Model Based Systems Engineering (MBSE) Media Study

Prepared by: Julia Murray

May 2, 2012
TABLE OF CONTENTS

1.0 MODEL-BASED SYSTEMS ENGINEERING (MBSE) INITIATIVE ...................................................... 5

2.0 MBSE DEFINITIONS & ADVANTAGES .................................................................................. 6
  2.1 INTERNATIONAL COUNCIL ON SYSTEMS ENGINEERING (INCOSE) DEFINITION & ADVANTAGES .... 6
    2.1.1 Model-Based Systems Engineering (MBSE) Definition .......................................................... 6
    2.1.2 Advantage to using MBSE .................................................................................................... 7
  2.2 SYSML FORUM MBSE DEFINITION & ADVANTAGE ............................................................ 8
    2.2.1 SysML Forum MBSE Definition ............................................................................................ 8
    2.2.2 SysML Forum MBSE Advantage ........................................................................................... 8

3.0 INCOSE SYSTEMS ENGINEERING VISION 2020 ............................................................... 9
  3.1 VISION 2020 INTEGRATING FRAMEWORK .......................................................................... 9
  3.2 SYSTEMS ENGINEERING VISION .......................................................................................... 10
  3.3 ROLE OF MBSE WITHIN SYSTEMS ENGINEERING VISION 2020 ....................................... 12

4.0 MBSE METHODOLOGIES ....................................................................................................... 14
  4.1 INCOSE OBJECT-ORIENTED SYSTEMS ENGINEERING METHOD (OOSEM) ................. 14
  4.2 IBM RATIONAL TELELOGIC HARMONY-SE .................................................................... 16
  4.3 IBM RATIONAL UNIFIED PROCESS FOR SYSTEMS ENGINEERING (RUP-SE) ............. 17
  4.4 JPL STATE ANALYSIS (SA) ................................................................................................... 19
  4.5 VITECH MODEL-BASED SYSTEMS ENGINEERING (MBSE) METHODOLOGY .............. 20
  4.6 DORI OBJECT-PROCESS METHODOLOGY (OPM) ............................................................. 22
  4.7 WIEKIEKS SYSTEMS MODELING PROCESS (SYSMOD) ................................................... 23
  4.8 FERNANDEZ ISE & PROCESS PIPELINES IN OO ARCHITECTURES (ISE&PPOOA) ......... 24

5.0 MBSE MODELING LANGUAGE STANDARDS ......................................................................... 26
  5.1 OMG SysML™ ...................................................................................................................... 26
  5.2 UML ....................................................................................................................................... 28

6.0 MBSE SOFTWARE TOOLS ....................................................................................................... 29
  6.1 MBSE TOOLS AND RELEVANCE OF OTHER “DE-FACTO” MBSE VISUAL MODELING STANDARDS ... 29
  6.2 SAE ARCHITECTURE ANALYSIS & DESIGN LANGUAGE (AADL) .................................. 29
  6.3 VITECH CORE ....................................................................................................................... 30
  6.4 MAGICDRAW® ...................................................................................................................... 31
    6.4.1 Related MagicDraw Products and Plugins ............................................................................ 32
  6.5 PHOENIX INTEGRATION ...................................................................................................... 32
    6.5.1 Design Space Navigation/Optimization & Variable Sensitivity Analysis ......................... 33
    6.5.2 ModelCenter ...................................................................................................................... 33
    6.5.2.1 Benefits ............................................................................................................................ 34
  6.6 IBM® RATIONAL® RHAPSODY® .......................................................................................... 34
  6.7 ECLIPSE .................................................................................................................................. 35
    6.7.1 What is Eclipse and the Eclipse Foundation? .................................................................. 35
    6.7.2 A Unique Model for Open Source Development ........................................................... 36
    6.7.3 About Papyrus .................................................................................................................. 36
  6.8 ISIGHT & THE SIMULIA EXECUTION ENGINE .................................................................... 36
    6.8.1 Dassault Systemes Company Information ..................................................................... 36
    6.8.2 SIMULIA Overview .......................................................................................................... 36
    6.8.3 Isight & SEE Overview ...................................................................................................... 37

7.0 COMPANIES WORKING TO IMPLEMENT MBSE .................................................................. 38
  7.1 LOCKHEED MARTIN .............................................................................................................. 38
  7.2 DEERE & CO .......................................................................................................................... 38
  7.3 ROCKWELL COLLINS .......................................................................................................... 38
<table>
<thead>
<tr>
<th>Page</th>
<th>Content</th>
</tr>
</thead>
<tbody>
<tr>
<td>44</td>
<td>DOD SYSTEMS 2020 VISION</td>
</tr>
<tr>
<td>41</td>
<td>UNIVERSITIES</td>
</tr>
<tr>
<td>40</td>
<td>U.S. ARMY RESEARCH, DEVELOPMENT AND ENGINEERING COMMAND</td>
</tr>
<tr>
<td>39</td>
<td>NORTHRUP GRUMMAN</td>
</tr>
<tr>
<td>39</td>
<td>MODERN TECHNOLOGY SYSTEMS INC</td>
</tr>
<tr>
<td>39</td>
<td>NORTHROP GRUMMAN</td>
</tr>
<tr>
<td>41</td>
<td>TECHNION ISRAEL INSTITUTE OF TECHNOLOGY</td>
</tr>
<tr>
<td>42</td>
<td>MASSACHUSETTS INSTITUTE OF TECHNOLOGY</td>
</tr>
<tr>
<td>42</td>
<td>STEVENS INSTITUTE OF TECHNOLOGY</td>
</tr>
<tr>
<td>43</td>
<td>TECHNICAL UNIVERSITY OF DEMARK</td>
</tr>
<tr>
<td>44</td>
<td>DOD SYSTEMS 2020 VISION</td>
</tr>
<tr>
<td>45</td>
<td>INTEGRATED GAME-CHANGING SOLUTION STRATEGY</td>
</tr>
<tr>
<td>46</td>
<td>9.2.1 Model Based Engineering (MBE)</td>
</tr>
<tr>
<td>47</td>
<td>9.2.2 Platform Based Engineering (PBE)</td>
</tr>
<tr>
<td>47</td>
<td>9.2.3 Capability on Demand (COD)</td>
</tr>
<tr>
<td>48</td>
<td>9.2.4 Trusted Systems Design (TSD)</td>
</tr>
<tr>
<td>49</td>
<td>9.2.4.1 Related to DoD Systems 2020 Initiative</td>
</tr>
<tr>
<td>50</td>
<td>LEAN ENGINEERING</td>
</tr>
<tr>
<td>50</td>
<td>LEAN IMPLEMENTATION DEVELOPS FROM TPS</td>
</tr>
<tr>
<td>50</td>
<td>DIFFERENCES FROM TPS</td>
</tr>
<tr>
<td>51</td>
<td>LEAN GOALS AND STRATEGY</td>
</tr>
<tr>
<td>52</td>
<td>STEPS TO ACHIEVE LEAN SYSTEMS</td>
</tr>
<tr>
<td>52</td>
<td>10.4.1 Design a simple manufacturing system</td>
</tr>
<tr>
<td>52</td>
<td>10.4.2 There is always room for Improvement</td>
</tr>
<tr>
<td>52</td>
<td>10.4.3 Continuously Improve</td>
</tr>
<tr>
<td>53</td>
<td>HOW CAN LEAN CONNECT WITH MBSE</td>
</tr>
<tr>
<td>54</td>
<td>LOOK AHEAD WHERE WE ARE GOING</td>
</tr>
<tr>
<td>54</td>
<td>INFLUENCE OF SYSTEMS ENGINEERING PRINCIPLES IN A TECHNICAL SOCIETY</td>
</tr>
<tr>
<td>54</td>
<td>INNOVATIVE APPROACHES TOWARD SYSTEMS ENGINEERING EDUCATION DELIVERY</td>
</tr>
</tbody>
</table>
# LIST OF TABLES & FIGURES

<table>
<thead>
<tr>
<th>Figure/Section</th>
<th>Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Figure 1-1</td>
<td>MBSE Modeling at Multi Levels of the System</td>
<td>5</td>
</tr>
<tr>
<td>Figure 2-1</td>
<td>MBSE Across the System Life Cycle</td>
<td>6</td>
</tr>
<tr>
<td>Figure 2-2</td>
<td>MBSE Model Centric Approach</td>
<td>6</td>
</tr>
<tr>
<td>Figure 2-3</td>
<td>Transition to MBSE</td>
<td>7</td>
</tr>
<tr>
<td>Figure 2-4</td>
<td>INCOSE MBSE Roadmap</td>
<td>7</td>
</tr>
<tr>
<td>Figure 2-5</td>
<td>Technical and Business Advantages Of A MBSE Approach</td>
<td>8</td>
</tr>
<tr>
<td>Figure 3-1</td>
<td>An Integrating Framework For The Systems Engineering Vision 2020</td>
<td>10</td>
</tr>
<tr>
<td>Table 3-1</td>
<td>Drivers And Inhibitors Influencing The Practice Of Systems Engineering</td>
<td>11</td>
</tr>
<tr>
<td>Figure 3-2</td>
<td>Cross-Domain Model Integration</td>
<td>13</td>
</tr>
<tr>
<td>Figure 4-1</td>
<td>OOSEM Pyramid - Top-Down (Functional Decomposition) Approach</td>
<td>15</td>
</tr>
<tr>
<td>Figure 4-2</td>
<td>OOSEM Activities - Integration W/Object-Oriented S/W Development, H/W Development, And Test</td>
<td>15</td>
</tr>
<tr>
<td>Figure 4-3</td>
<td>Standard SysML Activity Diagram</td>
<td>16</td>
</tr>
<tr>
<td>Figure 4-4</td>
<td>IBM Rational Telelogic Harmony-SE Process</td>
<td>17</td>
</tr>
<tr>
<td>Figure 4-5</td>
<td>RUP-SE - Viewpoint of SE Lifecycle</td>
<td>18</td>
</tr>
<tr>
<td>Figure 4-6</td>
<td>RUP-SE - Requirements Allocations vs. derived Method</td>
<td>18</td>
</tr>
<tr>
<td>Figure 4-7</td>
<td>Model- and State-Based Control Architecture</td>
<td>19</td>
</tr>
<tr>
<td>Figure 4-8</td>
<td>Elements and Relationships of State Analysis Synthesized with Functional Analysis</td>
<td>20</td>
</tr>
<tr>
<td>Figure 4-9</td>
<td>VITECH MBSE Domains</td>
<td>21</td>
</tr>
<tr>
<td>Figure 4-10</td>
<td>VITECH Onion Model</td>
<td>21</td>
</tr>
<tr>
<td>Figure 4-11</td>
<td>Example of Top Level Specification of the OPM Metamodel</td>
<td>22</td>
</tr>
<tr>
<td>Figure 4-12</td>
<td>OPM Zooming into System Developing</td>
<td>23</td>
</tr>
<tr>
<td>Figure 4-13</td>
<td>Weikens Systems Modeling Process (SYSMOD) Methodology</td>
<td>24</td>
</tr>
<tr>
<td>Figure 4-14</td>
<td>Fernandez ISE &amp; Process Pipelines in OO Architectures (ISE&amp;PPOOA) Methodology</td>
<td>25</td>
</tr>
<tr>
<td>Figure 5-1</td>
<td>Relationship Between SysML and UML</td>
<td>26</td>
</tr>
<tr>
<td>Figure 5-2</td>
<td>SysML Diagram Types</td>
<td>27</td>
</tr>
<tr>
<td>Figure 5-3</td>
<td>The Four Pillars of SysML</td>
<td>27</td>
</tr>
<tr>
<td>Figure 5-4</td>
<td>Simple Model Construction</td>
<td>31</td>
</tr>
<tr>
<td>Figure 6-1</td>
<td>Example of Core Model Communication</td>
<td>31</td>
</tr>
<tr>
<td>Figure 6-2</td>
<td>Delivers Multidisciplinary Design Optimization (MDO)</td>
<td>32</td>
</tr>
<tr>
<td>Figure 6-4</td>
<td>Simplifies System-of-Systems (SOS) Analysis</td>
<td>33</td>
</tr>
<tr>
<td>Figure 9-1</td>
<td>Integrated Game-Changing Solution Strategy- Connected to Systems 2020 Objections</td>
<td>46</td>
</tr>
</tbody>
</table>
1.0 Model-Based Systems Engineering (MBSE) Initiative

