods that support mutual understanding through engagement and debate (Yearworth & Cornell 2012). Subject matter experts provide operational experience and domain knowledge to identify requirements as well as to assist in design evaluations and decisions. Scenarios are used to assess the effects and goals of a cognitive in context. They can also be used during interviews to elicit knowledge from operational stakeholders.

Through MBSE the effectiveness of SE is improved by explaining observed system behavior to support predictions for specific situations that might have been unanticipated (emergent behavior). MBSE support development of integrated, distributed architectures that would otherwise be unmanageable or error prone if managed by a document-centric SE approach. MBSE also improve productivity by reuse and reducing time, effort, or cost for a given amount of data or understanding, or reduction in errors in the SE process (Gau Pagnanelli et al. 2012).

As stated in the INCOSE 2020 vision, it is expected that MBSE will replace the document-centric approach as well as influence the future practice of systems engineering (Gau Pagnanelli et al. 2012). With MBSE information is essentially a structured model that describes the information as data with relationships to other data. Therefore, the underlying model is important to address all the aspects of SE (specification, design, integration, validation, and operation of a system) (le Roux, G., 2016). The potential benefits of MBSE above document-based processes include the following (Ramos et al. 2012, Gau Pagnanelli et al. 2012, Hallqvist & Larsson 2016, le Roux, G., 2016, Meyer 2014):

1. Information that is captured in a standardized way to improve communication among model designers and stakeholders. This will improve understanding and insight.
2. The probability of errors and ambiguity is reduced as complexity in the system can be effectively addressed.
3. One model with multiple views provide a complete and consistent representation of the system across different system missions and life cycle phases. This supports a holistic approach to capture the structure and behavior of a system to deal with their complexities.
4. Time and resources are reduced as model elements are reused in the different views and maintenance of system artefacts.
5. Time to respond to stakeholders' inquiries is decreased as new or different views of the model can be expeditiously exported.
6. Traceability and consistency checking is supported.
7. One model may contain the information from several documents.

Elements of Model Based Systems Engineering

MBSE is not only about a language, such as SysML, or a tool; it encompasses much more. The implementation of MBSE should focus on assisting systems engineers to understand and communicate the problem, not only the specific tool, or language, or approach (Ramos et al. 2012, Tschirner et al. 2015).

Any MBSE implementation should address at least the following MBSE support elements (Meyer 2014, Martin & James 1996, Tschirner et al. 2015):

Process. The process provides the logical sequence of tasks performed at various levels of detail and aggregation to achieve a particular objective. Many concepts exist for MBSE processes and need to be tailored per specific implementation environment. The process needs not be the same as for traditional SE standards (Gau Pagnanelli et al. 2012).

Method. The method consists of the set of techniques for performing a task as listed in the process. Each method may also be a process in itself. The Object Oriented Systems Engineering Method (OOSEM) integrates top-down functional decomposition with a model-based approach that uses SysML to support the specification, analysis, design, and verification of systems (Karban et al 2012).

Toolset. The tools are the instruments that enhance the efficiency of the tasks in a specific method. The tools only facilitate how the tasks are performed and provide the repository of the information captured in the model. The language for modelling (e.g. SysML) and architecture framework (MoDAF) forms part of the toolset. The toolset has to provide the following capabilities in support of MBSE (Gau Pagnanelli et al. 2012, Herzog et al. 2010):

1. An integrated data environment to facilitate data coordination among design disciplines (electrical, mechanical, aerospace, software, systems engineering) that support very large system models.
2. A multi-user environment that enables multiple engineering disciplines to work concurrently from a single authoritative data source.
3. A robust query engine that allows for rapid assessment of an integrated database.
4. A document generation capability to export artefacts for stakeholder interaction.

The dataset as captured in the tool repository is the model, artifacts are not the model. Artifacts are views of the model that aid in model comprehension. When models need to be shared, it is the dataset that needs to be shared more so than the artifacts (Malone et al. 2016).

Problems with Implementing Model Based Systems Engineering

Implementing MBSE to replace a document centric SE approach is not a trivial task. Similar to implementing organizational change, many pitfalls exist. The barriers to successful implementation of MBSE derived from literature are summarized as follows (Haskins 2011, Kass & Kolozs 2016, Tschirner et al. 2015, Voirin et al. 2015, Bonnet et al. 2015, Hallqvist & Larsson 2016, le Roux, G., 2016, Herzog et al. 2010, Morkevicius et al. 2016):

Economic. Personnel or organizational investment is required to implement MBSE. The cost of implementing MBSE may be high due to tool acquisition and training. It is often difficult to motivate these costs.

Management Commitment. Managers tend to be averse to the risks of adopting a new method. The lack of commitment from management will limit funding required for tools and training. This may derive from a lack of a common and clear view on the purpose and goals for implementing MBSE.

Cultural Change. Often existing SE processes and methods are entrenched in the organization with acceptable results. Systems engineers may be adverse to learning the new methods and modelling languages, or working with new tools. A steep learning curve on how to use a new method and tool may cause engineers to feel insecure.

Organizational Change. This include structuring of the project, management of presentation of information and management of changes to information. Sometimes no relationship exists between the different teams working on the project, causing a lack of traceability between them. The models are then difficult to maintain with duplicated or redundant information.

Too Ambitious Expectations. Due to the opportunities of modeling tools, the purpose with the model must be clearly stated and understood by all stakeholders. Disappointment when expectations are not met will derail MBSE implementations.

Technical. Technical issues include the lack of interoperability and standardization in current tools, lagging technology advancements and immaturity of products. The ability to link tools from different vendors may be difficult without standardized definitions and modeling elements. Tools need to be customized to the process of the organization.

Method. Using MBSE to execute existing SE processes and methods slow down cultural and organizational change to limit the advantages of MBSE. Also not choosing and adapting the suited methods for the specific environment limits the gains of MBSE.

Configuration Management. Proper change and version management is required in the tool and the process. Inadequate control of access to MBSE repositories may cause security breaches or damage to the work documents, system designs, or code modules stored in the repository.

Acceptance of model deliverables. MBSE outputs are not the same as document-centric approaches. The stakeholders may have difficulty accepting this in their environment and management processes. Additional training of all stakeholders on reading and interpreting diagrams of model views and other artefacts is required.

Too Complex diagrams. Complex diagrams are ambiguous to understand and difficult to communicate, often hiding issues of completeness. Engineers tend to model familiar aspects in too much detail, while overlooking others. Different views need to be identified in the method to represent the model data. A view is not only a diagram, it may include other info such as tables, charts or matrices.

B. Graphical Modeling Languages

The modeling tools, which are previously referred in Section I-B, can be classified as another group of SE standards. This group includes the common representations used to describe a system. The modeling techniques used in the field of SE, to develop systems (modeling concepts, properties, attributes, structure, behavior, entities, interactions, relations, environment, etc.), have always been predominantly qualitative and based on a graphical or pictorial representation. These techniques require a corresponding describing language (i.e., graphical modeling language or visual modeling language), used to represent reality, that involves semantics (i.e., set of symbols or signs that form the basis of representations) and syntax (i.e., the proper ways of combining the symbols and signs to form thoughts and concepts).

The functional flow block diagrams (FFBDs), which are developed in the 1950s, have been, for many years, the classical representation of SE with a wide spread use within the community. This tool illustrates a step-by-step sequence of a system’s functional flow through a functional-decomposition approach. During the 1970s, the structured analysis and design technique (SADT) emerged as the graphical language to communicate ideas [31], and to understand and describe systems as a hierarchy of functions. In 1993, the National Institute of Standards and Technology (NIST) launched the Integration Definition for Function Modeling (IDEF0), which is a graphical notation belonging to the IDEF suite of modeling approaches and derived from the SADT. This notation was developed to represent activities or processes that are carried out in an orderly manner [32], illustrating the functional perspective of a system, the data flow, and the system control.

The enhanced FFBD (EFFBD) and the IDEF0 have been the main modeling languages used in SE in the last decades. Other tools include, for example, the N2 charts, state-transition diagrams, and petri nets.

These traditional functional decomposition procedures/ representations are being “replaced” by object-oriented approaches. The modern object-oriented practices, with its roots in software engineering, are now pervasive in the systems engineering field. Oliver et al. [33] traced the origin of the objectoriented paradigm back to the 1970s, with the development of abstract data types and the introduction of classes to programming languages, like Simula67, in order to provide procedure, data, and control abstractions. In the 1990s, these principles have been extended to the analysis and design of software, through the Booch method, the object modeling technique (OMT), and then through the de facto UML [34]. The characteristics of the software systems are different from those of the systems of SE (that may also include software components), and consequently, the UMLlacks support from aspects, like the whole/part decomposition, or the interconnections provide by physical things (and not by compilers), or the trade studies.