(Source data: http://www.omgwiki.org/MBSE/doku.php#mbse_initiative_overview & www.incose.org/enchantment/docs/07docs/07jul_4mbseroadmap.pdf)

The MBSE Initiative was initiated at the INCOSE International Workshop in 2007. Its Charter was to “Promote, advance, and institutionalize the practice of MBSE to attain the MBSE 2020 Vision through broad industry and academic involvement in:

- Research
- Standards
- Processes, Practices, & Methods
- Tools & Technology
- Outreach, Training & Education

Additional details on this are covered in the next section.

![Figure 1-1: MBSE Modeling at Multi levels of the System](image)
2.0 MBSE Definitions & Advantages
Definitions and advantages of Model-Based Systems Engineering (MBSE) has been defined by both International Council on Systems Engineering (INCOSE) and SysML Forum and are covered below.

2.1 International Council on Systems Engineering (INCOSE) Definition & Advantages
(Source data: http://www.incose.org/Enchantment/docs/07docs/07Jul_4mbseroadmap.pdf)

2.1.1 Model-Based Systems Engineering (MBSE) Definition
"Is 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", shown in Figure 2-1.

![MBSE Across the System Life Cycle](image1)

**Figure 2-1: MBSE Across the System Life Cycle**

MBSE is part of a long-term trend toward model-centric approaches adopted by other engineering disciplines, including mechanical, electrical and software, as depicted in Figure 2-2.

![MBSE Model Centric Approach](image2)

**Figure 2-2: MBSE Model Centric Approach**
In particular, MBSE is expected to replace the document-centric approach that has been practiced by systems engineers in the past and to influence the future practice of systems engineering by being fully integrated into the definition of systems engineering processes, depicted in Figure 2-3: Transition to MBSE.

Figure 2-3: Transition to MBSE

INCOSE MBSE Roadmap is shown in Figure 2-4.

<table>
<thead>
<tr>
<th>MBSE Capability</th>
<th>Reduced cycle times</th>
<th>System of systems interoperability</th>
<th>Design optimization across broad trade space</th>
</tr>
</thead>
<tbody>
<tr>
<td>Institutionalized MBSE across Academia/Industry</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Well Defined MBSE</td>
<td></td>
<td></td>
<td>Cross domain effects based analysis</td>
</tr>
<tr>
<td>Ad Hoc MBSE Document Centric</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Extending Maturity and Capability

- Distributed & secure model repositories crossing multiple domains
- Architecture model integrated with Simulation, Analysis, and Visualization
- Defined MBSE theory, ontology, and formalisms
- Matured MBSE methods and metrics, Integrated System/HW/SW models
- Emerging MBSE standards
- Architecture model integrated with Simulation, Analysis, and Visualization
- Defined MBSE theory, ontology, and formalisms
- Matured MBSE methods and metrics, Integrated System/HW/SW models
- Emerging MBSE standards

Refer to activities in the following areas:
- Planning & Support
- Research
- Standards Development
- Processes, Practices, & Methods
- Tools & Technology Enhancements
- Outreach, Training & Education

Figure 2-4: INCOSE MBSE Roadmap

2.1.2 Advantage to using MBSE

MBSE enhances the ability to capture, analyze, share, and manage the information associated with the complete specification of a product, resulting in the following benefits:

- **Improved communications** among the development stakeholders (e.g. the customer, program management, systems engineers, hardware and software developers, testers, and specialty engineering disciplines).
- **Increased ability to manage system complexity** by enabling a system model to be viewed from multiple perspectives, and to analyze the impact of changes.
- **Improved product quality** by providing an unambiguous and precise model of the system that can be evaluated for consistency, correctness, and completeness.
- **Enhanced knowledge capture** and reuse of the information by capturing information in more standardized ways and leveraging built in abstraction mechanisms inherent in
model driven approaches. This in turn can result in reduced cycle time and lower maintenance costs to modify the design.

- **Improved ability to teach and learn systems engineering fundamentals** by providing a clear and unambiguous representation of the concepts.

### 2.2 SysML Forum MBSE Definition & Advantage
(Source information is from: [http://www.sysmlforum.com/faq/what-is-MBSE.html](http://www.sysmlforum.com/faq/what-is-MBSE.html))

#### 2.2.1 SysML Forum MBSE Definition

MBSE is an umbrella term that describes an approach to Systems Engineering that:

- Emphasizes a system architecture model as the primary work artifact throughout the System Development Life Cycle (SDLC)
- Combines traditional systems engineering best practices with rigorous visual modeling techniques
- Typically uses the SysML as a standard visual modeling language. SysML not only supports rigorous modeling techniques among Systems Engineers, it also serves as a lingua franca among various other kinds of engineers (Software Engineers, Electrical Engineers, Mechanical Engineers, etc.) and stakeholders who need to understand MBSE work products, such as visual requirements analysis and verification, CONOPS, functional analysis and allocations, performance analysis, trade studies, and system architectures.

#### 2.2.2 SysML Forum MBSE Advantage

MBSE offers systems engineers the following advantages over traditional document-centric Systems Engineering approaches:

<table>
<thead>
<tr>
<th>Technology drivers</th>
<th>Technology Advantages</th>
<th>Business Advantages</th>
</tr>
</thead>
<tbody>
<tr>
<td>Model Requirements</td>
<td>Ensure that the requirements are an integral part of the model and all other parts of the model can be traced back to requirements</td>
<td>Validate that you are &quot;building the right system&quot;...[Boehm 1984]</td>
</tr>
<tr>
<td>Model Analysis &amp; Design</td>
<td>Provide a precise architectural blueprint organization by the views that are meaningful to all systems stakeholders</td>
<td>Verify that you are &quot;building the system the right way&quot; [Boehm 1984]</td>
</tr>
<tr>
<td>Model Simulation</td>
<td>Automate system verification and validation</td>
<td>Reduce errors and costs early in the lifecycle</td>
</tr>
<tr>
<td>Model Code</td>
<td>Automate generation of production quality code</td>
<td>Accelerated time to market</td>
</tr>
<tr>
<td>Model Test</td>
<td>Automate testing</td>
<td>Ensure system implementation is correct and reliable</td>
</tr>
</tbody>
</table>

**Figure 2-5: Technical and Business Advantages Of A MBSE Approach**
3.0 INCOSE Systems Engineering Vision 2020

This INCOSE Systems Engineering 2020 vision of the evolution of systems engineering began with an assessment of the current state of practice and trends, identified drivers and inhibitors, then extrapolated the future state of practice in systems engineering. This analysis was conducted in five focus areas - the Global Systems Engineering Environment, Systems and their Nature, Systems Engineering Processes, Models and Model-Based Systems Engineering, and Systems Engineering Education.

Several significant trends in the global environment are leading to the emergence of a more widespread and effective application of the systems engineering practice. There is a growing realization that systems engineering is essential to successfully design, develop and sustain the highly complex systems of the 21st century. Yet, the profession suffers from the lack of a set of unified principles, models that support a wide range of domains, and consistent terminology and definitions. Furthermore, there is a prevailing perception that systems engineering is overly cumbersome and not readily applicable to small projects or small businesses.

One can expect a shift away from a “one size fits all” definition and application of systems engineering to a more specifically defined and precise application of systems engineering in diverse domains. The future systems engineering environment will fully support life cycle perspectives such as supply chains and system sustainment.

The following areas within the System Engineering 2020 Reports (both 2007 & 2008) are covered in this media study in more detail:
- Vision 2020 integrating framework,
- Systems engineering vision, and
- Role of MBSE within systems engineering vision 2020

3.1 Vision 2020 Integrating Framework
As illustrated in Figure 3-1 below, the INCOSE Systems Engineering Vision 2020 forecasts the future state of the practice of systems engineering, extrapolated from evolutionary developments in the current state of practice and trends, and a set of drivers and inhibitors that will influence the future state. The analysis was conducted in five focus areas, as follows:
- Global Systems Engineering Environment sets the international context and imperatives;
- Systems and their Nature describes the scope and technical areas of the discipline;
- Systems Engineering Processes defines elements of the practice of the discipline;
- Models and Model-based Systems Engineering introduces new capability into systems engineering practices; and,
- Systems Engineering Education provides an insight into the primary source of communicating and advancing the principles and practice.

Table 3-1 contains a sampling of observed drivers and inhibitors, categorized according to each of the five focus areas, which have significant potential to influence the state of practice of systems engineering.
3.2 Systems Engineering Vision

The current state of systems engineering practice is evolving under the influence of the respective drivers and inhibitors called out in Table 3-1. Through a series of workshops experts in systems engineering and related fields were brought together to hypothesize about the future of systems engineering, beyond the next decade.

In a best case scenario, the experts anticipate that systems thinking and systems engineering will guide the way people think about solving problems in the next decade and systems engineering will become an established international “inter-disciplinary connector” or “meta-discipline.” Increased diversity within the stakeholder population creates the need for shared sense making through shared mental models; a need that systems engineering will satisfy. In the future systems engineering will be used to address the significant social, economic, environmental and planning issues of the day.