Fig. 2. Simplified sequence diagram (sd) for the use case “measure the blood pressure,” including an weak sequencing of occurrences, two lifelines, activations, a series of synchronous, asynchronous and reply messages, an alt operand of a combined fragment for the alternative courses of action derived from the displayed message at the measurement device, and an interaction use (frame reference) that specifies an interaction described on other sd.

In order to incorporate these and other features, the OMG and the INCOSE have joined efforts and developed an extension of UML for SE: the systems modeling language (SysML) [15], which was released in 2007. This graphical SysML, which supports the specification, analysis, design, and verification of complex systems, is considered as the next de facto modeling language for SE. According to Oliver et al. [33] “SysML continues to lack a few of the needed concepts, but has extended others in useful ways beyond historic SE practice.”

The SysML-modeling tools usually store the user model as structured data in a model repository, and the model enters and retrieves that information by using the graphical representation that is, the diagrams. The SysML diagrams, which reflect various aspects of a system, are nine and are organized in four major blocks that are known as the four pillars of SysML and represent four key modeling facets: the requirements of the system, the structure, the behavior, and the parametric relationships (Fig. 2 depicts an example of an SysML diagram developed in the Artisan Studio tool). These different views match particular viewpoints (the stakeholders’ perspectives) and enable the holistic approach required by SE.

As UML, the modeling language for SE is not attached to any methodology. The SysML also supports model and data interchange via the XMI and via the evolving neutral ISO AP233 standard (this application protocol aims to support the exchange of data during the whole system development lifecycle and across different domain engineering disciplines allowing the creation of one consistent view of the system). The XMI provides interoperability capabilities such as, to export selected parts of an SysML model (in the model repository) to another UML tool in order to support software development, and to import and export parametric diagrams relating data to engineering analysis tools [15].

The unified profile for DoDAF/MoDAF (UPDM) is also an extension of UML to describe SoS and enterprise architectures compliant with DoDAF and MoDAF requirements.

Fig. 3. Example of an (left) OPD and (right) the corresponding OPL.

This profile is particularly tailored for military acquisition programs.

The OPM, which was founded by Dori in 2002, and the corresponding graphical and textual representations, object process diagrams (OPDs), and object process language (OPL), enlarge the domain of object-oriented-modeling tools for SE. The provided bimodality (i.e., graphical and textual) facilitates the understanding of complexity since it is very similar to the power of both sides of the brain, i.e., the right side that acts like the visual interpreter and the left side that acts like the language interpreter. According to Grobshtein and Dori [36], this intuitive dual notation provides a single model that is comprehensible to the different stakeholders (both technical and nontechnical) involved in the development process.

They are available at the software environment object process CASE tool (OPCAT). According to Booch et al. [34], OPM "is a comprehensive novel approach to SE. Integrating function, structure, and behavior in a single, unifying model, OPM significantly extends the system modeling capabilities of current object-oriented methods." Fig. 3 provides an example of anOPD and the corresponding OPL.

The OPM is based on three fundamental aspects of a system: the structure (how it is made), the function (what it does), and the behavior (how it changes over time). The function is enabled by the architecture of the system that combines the structure and the behavior. The graphics (i.e., OPDs) and the natural language (i.e., OPL) express these characteristics in a unified frame of reference that corresponds to an integrated single model.

The SysML and the OPDs/OPL constitute the current stateof-the-art systems modelling languages. SysML being a more "institutionalized/standardized" language with the support of the OMG and the INCOSE, and the OPDs/OPL a more intuitive simpler language with less training effort, it seems interesting to combine the advantages of both languages creating synergies between them [36]. This integration can strongly contribute to a common understanding of the system and to improved communications between different stakeholders, as well as to a proficient SE collaborative development environment.

According to Wilkiens [37] there will be, in the future, a great demand to model languages since systems will become increasingly complex and there are considerable advantages in modeling and simulating before using them in practice. The author synthesizes the advantages of the modeling languages with the following idea: "The modeling language allows me to move on different abstraction levels. The more abstract I get the simpler the system appears to be. This is the art of being concrete on an abstract level."

(Ramos et al. 2012)

Model Based Systems Engineering Implementation Guidelines

Despite the pitfalls listed and discussed above, implementing MBSE within an organization can still be successful. Hallqvist & Larsson (2016) encourage a process of "Think Big - Start Small – Evolve". Due to the large amount of uncertainty, success is enhanced by introducing small steps at a time. However, the steps should be aligned and aimed towards a known end goal. Also, an MBSE environment is, in itself, a system, and as such needs to be architected using Systems Engineering (SE) processes (Malone et al. 2016).

To facilitate a successful MBSE implementation, the following guidelines were suggested by various authors (also note the similarity to the pitfalls identified) (Ramos et al. 2012, Gau Pagnanelli et al. 2012, Voirin et al. 2015, Bonnet et al. 2015, Hallqvist & Larsson 2016, Malone et al. 2016, le Roux, G., 2016, Herzog et al. 2010, Karban et al 2012, Bayer et al. 2013, Morkevicius et al. 2016):

Modelling Purpose and Scope. Carefully define modeling purposes, objectives, and scope. Use specific model types to address specific problems. Refrain from modelling for the sake of modelling. The aim should be to tell the story of the system over its lifecycle, as relevant to the project. MBSE success depends on the ability of the model to satisfy the analytical and documentation demands of the stakeholders.

Senior Management Commitment. Ensure strong sponsorship, investment and commitment from senior management. Management must understand the advantages of MBSE and be willing to change the structure and organization of the SE teams. Industry requires more 'success stories' and evidence that MBSE resolves the uncertainty of complex environments to overcome psychological barriers to any change.

Champions. Along with senior management commitment, it is crucial to identify MBSE Champions. These engineers need to have a strong modelling background, and should be tool experts and be the early adopters. They play the role of an evangelist.

Quick Implementation. Implement MBSE on less complex and smaller pilot projects and gradually introduce the tools to other programs. The risk is too high on projects with short deadlines. This will enable systems engineers to gain confidence, learn the languages and fine-tune methods before committing to complex and high risk projects. It may be useful to implement an MBSE process to develop a MBSE process for implementation in the organization.

Cultural Change. Support cultural change by organizational change and continuous improvements principles. Implementing cultural change requires all stakeholders, not only the SE organization, to work together for the same goal. This will be supported by promoting and demonstrating the expected benefits of MBSE.

MBSE Methodology. Establishing a well-defined adequate and customized MBSE methodology is key to success. The methodology needs to support a migration from document centric approaches. Although several suitable MBSE methodologies are available (e.g. OOSEM, Harmony, SYSMOD, FAS, SIMILAR), only few are genuinely targeted to systems engineering on a specific project. Existing methodologies is only the starting point for a transformation to save on time and resources. Due to multiple MBSE paradigms available for implementation, evaluation and selection criteria must be established based on process effectiveness, specific model requirements, and suitability considerations within the organizational context. The MBSE techniques need to be intelligent and intuitive to help address increasing complexity of systems with integrated people, technologies, hardware, software, processes, and enterprises. Set clear modeling objectives that identify model inputs and outputs. Typical aspects to be addressed include the following:

1. The process for eliciting information for the models.
2. Selection of a modelling language.
3. Choice of architecture framework to guide structuring of models.

4. Definition of views to be built for representing the information the model.
5. Definition of artifacts to deliver to the stakeholders.
6. Definition of model validation and verification processes.
7. Allow models and the process to evolve.

MBSE Tools. Ensure that MBSE tools (tool chains) for modelling and management of model information are adequate and customized to the requirements of the methods implemented in the organization. Data model extensibility as well as import and export utilities are essential to the success of the MBSE environment. The document is still the primary artifact for communication with stakeholders. A consequence of this is that the model may contain information that will not end up in the official documentation.

Modelling Language. The best suited modelling language for the organization and its operational environment needs to be selected. It is also essential the adequately train modelers and other stakeholders in modelling languages.

MBSE Best Practices. MBSE best practices, which include modelling guidelines, need to be identified and promoted. These address the layout of diagrams, naming conventions, expected diagrams and views. These guidelines should be captured in an easily accessible model management plan or wiki for team members. Establishing a center of excellence will also assist in promoting and communication MBSE best practices.

Mentorship. Provide continuous mentorship to MBSE practitioners. This goes beyond training to include disseminating experience. The mentors should promote the use of modeling tools and interoperability standards. Another option is to establish a community of users sharing experiments, complementary tools and practices with regular tool user group meetings

Develop Suitable Metrics. Suitable metrics and a value model for MBSE will be used to manage effective modelling through validation. Implementation of methods and tools need to be validated before using them on a project. The requirements that contribute to the success in the areas of effectiveness, suitability, and model include:

1. Product line and life cycle characteristics.
2. Integration complexity.
3. Model completeness and correctness.
4. Modeling productivity (initial and changes).
5. Consistency with organization processes and practitioner expertise.
6. Computing/data/tool infrastructure.