Additional systems engineering capabilities will be created to support the adaptation of system engineering methodology to the agile and robust operations of extended enterprises and businesses of all sizes. This will include:

- Advanced systems theory application
- Increased use of analytical methods and tools
- Advances in engineering education with an emphasis on interdisciplinary integration
- Improved use/integration of engineering specialties
- Improved understanding of psychology, languages and culture
- Improved shared understanding of systems engineering concepts among all stakeholders

Stovepipes between disciplines will begin to blur as the benefits – diversity of ideas, lower organizational costs and greater development efficiency – of interdisciplinary integration are realized. Stakeholders will take the time to identify the characteristics (capability, culture, communication and decision making style, appetite for risk, ethics, etc.) of their global business partners.
<table>
<thead>
<tr>
<th>Area</th>
<th>Drivers</th>
<th>Inhibitors</th>
</tr>
</thead>
</table>
| Global Environment | • Ability to create, produce and operate on a global basis, supported by rapid and extensive transportation of people and goods, plus high-bandwidth, globally integrated communications capability  
• Increasing collaboration among professional groups and other stakeholders, worldwide | • Lack of mature standards and their uniform adoption  
• Increased global tensions; local customs and practices and, decreasingly, language  
• Heritage of systems engineering impacts its perceived application to social and non-traditional systems, and small to medium enterprises |
| Nature of systems | • Nonlinearly increasing ability to incorporate a greater capability for a given cost; to augment the embedded intelligence in systems and to provide systems with global dispersion  
• Emerging basic conceptual and technological areas with systems applications, such as chaos theory, nano-technology and genetic engineering | • Human limits in comprehension and control of man-made systems  
• Immature view of the roles of people in (increasingly complex) systems, natural entities, processes and procedures, and facilities  
• Lag in manufacturing and sustainment capabilities  
• Slow grasp of the significance of new technologies and concepts |
| SE Processes      | • Evolution of standards that give a consistent view of process sets that embrace all systems engineering functions on a life cycle basis  
• Emerging understanding of software engineering and systems engineering synergies | • Lack of tools that enable use of existing standards  
• Excessively complex process sets and burdensome formalism creates a barrier to acceptance of systems engineering practices, especially in small and medium enterprises |
| MBSE             | • Emergence and maturation of modeling languages and information standards  
• Continuing evolution of information technology as an enabler of modeling techniques | • Inherent difficulty integrating models across organizational, lifecycle and other boundaries  
• Limitation of model /data exchange capabilities within the modeling tools  
• Limited MBSE skills |
| SE Education     | • Recognition of need to inculcate systems thinking at earlier stages in individual educational experiences and with a broader context  
• Distance education putting the classroom in the home  
• Transition to transdisciplinarity in engineering education | Lack of an accepted set of theory and principles  
Lack of an overarching vision or guiding framework for research  
Lagging posture of curricula, exacerbated by funding shrinkage  
Challenge of educating current employers in new concept |
3.3 Role of MBSE within Systems Engineering Vision 2020

Per INCOSE Systems Engineering Vision 2020, the projected state of MBSE practice in 2020 will extend MBSE to modeling domains beyond engineering models to support complex predictive and effects-based modeling. This will include the integration of engineering models with scientific and phenomenology models, social, economic, and political models and human behavioral models. The key characteristics of MBSE in 2020 include:

- Domain-specific modeling languages and visualization that enable the systems engineer to focus on modeling of the user domain
- Modeling standards based on a firm mathematical foundation that support high fidelity simulation and real-world representations
- Extensive reuse of model libraries, taxonomies and design patterns
- Standards that support integration and management across a distributed model repository
- Highly reliable and secure data exchange via published interfaces.

Domain-specific modeling languages built on the general purpose systems modeling language will increase the abstraction level to represent the user domain. Validated and specialized model libraries for specific domains will be established that can be reused across organizations and evolved over time. A system engineer familiar with the domain will be able to rapidly search the distributed model repository and evaluate a broad trade space of solutions based on an understanding of the user requirements and measures of effectiveness. Thus, the systems engineer will readily perform what-if analysis to assess the requirements, design, and technology impacts, and optimize the solution, using multi-dimensional visualization capabilities.

The conceptual depiction in Figure 3-2 indicates a desired goal and direction toward advancing capabilities for MBSE. The integrated capabilities shown will dramatically increase the application of MBSE to support marketing research, decision analysis, integration with biological system models, environmental impact analysis and the design of social systems in support of urban planning and government social programs, to name a few.

Collaborative environments will be set up quickly for new project teams and extended across global government, industry and academic teams. Management of product data throughout the lifecycle will provide improved logistics and Operations and Maintenance phase support, since design data is retained in standards-based repositories. Virtual development environments will minimize the need for physical prototypes and will accelerate new product development while providing realistic verification against customer driven requirements.
System development times will be substantially reduced relative to current practice, while improving overall system quality and availability. This will be accomplished by a combination of failure-mode avoidance and knowledge-based engineering. The application of increased computer power will enable rapid system design with models in virtual development environments, greatly reducing the need for physical prototypes.

The breadth of applied models will not be confined to the traditional extent of dealing with the immediate system interfaces. Relationships will emerge between individual system models that are built to analyze systems across life cycles spanning decades and the social and economic models that estimate the impact of cumulative decisions on the economy and our environment. “Intelligent systems” will add significant effort to up-front systems engineering with the benefit of enabling planned change at an industry, domain, or societal level. System models and simulations will explore the impacts to our environment, society, and the stakeholder organizations in ways that are ignored today and discovered by analysis of impacts after systems become operational.

A complete discussion of the MBSE vision must also be accompanied by realistic concerns about inhibitors to its progress such as those described in Table 3-1. Difficult technical and cultural challenges remain to be overcome in order to realize the many facets and benefits of the envisioned MBSE evolution. Meaningful progress will require both market forces and motivated visionaries inclined to take some risks in pushing the envelope to demonstrate value, exploiting opportunities and setting an example for others to follow.
4.0 MBSE Methodologies

In general, a methodology can be defined as the collection of related processes, methods, and tools used to support a specific discipline [Martin, 1996]. That more general notion of methodology can be specialized to MBSE methodology, which we characterize as the collection of related processes, methods, and tools used to support the discipline of systems engineering in a “model-based” or “model-driven” context [Estefan, 2008].

In 2007, a formal survey of candidate MBSE methodologies was published as part of the work of the INCOSE MBSE Focus Group that later was formalized as the INCOSE MBSE Initiative. In 2008, that formal survey of candidate MBSE methodologies was published under the auspices of an INCOSE technical publication. The 2008 report surveyed six (6) candidate MBSE methodologies:
- INCOSE Object-Oriented Systems Engineering Method (OOSEM)
- IBM Rational Telelogic Harmony-SE
- IBM Rational Unified Process for Systems Engineering (RUP-SE)
- Vitech Model-Based Systems Engineering (MBSE)
- JPL State Analysis (SA)
- Dori Object-Process Methodology (OPM)

Additional methodologies identified as gaps since the 2008 INCOSE Survey:
- Weilkiens Systems Modeling Process (SYSMOD)
- Fernandez Process Pipelines in OO Architectures (PPOOA)

An overview of each of these methodologies is covered below.

It should be noted that the scope of the survey report went beyond a simple survey and also documented a number of key issues related to the discipline of MBSE, including the following:
- differentiating processes, methods, and tools; characterizing the role of lifecycle models (project, acquisition, and systems engineering);
- an explanation of models in support of MBSE processes; and,
- documenting the roles of model-based hazard analysis, UML/OMG SysML, OMG model-driven architecture (MDA), and executable UML foundation.

4.1 INCOSE Object-Oriented Systems Engineering Method (OOSEM)
- Integrates top-down (functional decomposition) approach with model-based approach
- Leverages object-oriented concepts
  - Uses OMG SysML (formerly UML) to support specification, analysis, design and verification of systems
- Intended to ease integration w/object-oriented S/W development, H/W development, and test
- OOSEM includes the following activities:
  - Analyze Stakeholder Needs
  - Define System Requirements
  - Define Logical Architecture
  - Synthesize Candidate Allocated Architectures
  - Optimize and Evaluate Alternatives
  - Validate and Verify System
• Tool- and vendor-neutral

Figure 4-1: OOSEM Pyramid - Top-Down (Functional Decomposition) Approach

Requirements Traceability is Managed Through the Entire MBSE Process

Figure 4-2: OOSEM Activities - Integration W/Object-Oriented S/W Development, H/W Development, And Test
4.2 IBM Rational Telelogic Harmony-SE

- Service Request-Driven Approach
  - Described by means of SysML structure diagrams
  - State/mode changes (activities) described as operational contracts
- Somewhat mirrors “Vee” model
- Task flow and work products (artifacts) include following top-level process elements:
  - Requirements Analysis
  - System Functional Analysis
  - Architectural Design
- Detailed task flows and work products provided for each process element
  - Modeled as SysML activity diagrams

Figure 4-3: Standard SysML Activity Diagram
4.3 IBM Rational Unified Process for Systems Engineering (RUP-SE)

- Extends RUP style of concurrent design and iterative development to support SE in Model-Driven Systems Development (MDSD)
  - New roles, e.g., systems engineers
  - New artifacts & disciplines, e.g., security, training, logistics
- Emphasis on business modeling, business actors and flow of events to adequately define system requirements
- Viewpoints for SE, i.e., model levels and model viewpoints
- Introduces concept of locality — Member of system partition representing generalized or abstract view of physical resources; linked by connections (M. Cantor)
- Scalability enhancements where each subsystem coupled w/locality has own derived requirements
  - Allocated vs. derived requirements — locality/subsystem is allocated if fulfilling system requirement; is derived if identified by studying how collaborates with other to meet system requirement
  - Subsystem-level flowdown activity via use-case flowdown activity and flow of events in “white-box view of system
- Support for designing additional components (e.g., H/W) in addition to RUP S/W focus
Figure 4-5: RUP-SE - Viewpoint of SE Lifecycle

Figure 4-6: RUP-SE - Requirements Allocations vs. derived Method
4.4 JPL State Analysis (SA)

- JPL-developed methodology that leverages model- and state-based control architecture
  - **state** - representation of momentary condition of evolving system
  - **models** - describe how state evolves
  - **state variables** - abstractions representing “knowledge” of state
    - Known state of system is value of its state variables at time of interest
  - Together, state and models supply what is needed to operate system, predict future state, control toward desired state, & assess performance
  - SA methodology defines iterative process for state discovery & modeling
    - Allows models to evolve as appropriate across project lifecycle
    - SA requirements process helps bridge gap between requirements on software specified by systems engineers
      - **Mission Data System (MDS)** — embedded software architecture designed to provide multi-mission information and control architecture for robotic exploration spacecraft
    - SA information compiled in “State Database”

![Diagram of Model- and State-Based Control Architecture](image)

**Figure 4-7: Model- and State-Based Control Architecture**
4.5 Vitech Model-Based Systems Engineering (MBSE) Methodology

- Four (4) primary concurrent SE activities linked and maintained through common system design repository
- Each activity linked within context of associated “domains”
  - Process Domain (SE activities)
  - Source Requirements Domain
  - Behavior Domain
  - V&V Domain
  - Architecture Domain
- Strong adherence to agreed upon System Definition Language (SDL), i.e., SE schema or ontology to manage syntax and semantics of model artifacts
- Uses incremental process known as “Onion Model”
  - Allows complete interim solutions at increasing levels of detail during system specification process
  - Lower risk design approach by checking for completeness and discovering constraints early in design process
- Detailed testing methods described in support of system V&V activity
Figure 4-9: Vitech MBSE Domains

Figure 4-10: Vitech Onion Model
4.6 Dori Object-Process Methodology (OPM)

- Formal paradigm to systems development, lifecycle support, and evolution
  - Combines simple OPDs (diagrams) with OPL (constrained natural language)
  - Basic building blocks:
    - **object** - thing that exists or has potential of existence, physically or mentally,
    - **process** - pattern of transformation that object undergoes,
    - **state** - situation object can be at
- Reflective methodology refers to system lifecycle as system evolution
  - In OPM, “process” is a reserved word; therefore, “system process” is used
  - System Developing process (SD1) contains three main stages, each can be “zoomed” of which can be further zoomed
    - Requirement Specifying
    - Analyzing & Developing
    - Implementing
    - [also, Using & Implementing stage]
- Visual models tool- and vendor-specific (see Tool Support below)

### Visual Models Tool- and Vendor-Specific

**SD**

<table>
<thead>
<tr>
<th>OPM exhibits Ontology and Notation, as well as System Developing.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Notation represents Ontology</td>
</tr>
<tr>
<td>System Developing requires Notation and Ontology.</td>
</tr>
<tr>
<td>System Developing yields System.</td>
</tr>
<tr>
<td>User is environmental and physical</td>
</tr>
<tr>
<td>User handles System Developing.</td>
</tr>
</tbody>
</table>

**Figure 4-11: Example of Top Level Specification of the OPM Metamodel**
4.7 Weiklens Systems Modeling Process (SYSMOD)

The process is introduced in the book Systems Engineering with SysML/UML by Tim Weiklens. Tim Weiklens, managing director of the German consulting company oose GmbH, is a member of the OMG working groups about SysML and UML and has written sections of the SysML specification. He is involved in the INCOSE MBSE activities and co-founder of the MBSE Challenge Team SE^2 Telescope Modeling.

The Systems Modeling Process (SYSMOD) is a pragmatic approach to model the requirements and the functional and physical architecture of a system. It provides a toolbox of tasks with input and output work products, guidelines and best practices. SYSMOD uses the OMG Systems Modeling Language (OMG SysML).

- User-oriented approach for requirements engineering and system architectures
- Allows different levels of modeling intensities
- Guidelines and examples provided for each process activity
- SYSMOD Includes the following activities:
  - Identify stakeholder
  - Elicit requirements
  - Define system context
  - Analyze requirements, e.g. with use cases
  - Define domain model
  - Define system architecture on different levels (functional, logical, physical)
- Provides additional activities, e.g., for functional architectures or variant modeling
- Tool vendor independent methodology
4.8 Fernandez ISE & Process Pipelines in OO Architectures (ISE&PPOOA)

- ISE&PPOOA (Integrated Systems Engineering and PPOOA) provides an integrated process, methods and a tool for systems engineering of software intensive mechatronic systems.
- The ISE part of the process includes the first steps of a systems engineering process applicable to any kind of system, not only the software intensive ones. The ISE subprocess integrates traditional systems engineering best practices and MBSE.
- The PPOOA part of the process emphasizes the modeling of concurrency as earlier as possible in the software engineering part of the integrated process.
- The integration between the systems engineering subprocess and the PPOOA software engineering subprocess is achieved by using a responsibility driven software analysis approach supported by CRC cards, a technique proposed in the OOPSLA’89 by Beck and Cuningham.
- ISE&PPOOA provides a collection of guidelines or heuristics to help the engineers in the architecting of a system.
- One of the project deliverables is the functional architecture representing the functional hierarchy using the SysML block definition diagram. This diagram is complemented with activity diagrams for the main system functional flows. The N2 diagram is used as an interface diagram where the main functional interfaces are identified. A textual description of the system functions is also provided as part of the deliverable.
- Other of the deliverables is the physical architecture, representing the system decomposition into subsystems and parts using the SysML block definition diagram. This diagram is complemented with SysML internal block diagrams for each subsystem and activity and state diagrams as needed. A textual description of the system blocks is also provided. The heuristics used for the particular architecture solution are identified and documented.
- The software subsystem architecture is described in PPOOA using two views supported by one or more diagrams using UML notation. One view is the static of structural view and the other is the dynamic or behavioral view of the system. The system architecture diagram represents the system components and the composition and usage relations between them. Coordination mechanisms used as connectors are also represented. The system behavioral view is supported by UML/SysML activity
diagrams representing an internal view of the flow of actions performed by the system in response to an event

- Bidirectional functional allocation between components and system responses modeled as activity diagrams is considered and tool supported
- ISE&PPOOA makes easy the architecture evaluation from time responsiveness characteristics

![Diagram](image)

**Figure 4-14: Fernandez ISE & Process Pipelines in OO Architectures (ISE&PPOOA) Methodology**
5.0 MBSE Modeling Language Standards
(Source data: http://www.incose.org/productspubs/pdf/techdata/mttc/mbse_methodology_survey_2008-0610_revb-jae2.pdf)

Unified Modeling Language™, UML®, and Systems Modeling Language™, OMG SysML™, (both covered it more details below) are visual modeling language standards managed under the auspices of the Object Management Group™, OMG™; an open membership, not-for-profit consortium that produces and maintains computer industry specifications for interoperable enterprise applications.

UML and OMG SysML are intended to be complementary. SysML was developed in partnership between the OMG and INCOSE and is based on the UML, specifically, UML 2.0 (see Figure 5-1) SysML provides an additional set of modeling diagrams to model complex systems that include hardware, software, data, procedures and other system components. Together, UML and SysML go a long way to help unify what has historically been a communication chasm between the systems and software engineering communities.

5.1 OMG SysML™
(Source data: http://www.omgsysml.org/)

The OMG Systems Modeling Language (OMG SysML™) is a general-purpose graphical modeling language for specifying, analyzing, designing, and verifying complex systems that may include hardware, software, information, personnel, procedures, and facilities. In particular, the language provides graphical representations with a semantic foundation for modeling system requirements, behavior, structure, and parametrics, which is used to integrate with other engineering analysis models. SysML represents a subset of UML 2 with extensions needed to satisfy the requirements of the UML™ for Systems Engineering RFP as indicated in Figure 5-1. SysML leverages the OMG XML Metadata Interchange (XMI®) to exchange modeling data between tools, and is also intended to be compatible with the evolving ISO 10303-233 systems engineering data interchange standard.

![Figure 5-1: Relationship Between SysML and UML](image)

The SysML diagram types are identified in Figure 5-2 for a more detailed description refers to the OMG SysML Tutorial for an overview of the language or the APL MBSE with SysML course material.
Figure 5-2: SysML Diagram Types

Figure 5-3 shows an example of the 4 pillars of the OMG systems Modeling Language (OMG SysML™).

1. Structure
   - Interaction
   - State machine
   - Activity/function

2. Behavior
   - Definition
   - State machine

3. Requirements

4. Parametrics

Note that the Package and Use Case diagrams are not shown in this example, but are respectively part of the structure and behavior pillars.

Figure 5-3: The Four Pillars of SysML
Multiple vendors announced SysML implementations:

- Atego (previously) ARTISAN Software Tools
- CoFluent Design
- IBM
  - Rhapsody
  - Tau
- InterCAX
- No Magic MagicDraw SysML
- Papyrus for SysML (open source eclipse modeling tool)
- SOFTEAM MODELIO - Solutions for System Architects
- Software Stencils - Microsoft Visio SysML and UML templates
- Sparx Systems

5.2 UML


The Unified Modeling Language™ - UML - is OMG's most-used specification, and the way the world models not only application structure, behavior, and architecture, but also business process and data structure.

UML includes a set of graphic notation techniques to create visual models of object-oriented software-intensive systems. UML is used to specify, visualize, modify, construct and document the artifacts of an object-oriented software-intensive system under development.[2] UML offers a standard way to visualize a system's architectural blueprints, including elements such as:

- activities
- actors
- business processes
- database schemas
- (logical) components
- programming language statements
- reusable software components

UML combines techniques from data modeling (entity relationship diagrams), business modeling (work flows), object modeling, and component modeling. It can be used with all processes, throughout the software development life cycle, and across different implementation technologies. UML has synthesized the notations of the Booch method, the Object-modeling technique (OMT) and Object-oriented software engineering (OOSE) by fusing them into a single, common and widely usable modeling language. UML models may be automatically transformed to other representations (e.g. Java) by means of QVT-like transformation languages. UML is extensible, with two mechanisms for customization: profiles and stereotypes.
6.0 MBSE Software Tools
The following tools have been captured and top level information each tool has been provided below:

- SAE Architecture Analysis & Design Language (AADL)
- Vitech CORE
- MagicDraw®
- Phoenix Integration
- ModelCenter
- IBM® Rational® Rhapsody®
- Eclipse
- Isight & SIMULIA

6.1 MBSE Tools and Relevance of other “de-facto” MBSE visual modeling standards

A recommendation for the MBSE tool vendors that currently do not support UML and/or OMG SysML is to add this capability to the product roadmap as soon as possible; particularly, SysML. The advantage of using industry standard visual modeling languages over vendor-specific modeling languages is clear and does not warrant debate. Some MBSE tools, for example, only support the Enhanced Function Flow Block Diagram (EFFBD) visual modeling capability, which in some cases (e.g., Vitech CORE/CoreSim) support executable modeling constructs that allows the systems engineer to run discrete-event simulations based on EFFBD models. This is a very powerful capability and there is no technical reason that such an executable capability could not be added to OMG SysML diagrams such as activity diagrams and state diagrams. Bock has documented how both UML (UML 2 specifically) and SysML can be used for activity modeling and how these standards can extended to fully support EFFBDs.

6.2 SAE Architecture Analysis & Design Language (AADL)
(Source data: http://www.aadl.info/aadl/currentsite/)

This standard defines a language for describing both the software architecture and the execution platform architectures of performance-critical, embedded, real-time systems; the language is known as the SAE Architecture Analysis & Design Language (AADL). AADL has the following features:

- Gives you the power to specify and generate a single model that can be analyzed for multiple qualities.
- Provides an industry-standard, textual and graphic notation with precise semantics to model applications and execution platforms
- Features an XML interchange format that supports the exchange of models between subcontractors, integrators, and agencies
- Includes a UML profile that presents AADL as a specialized modeling notation within UML framework
- Supported by commercial and open source tool solutions—including the SEI Open Source AADL Tool Environment (OSATE)
An AADL model describes a system as a hierarchy of components with their interfaces and their interconnections. Properties are associated to these constructions. AADL components fall into two major categories: those that represent the physical hardware and those representing the application software. The former is typified by processors, buses, memory, and devices, the latter by application software functions, data, threads, and processes. The model describes how these components interact and are integrated to form complete systems. It describes both functional interfaces and aspects critical for performance of individual components and assemblies of components. The changes to the runtime architecture are modeled as operational modes and mode transitions.

6.3 Vitech CORE
(Source data http://www.vitechcorp.com/products/core.shtml)

CORE® is a comprehensive modeling environment built for complex systems engineering problems. Whether you are performing a one-month study or a long-term architecture design, project success depends on getting started quickly, finding the right solution, and being responsive to program changes. Leveraging CORE’s deep model-centric capability delivered through a rich, intuitive user framework delivers early and continuous insight into your problem domain.

Leverage CORE’s integrated modeling capabilities to assess and control design and program risks. By linking all elements of your system through a central model, you’ll have greater visibility into drivers for risk and learn more quickly where the weaknesses are in your design. Stay competitive by avoiding poor designs that are more expensive. Build better models and deliver better products to market through:

- Integrated requirements management ensure that you capture customer needs accurately
- Fully executable behavior models help you unlock elegant functionality
- Architecture development tools guide you swiftly to subsystems and components
- Validation and Verification (simulation) highlights gaps and missing functions
- Comprehensive system documentation ensures you deliver a complete solution with a clear message

CORE enables you to make the shift to MBSE quickly and easily by guiding you through a simple, layered approach to development. As new users join your team, you’ll be able to collaborate more easily and share a common modeling language and to focus on the engineering work and not the tools.