Communication with Stakeholders. The MBSE practitioners must establish interaction with stakeholders using the model as reference to gain their acceptance. Interaction includes regular model reviews, which also involve domain experts. The stakeholders (modelers, clients and users) need to agree on the format of contracting, deliverables, quality control and configuration management. Although SysML is fairly large, rich, and comprehensive, appropriate to provide a detailed description of the system, and uses a standard notation that is supported by several commercial tools, but is cumbersome and requires significant learning efforts. Nontechnical stakeholders are often not able to work with this language.

South African MBSE Initiative

Supporting the implementation of MBSE in South Africa require support from INCOSE and academia. For this reason, the South African MBSE Initiative needs to be started. The aim should be to coordinate and support efforts to implement MBSE. This will build on knowledge and support from the international INCOSE MBSE initiative. The work and insight gained will be shared at the INCOSE International Workshop. Participation in the South African MBSE

Initiative by the industry and academia will be encouraged. The specific objectives of the South African MBSE Initiative will be the following:

1. Facilitate training in MBSE methodologies and modelling languages (i.e. SysML).
2. Identify the organizations (industry) that implement or plan to implement MBSE.
3. Set MBSE and specific modelling guidelines.
4. Identify the MBSE tools that are used to support the modelling of systems.
5. Identify the MBSE outputs generated as part of the SE process as well as how they are utilized by the stakeholders.

Typical MBSE Research Required

Outputs from the South African MBSE Initiative will be used to guide SE research in academic institutions. This paper already identified possible research issues and questions to be addressed. Typical research topics include the following:

1. Determine the RSA academic institutions that teach MBSE and/or any of its modelling languages as part of a curriculum.
2. Determine how to tailor MBSE processes and methods for modelling of a system in a specific environment.
3. Determine methods to address complexity in systems (and systems of systems) and integrate them into the MBSE toolset. This may include other modelling approaches such as Agent Based Modelling and System Dynamics.
4. Determine verification and validation processes for tailoring of methodologies and tools as well as the output (product) models of an MBSE implementation.
5. Assess the perception on what is MBSE and at what level it is performed. Many systems engineers or other engineers think they do MBSE, but is not aware of all the aspects thereof.

Conclusion

As discussed in this paper, systems engineering is seen as an established approach to develop systems. Modelling is applied to capture and represent the mental models of the systems' stakeholders. These models, consisting of various viewpoints on the system structure and behavior, can be used to derive requirements for system concepts and actual system development.

MBSE focuses on models to support the SE process through the entire life cycle of a system. Implementing MBSE will lead the way away from document centric to model centric SE. This will improve the understanding and implementation of complex systems. However, despite all the advantages of MBSE, many pitfalls exist. Experience from international implementation of MBSE also helped to identify guidelines to overcome the pitfalls to enable successful implementation.

Implementation of MBSE in the South African environment requires a process suited for the local environment and scale of projects. Research and guidance is required to adapt processes, methods and tools successfully. Therefore, an South African MBSE Initiative is required to guide and coordinate efforts between industry and academia. Research on aspects of MBSE implementation is required to build a knowledge base on customization of processes, methods and tools.

References

Bayer, T., Chung, S., Cole, B., Cooke, B., Dekens, F., Delp, C., Gontijo, I. and Wagner, D., 2013, March. Update on the model based systems engineering on the europa mission concept study. In Aerospace Conference, 2013 IEEE (pp. 1-13). IEEE.

Bonnet, S., Voirin, J.L., Normand, V. and Exertier, D., 2015, October. Implementing the MBSE Cultural Change: Organization, Coaching and Lessons Learned. In INCOSE International Symposium (Vol. 25, No. 1, pp. 508-523).

Buede, D.M. 2000. Engineering Design of Systems - Models and Methods. John Wiley & Sons.

De Weck, O.L., Roos, D, and Magee, C.L., 2011. Engineering systems: Meeting human needs in a complex technological world. MIT Press.

Estefan, J. 2007. Survey of Model-Based Systems Engineering (MBSE) Methodologies. Jet Propulsion Laboratory, California Institute of Technology, Pasedena, CA.

Fowler, M., 2004. UML distilled: a brief guide to the standard object modeling language. Addison-Wesley Professional.

Friedenthal, S., Moore, A. and Steiner, R., 2012. Integrating SysML into a systems development environment a practical guide to SysML. Elsevier, pp.523-556.

Gau Pagnanelli, C.A., Sheeley, B.J. and Carson, R.S., 2012, July. Model-Based Systems Engineering in an Integrated Environment. In 22nd Annual INCOSE International Symposium. Rome, Italy.