- **Easy to use Parsers** help you get information into the model quickly so you can build the solution correctly the first time. Reduce rework by verifying you’ve secured the right requirements early.
- **Simple Model Construction** gives new users a flexible way to visualize the interactions between requirements, functions, and components so you can always keep your focus.
Multiple Modeling Notations help you evaluate the system through a variety of integrated graphical views: hierarchies, functional flow and enhanced functional flows, N2, IDEF0, and physical block. Change one diagram and see the update reflected on all impacted views.

Automated Document Creation helps you generate formal documentation, including DoDAF 2.0 views, instantly from the system definition database.

Enhanced Communication: CORE gives you a powerful new medium to capture the intent of your system and effectively communicate across your team. You'll be able to demonstrate aspects of your design without sacrificing traceability. CORE gives you the power of a fully integrated data model with the simplicity of a graphical environment.

6.4 MagicDraw®
(Source data https://www.magicdraw.com/what_is)

MagicDraw is an award-winning business process, architecture, software and system modeling tool with teamwork support. Designed for Business Analysts, Software Analysts, Programmers, QA Engineers, and Documentation Writers, this dynamic and versatile development tool facilitates analysis and design of Object Oriented (OO) systems and databases. It provides the industry's best code engineering mechanism (with full round-trip support for Java, C++, C#, CL (MSIL) and CORBA IDL programming languages), as well as database schema modeling, DDL generation and reverse engineering facilities.

Ten Reasons MagicDraw Literally Outpaces the Competition
1. Promotes quick learning with intuitive interface
2. Creates diagrams faster than any tool on the market
3. Derives models from existing source code in just seconds
4. Visualizes your model in a few quick steps
5. Keeps your team in the express lane by enabling them to work on the same model in parallel
6. Delivers source code from your UML model instantly
7. Eliminates tedious document preparation with automatic report generation!
8. Extends UML capabilities beyond UML 2 -- in a snap
9. Accelerates your ‘travel time’ between modeling domains
10. Enables speedy navigation through your models

6.4.1 Related MagicDraw Products and Plugins

MagicDraw has several add-ons to enhance the use of MagicDraw’s capability. The following is the list from the MagicDraw website. Follow the hyperlink for detailed information about each of the add-ons. https://www.magicdraw.com/magicdraw_addons

- Teamwork Server
- Cameo XSD Import Import Plugin
- Nomos OCL Business Rules Testing Addon
- UPDM Plugin
- SysML Plugin
- Cameo Simulation Toolkit
- Cameo Business Modeler
- Cameo DataHub
- Cameo Workbench
- Cameo SOA+
- Cameo Data Modeler
- TOGAF Plugin
- QVT Plugin
- Merge Plugin
- CameoMDA Plugin
- MagicRQ Plugin
- Cameo Inter-Op
- RSXConverter
- RConverter
- ParaMagic Plugin
- Methodology Wizards Plugin
- SPEM Plugin
- MARTE Profile
- MagicDraw Reader on iPhone
- DoDAF Plugin
- CSV Import Plugin

6.5 Phoenix Integration

(Source data: http://www.phoenix-int.com/)

Phoenix Integration, a global leader in software integration and Multidisciplinary Design Optimization (MDO), see Figure 6-3: Delivers Multidisciplinary Design Optimization (MDO), provides process integration and design optimization software that empowers you to improve your decision-making and analysis management capabilities. Our software delivers MDO and simplifies System-of-Systems analysis, see Figure 6-4: Simplifies System-of-Systems (SoS) Analysis.

![Figure 6-3: Delivers Multidisciplinary Design Optimization (MDO)](image-url)
6.5.1 Design Space Navigation/Optimization & Variable Sensitivity Analysis

Multidisciplinary design and analysis plays a vital role in the Phoenix solution. Once the components of your process are integrated, you can employ various design exploration tools to create an optimized design, run what-if trade studies or further inspect the design space to find the weaknesses in your Model.

**Design Space Navigation/Optimization**
- Change parameters, goals, and constraints on-the-fly
- Locate best designs for specified parameters, goals and constraints
- View Pareto Optimal designs
- Visually follow progress of optimization algorithms
- Visually guide search algorithm within design space
- Generate design comparison reports

**Variable Sensitivity Analysis**
- Visualize key global variable sensitivities-main effects and interaction effects-to focus on most influential variables
- Visualize local variable sensitivities
- Drag control panel's interactive sliders to easily and visually navigate design space in real time
- View key relationships and trends using advanced plotting options including glyph, star glyph, parallel coordinates, scatter, scatter matrix, histogram, and 2D histogram

6.5.2 ModelCenter

ModelCenter® 10.0 by Phoenix Integration is a graphical environment for automation, integration, and design optimization that supports your entire product development team. It allows you to lower cost and reduce product development time. With PHX ModelCenter, you can quickly create an engineering process and then explore the design space to find the best design. PHX ModelCenter is adaptable, and works well with groups whose design processes change frequently.
The engineering process is inherently about making design decisions and tradeoffs between competing objectives. There are often too many choices and too little time to evaluate them all. Computer simulation has helped the process by allowing engineers and analysts to create “virtual” prototypes. Even with simulation, however, design is largely an intuition based guessing game to find good designs. Furthermore, design decisions increasingly have cross-discipline impacts that are not understood by any one expert thus limiting the effectiveness of commonly used experience based approaches.

PHX ModelCenter® enables engineers to automate and integrate design codes and build complex parametric models of systems. When performing trade studies with these models, users sometimes face having to navigate large, complex design spaces using basic plotting tools. Searching for the best designs with optimization algorithms and communicating and justifying results to management less familiar with design optimization technology can be a difficult task. Includes:

- Variable Influence Profiler (VIP)
- Prediction Profiler (PP)
- Data Visualizer

6.5.2.1 Benefits

- **Better Designs** - PHX ModelCenter gives you powerful tools to inspect designs and find weaknesses. Correcting these weaknesses early in the design process can save corporations millions. Trade studies within PHX ModelCenter help you and your team find optimal solutions with lower weight, higher performance, and more robustness in the design.

- **Reduces Errors** - Manual data translation and linking is costly and error-prone. Automating this step in the design process helps reduce mistakes.

- **Saves Time** - PHX ModelCenter eliminates the manual transfer of data between applications, thereby reducing design cycle time. With that "extra" time, you can evaluate more design alternatives which results in a better final product.

- **Cohesive Team Environment** - Your team of engineers can use PHX ModelCenter to coordinate data and share design applications over a network of computers. Design data that used to be sent via fax, email, or ftp can now be linked automatically through PHX ModelCenter.

- **Easy to Deploy** - PHX ModelCenter integrates seamlessly with existing applications used by your entire team.

6.6 IBM® Rational® Rhapsody®


IBM® Rational® Rhapsody® family provides collaborative design and development for systems engineers and software developers creating real-time or embedded systems and software. Rational Rhapsody helps diverse teams collaborate to understand and elaborate requirements, abstract complexity visually using industry standard languages (UML, SysML, AUTOSAR, DoDAF, MODAF, UPDM), validate functionality early in development, and automate delivery of innovative, high quality products.

Rational Rhapsody Designer for Systems Engineers enables a Model Based Systems Engineering (MBSE) approach with SysML for visualization of complex requirements and model execution for early validation of requirements, architectural trade off analysis and mitigation of project risks.
- Integrated requirements and modeling environment using industry standard SysML or UML diagrams
- Dynamically analyze and execute SysML parametric diagrams to assist in trade study analysis
- Simulation executes the model to help validate architecture and behavior including activity diagram execution using token passing semantics
- Full lifecycle traceability and analysis from requirements to design
- Static model checking analysis helps improve design consistency
- Automates tedious tasks with systems engineering toolkit
- Capture systems of systems with additional DoDAF, MODAF and UPDM Add On
- Includes configuration management interface support with advanced graphical difference and merging capabilities for parallel development
- Automate documentation across product lifecycle with Rational Publishing Engine integration
- Collaborative development with integration with the Jazz-based IBM® Rational® Team Concert solution through Eclipse
- Automate validation testing with Rational Rhapsody TestConductor Add On including Rational Quality Manager integration
- IBM® Rational® Rose® import for legacy system re-use
- Integrations with IBM Rational products such as IBM Rational DOORS, IBM Rational Systems Architect, IBM Rational ClearCase for full product lifecycle development
- Rational Rhapsody Designer for Systems Engineers 7.6.1 provides a client to share and review The Mathworks Simulink designs through the integration with Rational Rhapsody

6.7 Eclipse
Open source tool.
(Source data: http://www.eclipse.org/org/ and http://www.eclipse.org/modeling/mdt/papyrus/)

6.7.1 What is Eclipse and the Eclipse Foundation?
Eclipse is an open source community, whose projects are focused on building an open development platform comprised of extensible frameworks, tools and runtimes for building, deploying and managing software across the lifecycle. The Eclipse Foundation is a not-for-profit, member supported corporation that hosts the Eclipse projects and helps cultivate both an open source community and an ecosystem of complementary products and services.

The Eclipse Project was originally created by IBM in November 2001 and supported by a consortium of software vendors. The Eclipse Foundation was created in January 2004 as an independent not-for-profit corporation to act as the steward of the Eclipse community. The independent not-for-profit corporation was created to allow a vendor neutral and open, transparent community to be established around Eclipse. Today, the Eclipse community consists of individuals and organizations from a cross section of the software industry.

In general, the Eclipse Foundation provides four services to the Eclipse community: 1) IT Infrastructure, 2) IP Management, 3) Development Process, and 4) Ecosystem Development. Full-time staff are associated with each of these areas and work with the greater Eclipse community to assist in meeting the needs of the stakeholders.

Eclipse has several Resources and plug-ins available to download.
6.7.2 A Unique Model for Open Source Development

The Eclipse Foundation has been established to serve the Eclipse open source projects and the Eclipse community. As an independent not-for-profit corporation, the Foundation and the Eclipse governance model ensures no single entity is able to control the strategy, policies or operations of the Eclipse community.

The Foundation is focused on creating an environment for successful open source projects and to promote the adoption of Eclipse technology in commercial and open source solutions. Through services like IP Due Diligence, annual release trains, development community support and ecosystem development, the Eclipse model of open source development is a unique and proven model for open source development.

6.7.3 About Papyrus

Papyrus is aiming at providing an integrated and user-consumable environment for editing any kind of EMF model and particularly supporting UML and related modeling languages such as SysML and MARTE. Papyrus provides diagram editors for EMF-based modeling languages amongst them UML 2 and SysML and the glue required for integrating these editors (Gmf-based or not) with other MBD and MDSD tools.

Papyrus also offers a very advanced support of UML profiles that enables users to define editors for DSLs based on the UML 2 standard. The main feature of Papyrus regarding this latter point is a set of very powerful customization mechanisms which can be leveraged to create user-defined Papyrus perspectives and give it the same look and feel as a "pure" DSL editor.

6.8 Isight & the SIMULIA Execution Engine

(Source data: http://www.3ds.com/products/simulia/portfolio/isight-simulia-execution-engine/overview/)
SIMULIA and Isight are produced by Dassault Systèmes
The tool information on Isight and SIMULIA doesn’t specifically call out MBSE but their goal is to integration information across system engineering work products and models.

6.8.1 Dassault Systèmes Company Information

Dassault Systèmes, the 3D Experience Company, provides business and people with virtual universes to imagine sustainable innovations. Its world-leading 3D design software, 3D Digital Mock-Up and Product Lifecycle Management (PLM) solutions transform the way products are designed, produced, and supported. (http://www.3ds.com/company/about-dassault-systemes/our-vision/)

6.8.2 SIMULIA Overview

SIMULIA enables collaboration on performing virtual tests and meeting performance requirements. Its portfolio provides powerful tools that enable fast, accurate performance studies on parts, assemblies, components, and products designed with V6. It also enables organizations to capture their simulation knowledge, deploy approved methods, manage applications, and share simulation results to enable collaboration and accurate performance-based decisions.
6.8.3 Isight & SEE Overview

In today’s complex product development and manufacturing environment, designers and engineers are using a wide range of software tools to design and simulate their products. Often, chained simulation process flows are required in which the parameters and results from one software package are needed as inputs to another package and the manual process of entering the required data can reduce efficiency, slow product development, and introduce errors in modeling and simulation assumptions.

SIMULIA provides market-leading solutions that improve the process of leveraging the power of various software packages. Isight and the SIMULIA Execution Engine (formerly Fiper) are used to combine multiple cross-disciplinary models and applications together in a simulation process flow, automate their execution across distributed compute resources, explore the resulting design space, and identify the optimal design parameters subject to required constraints.

Our proven simulation automation and optimization solutions enable engineering teams to:

- Drastically reduce design cycle time through integrating workflow processes in an automated environment
- Deliver more reliable, better-quality products through accelerated evaluation of design alternatives
- Lower hardware investments through effective use of legacy systems and more efficient job distribution
- Eliminate the bottlenecks of ineffective communication by enabling secure design collaboration among partners
7.0 Companies working to implement MBSE
This section is not intended to list ALL companies working to implement MBSE but it is to highlight a few who have recently made presentations or released papers available for open use.

The presentations from each of the companies covered the multi-disciplinary engineering modeling approaches, key practices, and integration challenges of MBSE.

7.1 Lockheed Martin
Digital Tapestry, presented by Chris Oster
Link to presentation:

System Modeling, presented by John Watson
Link to Presentation:

7.2 Deere & Co
Transition to Model Based Embedded System Development, presented by Jim Ross
Link to presentation:

7.3 Rockwell Collins
Model-based Manufacturing Modeling, presented by Greg Pollari
Link to presentation:

7.4 Raytheon
Cost/Logistics Modeling, presented by Chris Adams
Link to presentation:

7.5 ATA Engr
Multi-Disciplinary Mechanical Modeling, Management and Exchange, presented by Dr. Clark Briggs
Link to presentation:
Additional companies pursuing MBSE and have presented at other INCOSE events.

7.6 JPL
Survey of Model-Based Systems Engineering (MBSE) Methodologies
Paper written by Jeff A. Estefan in 2007 and revised in 2008

Advancing the Practice of Systems Engineering at JPL
Paper written by P. A. “Trisha” Jansma and Ross M. Jones talks about MBSE in section 5: “The SEA PPTT Element also identifies existing and emerging tools and technologies that support systems engineering activities, especially those that provide model-based systems engineering capabilities. Model-based systems engineering (MBSE) is the application of scientific and engineering efforts to transform an operational need into a description of system performance parameters and a system configuration by creating executable, explicit representations (model) of a system in order to predict, simulate and explain the resultant behavior of the system from the structure.” http://trs-new.jpl.nasa.gov/dspace/bitstream/2014/38979/1/05-3271.pdf

JPL developed State Analysis Methodology (covered in Section 4.4)

JPL is and has been involved in various other projects.

7.7 Modern Technology Systems Inc.
(http://www.mtsi-va.com/)

Using Model Based System Engineering (MBSE) in Flight Test
Presented by Frank C. Alvidrez, Senior NetCentric Flight Test Engineer

7.8 Northrop Grumman
(www.northropgrumman.com/)
Model Based Systems Engineering (MBSE) Process Using SysML for Architecture Design, Simulation and Visualization
Presented by Gundars Osvalds Senior Principal Enterprise Architect
Link to presentation: http://chesapeakebayao.org/documents/AOC%20MBSE.pdf

Northrop Grumman Aerospace Systems
Paper written by James R van Gaasbeek, Systems Engineer

“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.”
This paper addresses both the potential (and/or current) benefits that may be realized by Model Based 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).


7.9 U.S. Army Research, Development and Engineering Command
Deployment of MBSE processes using SysML
Link to presentation: http://www.dtic.mil/ndia/2010systemengr/WednesdayTrack5-11091Tritsch.pdf

7.10 SAF Consulting
Mr. Sandy Friedenthal has been huge contributor to establishing and advancing the MBSE initiative.
(Source data: www.incose.org/symp2011/.../TO_MBSE_Abstract_Bio_Friedenthal.pdf)

Mr. Friedenthal recently became an independent consultant after a career at Lockheed Martin. As a Lockheed Martin Fellow, he led an initiative to advance the practice of Model Based Systems Development (MBSD) across the corporation. In this capacity, he was responsible for developing and implementing strategies to institutionalize the practice of MBSD across the LM Business Units. This included codifying and piloting the practice of MBSD, developing the tools infrastructure, and providing training. Mr Friedenthal also provided direct model based systems engineering support to multiple programs across the Business.

Sandy Friedenthal's experience includes the application of systems engineering throughout the system life cycle from conceptual design, through development and production on a broad range of systems including missile systems, electro-optical navigation and targeting systems, and information systems. He has been a systems engineering department manager responsible for ensuring systems engineering processes are implemented on the programs, and enhancing overall systems engineering capability. He has been a lead developer of advanced systems engineering processes and methods. Mr.Friedenthal also has been a leader of the Industry Standards effort through the Object Management Group and INCOSE to develop the Systems Modeling Language (OMG SysMLT) that was adopted in 2006. He co-authored the book "A Practical Guide to SysML".
8.0 Universities

This section is not intended to list ALL Universities involved with pursuing MBSE initiatives but it is to highlight a few who are actively involved and information is readily available online.

The following are some universities that have been on the forefront of MBSE or through some internet searching has MBSE programs/special classes established to cover MBSE Methodology.

8.1 Georgia Tech  
([http://mbse.gatech.edu/](http://mbse.gatech.edu/))  
Established a new MBSE Center on campus.

Model-Based Systems Engineering (MBSE) Center  
Previously the Product & Systems Lifecycle Management Center (PSLM Center)  
In fall 2010 the PSLM Center was transformed into the Model-Based Systems Engineering (MBSE) Center consistent with our updated mission focus. Our new website is being formed here: www.mbse.gatech.edu.

From 2004 to 2010 the Product & Systems Lifecycle Management Center (PSLM Center) was the Georgia Tech focal point for product lifecycle management (PLM) and systems of systems (SoS) research, education, and outreach.

8.2 University of Alabama - Huntsville  
Established a new class offered on MBSE and have been involved with various presentations covering MBSE, one is linked below.

Introduction to Model-Based Systems Engineering  
Gain an understanding of the scope, implications, and practice of Model-Based Systems Engineering (MBSE) that will equip you to effectively implement MBSE for your projects. Learn about the current state of MBSE, important developments, and how MBSE links to SE architecture frameworks, process standards, and notation languages. Use case studies and MBSE tools to gauge how MBSE can bring rigor to SE processes and facilitate better communication.

Huntsville Simulation Based Approaches for Systems Engineering  
Presented by: Mikel D. Petty, Ph.D. Center for Modeling, Simulation, and Analysis  

8.3 University of Maryland & The Institute for Systems Research  
The Institute for Systems Research is an internationally recognized leader in systems research. We emphasize advanced strategies for high-level synthesis and analysis of complex, multidisciplinary engineering systems, using model-based systems engineering techniques.

The Institute for Systems Research has a Masters of Science in System Engineering that introduces the MBSE methods & tools.  
([http://www.isr.umd.edu/MSSE/intro.htm](http://www.isr.umd.edu/MSSE/intro.htm))
The program draws on the engineering, computer science, and management experience of University of Maryland faculty, and makes optimum use of the university’s advanced facilities, including commercial and open-source software tools for model-based systems engineering (MBSE). In addition, the University of Maryland’s location close to Washington, D.C., offers advantages and convenience for those seeking employment in, or already employed by the federal government and the many private industries supporting it.

8.4 Technion Israel Institute of Technology

Technion, from what I can tell from internet research, is heavily involved with MBSE initiatives but I wasn’t able to find any classes or degrees offered associated with MBSE at this time.

Here are some links I was able to find:

Prof. Dov Dori who has presented and authored several MBSE related topics at INCOSE events, head of the Enterprise Systems Modeling Laboratory at Technion (http://esml.iem.technion.ac.il/?page_id=356)

Here is one of the papers that Prof. Dov Dor has co-authored.  
*Is There a Complete Project Plan? A Model-Based Project Planning Approach*  
Paper written by: Amira Sharon (from Technion), Olivier L. de Weck (from MIT), Dov Dori (from Technion)  

8.5 Massachusetts Institute of Technology

MIT, from what I can tell from internet research, is heavily involved with MBSE initiatives but I wasn’t able to find any classes or degrees offered associated with MBSE at this time. They did hold a special event last year to cover Model Based Systems Engineering.  

8.6 Stevens Institute of Technology

Stevens Institute of Technology, from what I can tell from internet research, is heavily involved with MBSE initiatives but I wasn’t able to find any classes or degrees offered associated with MBSE at this time.

Dr. Robert J. Cloutier, PhD, a professor at Stevens Institute of Technology, has made some INCOSE MBSE presentations and authored some papers on MBSE.  
http://sse.stevens.edu/nc/people/faculty/faculty/181/

Robert Cloutier is an Associate Professor in the School of Systems and Enterprises at Stevens Institute of Technology. Rob has over 20 years experience in systems engineering & architecting, software engineering, and project management in both commercial and defense industries. His research interests include model based systems engineering and systems architecting using UML/SysML, reference architectures, systems engineering patterns, and architecture management.

*Applying Systems Engineering Modeling Language to System Effort Estimation Utilizing Use Case Points*  
Paper written by Mary A. Bone and Dr. Robert Cloutier
This paper proposes that as model based systems engineering begins to play a larger role in system development it may be utilized in estimating systems effort, making model based systems engineering more valuable to system engineering. Full abstract: http://www.incose.org/symp2011/download/abstract/Paper_ID179_Abstract.pdf

8.7 Technical University of Demark
(Source data: http://www.imm.dtu.dk/English/Industrial_collaboration/CCMBSE.aspx)
Technical University of Demark has a center on MBSE.