Hallqvist, J. and Larsson, J., 2016, July. Introducing MBSE By using Systems Engineering Principles. In INCOSE International Symposium (Vol. 26, No. 1, pp. 512-525).

Haskins, C., 2011, June. 4.6. 1 A historical perspective of MBSE with a view to the future. In Incose international symposium (Vol. 21, No. 1, pp. 493-509).

Hause, M. (2011, June). 4.6. 2 “Are we there yet?” Assessing Quality in Model Based Systems Engineering. In INCOSE International Symposium (Vol. 21, No. 1, pp. 510-522).

Hause, M. C. (2014, March). SOS for SoS: A new paradigm for system of systems modeling. In Aerospace Conference, 2014 IEEE (pp. 1-12). IEEE.

Herzog, E., Andersson, H. and Hallonquist, J., 2010, July. 4.5. 3 Experience from Introducing SysML into a Large Project Organisation. In INCOSE International Symposium (Vol. 20, No. 1, pp. 595-604).

Hitchins, D. K. 2008. Systems Engineering: A 21st Century Systems Methodology. Wiley. com.

Holt, J. and Perry, S., 2008. SysML for systems engineering (Vol. 7). IET.

International Council on Systems Engineering, “Systems Engineering Vision 2020,” INCOSE-TP-2004-004-02, Version/Revision: 2.03.

Karban, R., Zamparelli, M., Bauvir, B. and Chiozzi, G., 2012. Three years of MBSE for a large scientific programme: Report from the Trenches of Telescope Modeling. In Proceeding 22nd Annual INCOSE International Symposium.

Kass, N. and Kolozs, J., 2016, July. Getting Started with MBSE in Product Development. In INCOSE International Symposium (Vol. 26, No. 1, pp. 526-541).

le Roux, G., 2016, July. Model Base Systems Engineering used in developing a Telescope Manager. In INCOSE International Symposium (Vol. 26, No. 1, pp. 833-847).

Malone, R., Friedland, B., Herrold, J. and Fogarty, D., 2016, July. Insights from Large Scale Model Based Systems Engineering at Boeing. In INCOSE International Symposium (Vol. 26, No. 1, pp. 542-555).

Maria, A., 1997. Introduction to Modeling and Simulation. In Proceedings of the 29th conference on Winter simulation. IEEE Computer Society, 7-13.

Martin, James N., Systems Engineering Guidebook: A Process for Developing Systems and Products, CRC Press, Inc.: Boca Raton, FL, 1996.

Meyer, B., 2014, October. Generic Model‐Based Systems Engineering Methodology. In INCOSE International Symposium (Vol. 24, No. s1, pp. 207-222).

Morkevicius, A., Bisikirskiene, L. and Jankevicius, N., 2016. We Choose MBSE: What’s Next?. In Complex Systems Design & Management (pp. 313-313). Springer International Publishing.

Oliver D.W., Kelliher T.P., & Keegan Jr. J.G. 1997, Engineering Complex Systems with Models and Objects. McGraw-Hill.

OMG, 2013. Unified Profile for DoDAF and MODAF (UPDM), Version 2.1.

Piaszczyk, C., 2011. Model based systems engineering with Department of Defense Architectural Framework. Systems Engineering, 14(3), pp.305-326.

Ramos, A.L., Ferreira, J.V. and Barceló, J., 2012. Model-based systems engineering: An emerging approach for modern systems. IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews), 42(1), pp.101-111.

Rittel, H.W. and Webber, M.M., 1973. Dilemmas in a general theory of planning. Policy sciences, 4(2), pp.155-169.

Tschirner, C., Dumitrescu, R., Bansmann, M. and Gausemeier, J., 2015, April. Tailoring Model-Based Systems Engineering concepts for industrial application. In Systems Conference (SysCon), 2015 9th Annual IEEE International (pp. 69-76). IEEE.

Voirin, J.L., Bonnet, S., Normand, V. and Exertier, D., 2015, October. From initial investigations up to large‐scale rollout of an MBSE method and its supporting workbench: the Thales experience. In INCOSE International Symposium (Vol. 25, No. 1, pp. 325-340).

Walden, D.D., Roedler, G.J., Forsberg, K., Hamelin, R.D. and Shortell, T.M. eds., 2015. Systems engineering handbook: A guide for system life cycle processes and activities.

Yearworth, M., & Cornell, S. 2012. Contested Modelling. In Proceedings of the 56th Annual Meeting of the ISSS-2012, San Jose, CA, USA.