DTU Informatics – Department of Informatics Mathematical Modeling Competence Center in Model-based Software Engineering
The Competences Center in Model-based Software Engineering is an initiative of the Software Engineering Section of DTU Informatics in order to bring together industry and academia in the field of Model-based Software Engineering and to boost modern software engineering technologies in practice.
9.0 DoD Systems 2020 Vision

The United States Department of Defense (DoD) Office of the Assistant Secretary of Defense for Research and Engineering is sponsoring a strategic initiative called “Systems 2020” with the objective of fundamentally changing the capabilities for the design, adaptation, and manufacture of defense systems. With the current agility of adversaries and the pace of emerging events, the typical practice of fielding systems which fulfill specific point requirements defined years before the system’s initial use is inadequate. The Department seeks systems which are far more flexible and adaptable to changing environments, without major redesign.

With the growth in system complexity, the Department needs to make a comprehensive push to develop new system engineering foundations for the next decade. New engineering tools and approaches are needed to develop and field interoperable systems. Compared to the systems of today, future systems will have to:

- Use a much greater development tempo
- Operate with greater performance and agility
- Operate in far more complex environments
- Address deep cost and reliability constraints

The Systems 2020 initiative will provide new capabilities and approaches for faster delivery of flexible and adaptive systems that are trusted, assured, reliable and interoperable. Challenges identified across the spectrum are:

1. Development takes too long.
2. Change takes too long.
3. Replacement takes too long.
4. The environment is highly uncertain and complex.
5. System complexity is growing.

Systems 2020 Vision per the Director of Defense Research & Engineering (DDR&E), Congressional Testimony, May 18, 2010 made the following claims to Systems 2020:

- “Systems 2020 will give the DoD S&T and manufacturing communities a decisive edge in the design and manufacture of complex systems….
- Like the commercial sector, the DoD needs to adapt to the growing complexity of our major acquisition systems while managing cost, schedule and performance risk…
- Our Very complex systems must be adaptable and interoperable with other fielded systems…
- Systems 2020 will develop new system engineering foundations in basic research, applied research, prototype development, and workforce development…..
- When realized, the new tools and processes will enable a number of new design, development, and manufacture capabilities across the DoD acquisition enterprise….”
9.1 Systems 2020 Objectives and Constraints

Objectives
- DEVELOP FAST: Reduce by 3x the time to acquisition of first article for systems and solutions
- FLEXIBLE: Reduce by 4x the time to implement planned and foreseen changes in systems
- ADAPTABLE: Embed within systems the ability for changes at the tactical edge, as the mission evolves in unplanned and unforeseen ways, e.g., IED threat

Constraints
- Achieve the objectives while maintaining or enhancing:
  - Trust and Assurance –Able to withstand exploitation before or after fielding, enabling the leveraging of global supply chains
  - Reliability –Across a range of changing operational conditions
  - Interoperability–Working with other systems to meet user needs

9.2 Integrated Game-Changing Solution Strategy

After studying several candidate research areas (e.g., agile methods, formal methods, requirements tools, test tools), the study concluded that four research and technology areas had the best prospects of producing game-changing approaches to achieve the Systems 2020 goals and objectives of DEVELOP FAST, FLEX and ADAPT. These are:

- **Model-Based Engineering (MBE):** Changing the traditional DoD requirements-delay-surprise acquisition game. MBE applies product, process, property, environment, and mission models to ensure rapid, concurrent, and integrated development of DoD systems that can adapt to foreseeable and unforeseeable change.

- **Platform Based Engineering (PBE):** Changing the traditional DoD stovepipe acquisition game. The complement of MBE for portfolios or product lines, PBE invests in determining DoD-domain commonalities and variabilities, develops product-line architectures that package the commonalities into physical and informational platforms, and provides plug-compatible interfaces to the variable product line components.

- **Capability on Demand (COD):** Changing the traditional brittle DoD point-solution acquisition game. COD provides technology support for evolutionary acquisition strategies that combine short, stabilized build-to-specification increment developments with concurrent change anticipation, analysis, and self-adaptation. This amplifies the effects of MBE and PBE capabilities for rapid new-component generation and integration.

- **Trusted Systems Design (TSD):** Changing the traditional slow DoD acquire-certify-patch security assurance game. TSD includes up-front analysis and systems engineering of foreseeable threat patterns, uses MBE and PBE capabilities to build trust and assurance into DoD system architectures, and ensures that agile change adaptation fully addresses trust and assurance concerns.
These interrelated ideas will:
- Build on pockets of experience while pushing advanced design and manufacturing concepts
- Apply across system conception, design, manufacturing, deployment and evolution
- Provide opportunities to replace sequential development and fixed, single point user requirements

MBE, PBE, COD and Trusted Systems Designs are each covered in brief detail below.

### 9.2.1 Model Based Engineering (MBE)

As defined by the Office of the Deputy Assistant Secretary of Defense (ODASD) for the Systems 2020 initiative [ODASD, 2011], MBE is a model-based (vs. paper-based) approach to system design, capturing design choices and details, and executing the development of the system software and hardware. MBE enhances design reuse and speeds the implementation of design changes. Ideally, MBE allows programs to integrate the design of electronics, software, physical structure, mechanical systems, connectors, cabling, hydraulics, and all significant components into a virtual realization of a complex system.

The National Defense Industrial Association (NDIA) defines MBE as, An approach to engineering that uses models as an integral part of the technical baseline that includes the requirements, analysis, design, implementation, and verification of a capability, system, and/or product throughout the acquisition life cycle” [NDIA, 2011]. The NDIA MBE Subcommittee says that MBE “is an approach to engineering in which models:
- Are an integral part of the technical baseline
- Evolve throughout the acquisition life cycle
- Are integrated across all program disciplines (e.g., systems engineering, operations analysis, software engineering, hardware engineering, manufacturing, logistics, etc.)
- Can be shared and/or reused across acquisition programs, including between Government and Industry stakeholders.

Though some use MBE to describe only the computational models, the NDIA notes that, “Core to MBE is the integration of descriptive/design models with the computational models.” Model-Based Systems Engineering (MBSE) is a key part of the descriptive model.
As noted above (INCOSE) Systems Engineering (SE) Vision 2020 [INCOSE, 2007] defines 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. MBSE is part of a long-term trend toward model-centric approaches adopted by other engineering disciplines, including mechanical, electrical and software. In particular, MBSE is expected to replace the document-centric approach that has been practiced by systems engineers in the past and to influence the future practice of systems engineering by being fully integrated into the definition of systems engineering processes.

9.2.1.1 Related to DoD Systems 2020 Initiative
The application of modeling and simulation throughout the development process will foster more effective concept engineering and concurrent design, development, manufacturing, deployment and evolution. DoD looks to achieve Systems 2020 objectives through innovation in MBE area by:

- Virtual Design and Manufacturing (VLSI at higher levels)
- Manufacturing on Demand
- Test Driven Design
- Model Based Systems Engineering

MBE will enable real time assessment of system changes, robust evaluation of different approaches, and will impact DEVELOP FAST and FLEX areas of DoD Systems 2020 Initiative.

9.2.2 Platform Based Engineering (PBE)

Platform-Based Engineering (PBE) is a cost-effective, risk-mitigated system development approach that employs a common structure from which high-quality derivative products can be developed rapidly. PBE is especially effective in decreasing development cost, risks, and lead times while increasing product quality. Appropriately scoped, platforms simplify and accelerate the development of families of systems for a particular problem domain. They encompass domain-specific components and services that reflect the commonalities of systems in the domain (which can be configured as reusable physical or informational components), and variabilities across the domain (which need to be individually developed to achieve a domain product line), along with interface conventions that ensure that they can plug-and-play with the domain infrastructure and common components.

However, PBE has a potential downside. Locking into a platform strategy for the long term can severely limit an organization’s ability to evolve a product. It is this recognition that spurred the development of the adaptable PBE paradigm. Adaptable PBE offers the customary benefits of PBE without compromising the long-term evolvability of the system.


9.2.2.1 Related to DoD Systems 2020 Initiative
Applying architectural and automated design tools to the development of hybrid hardware, software, and networked systems as an enduring “platform” will evolve user capabilities. DoD looks to achieve Systems 2020 objectives through innovation in PBE area by:

- Architecture Patterns
- Intelligent Design Automation
9.2.3 Capability on Demand (COD)
(Source data: http://www.bitpipe.com/tlist/Capacity-on-Demand.html)

Capacity on demand (COD) is a purchasing option that allows companies to receive equipment with more computer processing, storage, or other capacity than the company needs at the time of purchase, and have that extra capacity remain unused and unpaid for until the company actually requires it. Vendors are promoting capacity on demand as a cost-effective and time-saving alternative to more traditional methods of upgrading. With COD, a vendor might provide a company with a fully-configured 24-processor computer server but only charge the company for the number of processors they actually use. The vendor provides the additional capacity hoping that when the company expands and needs more capacity, they will not look around elsewhere but will simply take advantage of the extra capacity the vendor had already provided. In this scenario, the company would contact the vendor to have the extra processors activated, and the vendor would bill them accordingly.

9.2.3.1 Related to DoD Systems 2020 Initiative

In the context of Systems 2020, Capability on Demand (COD) addresses the challenge of rapidly adapting fielded systems to unforeseeable new threats and opportunities. Keys to rapid adaptability include capabilities to rapidly detect and analyze new threat and opportunity trends and patterns, and capabilities to rapidly reconfigure the system to counter the threat or to capitalize on the opportunity. DoD looks to achieve System-2020 objectives through innovation in COD area by:

- Service Oriented Design and Development Methodologies
- Self Healing Systems
- Adaptive Algorithms
- Autonomic Computing

COD will allow fielded systems to rapidly respond to a changing environment as the mission evolves in unplanned, unforeseen ways, and will impact ADAPT area of DoD Systems 2020 Initiative.

9.2.4 Trusted Systems Design (TSD)
(Source data: http://en.wikipedia.org/wiki/Trusted_system)

In the security engineering subspecialty of computer science, a trusted system is a system that is relied upon to a specified extent to enforce a specified security policy. As such, a trusted system is one whose failure may break a specified security policy. Trusted systems are used for the processing, storage and retrieval of sensitive or classified information.

Central to the concept of U.S. Department of Defense-style "trusted systems" is the notion of a "reference monitor", which is an entity that occupies the logical heart of the system and is responsible for all access control decisions. Ideally, the reference monitor is (a) tamperproof, (b) always invoked, and (c) small enough to be subject to independent testing, the completeness of which can be assured. Per the U.S. National Security Agency's 1983 Trusted Systems Design Manual.
Computer System Evaluation Criteria (TCSEC), or "Orange Book", a set of "evaluation classes" were defined that described the features and assurances that the user could expect from a trusted system.

The highest levels of assurance were guaranteed by significant system engineering directed toward minimization of the size of the trusted computing base (TCB), defined as that combination of hardware, software, and firmware that is responsible for enforcing the system's security policy.

Because failure of the TCB breaks the trusted system, higher assurance is provided by the minimization of the TCB. An inherent engineering conflict arises in higher-assurance systems in that, the smaller the TCB, the larger the set of hardware, software, and firmware that lies outside the TCB. This may lead to some philosophical arguments about the nature of trust, based on the notion that a "trustworthy" implementation may not necessarily be a "correct" implementation from the perspective of users' expectations.

9.2.4.1 Related to DoD Systems 2020 Initiative

A key to trusted system design is the portion of Model Based Engineering that supports the analysis of tradeoffs between trust and assurance and other system objectives such as performance (avoiding too much security system overhead), reliability (avoiding single points of failure), usability (proliferation of keys and passwords), and interoperability (over-constraining information sharing) and will help with designing secure systems from unsecured vendors and subsystems. DoD looks to achieve System-2020 objectives through innovation in TSD area by:

- Composite Health Monitoring Systems
- Design for Test Methodologies
- Feedback Control (observability theory)

Trusted Systems Design will allow us to take advantage of innovation in the global supply chain, while ensuring that our systems operate as intended. Composing assured systems from COTS will allow speedy adoption of COTS for the war-fighter and will impact DEVELOP FAST area of DoD Systems 2020 Initiative.
10.0 Lean Engineering
(Source data: http://en.wikipedia.org/wiki/Lean_manufacturing)

Lean manufacturing, lean enterprise, or lean production, often simply, "Lean," is a production practice that considers the expenditure of resources for any goal other than the creation of value for the end customer to be wasteful, and thus a target for elimination. Working from the perspective of the customer who consumes a product or service, "value" is defined as any action or process that a customer would be willing to pay for.

Essentially, lean is centered on preserving value with less work. Lean manufacturing is a management philosophy derived mostly from the Toyota Production System (TPS) (hence the term Toyotism is also prevalent) and identified as "Lean" only in the 1990s.

10.1 Lean implementation develops from TPS

The discipline required to implement Lean and the disciplines it seems to require are so often counter-cultural that they have made successful implementation of Lean a major challenge. Some would say that it was a major challenge in its manufacturing 'heartland' as well.
Implementations under the Lean label are numerous and whether they are Lean and whether any success or failure can be laid at Lean's door is often debatable. Individual examples of success and failure exist in almost all spheres of business and activity and therefore cannot be taken as indications of whether Lean is particularly applicable to a specific sector of activity. It seems clear from the "successes" that no sector is immune from beneficial possibility.

Lean is about more than just cutting costs in the factory. One crucial insight is that most costs are assigned when a product is designed, (see Genichi Taguchi). Often an engineer will specify familiar, safe materials and processes rather than inexpensive, efficient ones. This reduces project risk, that is, the cost to the engineer, while increasing financial risks, and decreasing profits. Good organizations develop and review checklists to review product designs.

Companies must often look beyond the shop-floor to find opportunities for improving overall company cost and performance. At the system engineering level, requirements are reviewed with marketing and customer representatives to eliminate those requirements that are costly. Shared modules may be developed, such as multipurpose power supplies or shared mechanical components or fasteners. Requirements are assigned to the cheapest discipline. For example, adjustments may be moved into software, and measurements away from a mechanical solution to an electronic solution. Another approach is to choose connection or power-transport methods that are cheap or that used standardized components that become available in a competitive market.

10.2 Differences from TPS

Whilst Lean is seen by many as a generalization of the Toyota Production System into other industries and contexts there are some acknowledged differences that seem to have developed in implementation.

1. **Seeking profit** is a relentless focus for Toyota exemplified by the profit maximization principle (Price – Cost = Profit) and the need, therefore, to practice systematic cost reduction (through TPS or otherwise) to realize benefit. Lean implementations can tend to de-emphasize this key measure and thus become fixated with the
implementation of improvement concepts of "flow" or "pull". However, the emergence of the "value curve analysis" promises to directly tie lean improvements to bottom-line performance measurements.

2. **Tool orientation** is a tendency in many programs to elevate mere tools (standardized work, value stream mapping, visual control, etc.) to an unhealthy status beyond their pragmatic intent. The tools are just different ways to work around certain types of problems but they do not solve them for you or always highlight the underlying cause of many types of problems. The tools employed at Toyota are often used to expose particular problems that are then dealt with, as each tool's limitations or blind-spots are perhaps better understood. So, for example, Value Stream Mapping focuses upon material and information flow problems (a title built into the Toyota title for this activity) but is not strong on Metrics, Man or Method. Internally they well know the limits of the tool and understood that it was never intended as the best way to see and analyze every waste or every problem related to quality, downtime, personnel development, cross training related issues, capacity bottlenecks, or anything to do with profits, safety, metrics or morale, etc. No one tool can do all of that. For surfacing these issues other tools are much more widely and effectively used.

3. **Management technique rather than change agents** has been a principle in Toyota from the early 1950s when they started emphasizing the development of the production manager's and supervisors' skills set in guiding natural work teams and did not rely upon staff-level change agents to drive improvements. This can manifest itself as a "Push" implementation of Lean rather than "Pull" by the team itself. This area of skills development is not that of the change agent specialist, but that of the natural operations work team leader. Although less prestigious than the TPS specialists, development of work team supervisors in Toyota is considered an equally, if not more important, topic merely because there are tens of thousands of these individuals. Specifically, it is these manufacturing leaders that are the main focus of training efforts in Toyota since they lead the daily work areas, and they directly and dramatically affect quality, cost, productivity, safety, and morale of the team environment. In many companies implementing Lean the reverse set of priorities is true. Emphasis is put on developing the specialist, while the supervisor skill level is expected to somehow develop over time on its own.

### 10.3 Lean goals and strategy

The espoused goals of Lean manufacturing systems differ between various authors. While some maintain an internal focus, e.g. to increase profit for the organization, others claim that improvements should be done for the sake of the customer. Some commonly mentioned goals are:

- **Improve quality**: To stay competitive in today's marketplace, a company must understand its customers' wants and needs and design processes to meet their expectations and requirements.
- **Eliminate waste**: Waste is any activity that consumes time, resources, or space but does not add any value to the product or service.
  - Transport (moving products that are not actually required to perform the processing)
  - Inventory (all components, work in process and finished product not being processed)
  - Motion (people or equipment moving or walking more than is required to perform the processing)
  - Waiting (waiting for the next production step)
Overproduction (production ahead of demand)
- Over Processing (resulting from poor tool or product design creating activity)
- Defects (the effort involved in inspecting for and fixing defects)

Taking the first letter of each waste, the acronym "TIM WOOD" is formed. This is a common way to remember the wastes. The other alternative name that can used to remember is "DOT WIMP".

- Reduce time: Reducing the time it takes to finish an activity from start to finish is one of the most effective ways to eliminate waste and lower costs.
- Reduce total costs: To minimize cost, a company must produce only to customer demand. Overproduction increases a company’s inventory costs because of storage needs.

The strategic elements of Lean can be quite complex, and comprise multiple elements. Four different notions of Lean have been identified:
1. Lean as a fixed state or goal (Being Lean)
2. Lean as a continuous change process (Becoming Lean)
3. Lean as a set of tools or methods (Doing Lean/Toolbox Lean)
4. Lean as a philosophy (Lean thinking)

10.4 Steps to achieve lean systems
The following steps should be implemented to create the ideal lean manufacturing system:
1. Design a simple manufacturing system
2. Recognize that there is always room for improvement
3. Continuously improve the lean manufacturing system design

10.4.1 Design a simple manufacturing system
A fundamental principle of lean manufacturing is demand-based flow manufacturing. In this type of production setting, inventory is only pulled through each production center when it is needed to meet a customer's order. The benefits of this goal include:
- decreased cycle time
- less inventory
- increased productivity
- increased capital equipment utilization

10.4.2 There is always room for Improvement
The core of lean is founded on the concept of continuous product and process improvement and the elimination of non-value added activities. "The Value adding activities are simply only those things the customer is willing to pay for, everything else is waste, and should be eliminated, simplified, reduced, or integrated" (Rizzardo, 2003). Improving the flow of material through new ideal system layouts at the customer's required rate would reduce waste in material movement and inventory.

10.4.3 Continuously Improve
A continuous improvement mindset is essential to reach a company's goals. The term "continuous improvement" means incremental improvement of products, processes, or services over time, with the goal of reducing waste to improve workplace functionality, customer service, or product performance (Suzaki, 1987).

Stephen Shortell (Professor of Health Services Management and Organizational Behavior – Berkeley University, California) states:-
"For improvement to flourish it must be carefully cultivated in a rich soil bed (a receptive organization), given constant attention (sustained leadership), assured the right amounts of light (training and support) and water (measurement and data) and protected from damaging."

10.5 How can Lean connect with MBSE
(Source data: http://www.dtic.mil/ndia/2008systems/7052OLson_v2.pdf)
There are many industry engineering problems. Systems engineering needs to focus on improving engineering problems. Model and architecture based systems engineering for lean results. Models and architectures are powerful tools to help improve engineering and obtain measurable results. The future of architectures and models (MBSE initiative) is industry standards and tools.

10.5.1 Why focus on Models?
Models are very power because they:
- Are graphical (a picture is worth a 1000 words) and can be powerful communication tools
- Can scale up to complex systems and provide a tool to analyze complex relationships and dependencies
- Promote reuse (e.g., products components, requirements designs, test, interfaces, etc.) can improve productivity and quality
- Can be represented in an automated tool and simulated
Models are abstractions of reality constructed for a useful purpose consisting of:
- Formal notations and rules for representations
- Models components or building blocks
- Ways to model interfaces, interdependencies, and other relationships among the model components
A few Modeling examples: (which are all types of models/inputs needed for MBSE)
- Behavioral
- Structural
- Functional
- Process

10.5.2 Why focus on Architecture?
Architectures are very power because they:
- Are graphical (a picture is worth a 1000 words) and can be powerful communication tools
- Provide a framework of how components are related (e.g., interfaces, interdependencies relationships) and how components fit together
- Promote reuse (e.g., products components, requirements designs, test, interfaces, etc.) can improve productivity and quality
- Can be modeled in an automated tool
Architectures consist of:
- Components
- Interfaces, interdependencies, and other relationships among components
- Ordering and rules putting components together
11.0 Look ahead where we are going?
(Source data: www.incose.org/ProductsPubs/pdf/SEVision2020_20071003_v2_03.pdf)

11.1 Influence of systems engineering principles in a technical society
By 2020, it will acknowledged that the systems engineering curriculum is an appropriate academic foundation from which to address multi-disciplinary problem solving, such as the implications of technology on the environment and society, drug abuse prevention, national security, crime prevention, urban expansion, infrastructure development, etc.

Systems engineering educators and researchers need to participate in the application of systems thinking to governmental and large scale societal problem solving.

11.2 Innovative approaches toward systems engineering education delivery
In 2020, web-enabled information technologies will be commonplace. Widespread globalization of education programs, with no national boundaries, will be the norm. Information “chunking” will be used extensively in education and training. This will affect systems engineering in two ways.

1. Systems engineering education itself will be “chunked” to enable just-in-time delivery to a variety of consumers.
2. Information chunking, in which a basic unit of information is customized for presentation to specified audiences, will become a required systems engineering skill.

Use of technology (such as simulation, visualization, or gaming) will create major innovations for systems engineering education. Building on the computer gaming culture, the systems engineering community will benefit by creating interactive games that require systems thinking (e.g. model cities development, survival games involving multiple disciplines, economic decision-making). Faculties will encourage and support faculty members to keep pace with new knowledge and use of advanced technologies.