Chapter 1 Systems Engineering Overview

The systems engineering management process is intended to ensure that development cost, schedule, and technical performance objectives are met. Typical management activities include planning the technical effort, monitoring technical performance, managing risk, and controlling the system technical baseline.

The System Specification and Design process is used to specify system requirements that will meet the needs of the stakeholders.

- It then allocates the requirements to the components of the system
- The components are designed, implemented, and tested to ensure they satisfy the requirements
- The System Integration and Test process includes activities to integrate the components into the system and verify that the system satisfies its requirements
- Maintain traceability from the system goals to the system and component requirements and verification results to ensure that requirements and stakeholder needs are addressed
- Perform analysis to evaluate and select a preferred system solution that satisfies the system requirements and maximizes the effectiveness measures
- Synthesize alternative system solutions by partitioning the system design into components that can satisfy the system requirements

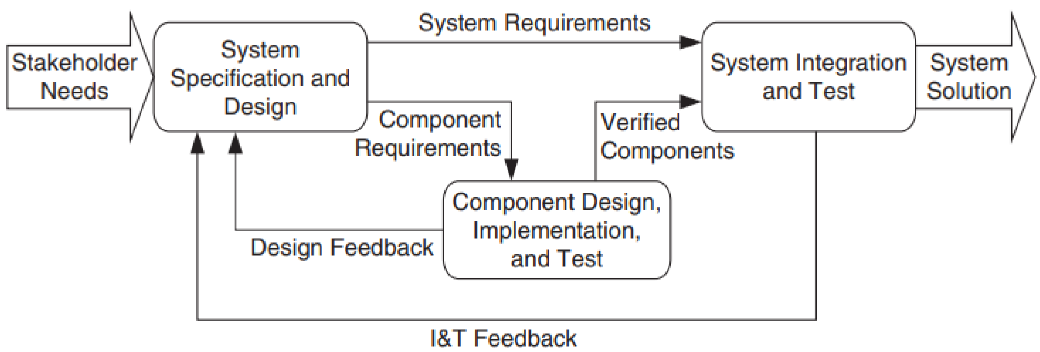

FIGURE 1.1

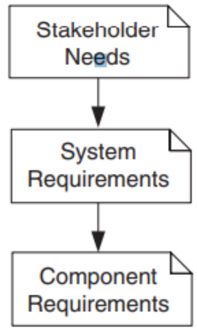

FIGURE 1.7

Stakeholder needs flow down to system and component requirements.

Stakeholders

- addressing the needs of other stakeholders who may be affected throughout the system's lifecycle
- Less obvious stakeholders are organizations and governments that express their needs via laws, regulations, and standards
- Analysis is performed to understand the needs of each stakeholder and to define effectiveness measures and target values
    - In terms of an automobile effectiveness measures may relate to the primary goal of addressing the transportation needs, such as the availability of transportation
    - the time to reach a destination, safety, comfort, environmental impact
    - costs associated with purchasing and owning an automobile

system boundary

- gas pump and maintenance equipment, road

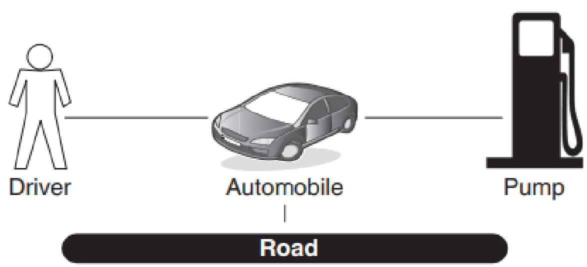

FIGURE 1.2

Defining the system boundary.

-

Functional requirements

- analyzing what the system must do to support its overall goals, such as functional requirements to meet transportation needs
- must perform functions related to accelerating, braking, and steering
- functional analysis identifies the inputs and outputs for each function
- must also be evaluated to determine the level of performance required for each function
- Additional requirements are specified to address other concerns of each stakeholder as defined by the system goals and effectiveness measures

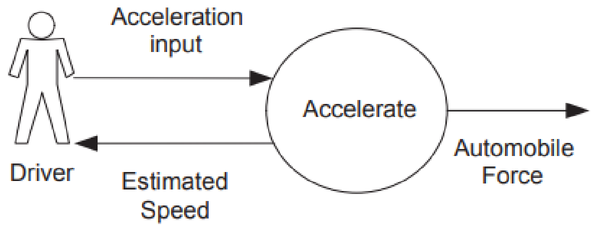

FIGURE 1.3

- Specifying the functional requirements.

System Requirements

- must be clearly traceable to stakeholder needs and validated to ensure that the requirements address those needs
- involves identifying system components and specifying the component requirements so that the system requirements will be met
- may involve first developing a logical system design that is independent of the technology used, and then a physical system design that reflects specific technology selections.
- system's physical components include the engine, transmission, differential, body, chassis, brakes, and so on
- Systems engineers should validate the assumptions that drive the constraints and perform the analysis to understand their impact on the design.

System Design Overview

- alternatives are evaluated to determine the system solution that achieves a balanced design while addressing multiple competing requirements
- The component requirements are input to the Component Design, Implementation, and Test process from Figure 1.1.
- Some components may be procured rather than developed, so designers need to understand the difference between what has been specified and what can be supplied
- The system test cases are defined to verify that the system satisfies its requirements

Multidisciplinary Systems Engineering Team

- The Requirements Team analyzes stakeholder needs, develops the concept of operations, and specifies and validates the system requirements
- The Architecture Team is responsible for synthesizing the system architecture by partitioning the system into components and defining their interactions and interconnections
- The Systems Analysis Team is responsible for performing the engineering analysis on different aspects of the system, such as performance and physical characteristics, reliability, maintainability, and cost, to provide the rationale for the technical specifications

- The Integration and Test Team is responsible for developing test plans and procedures and for conducting tests to verify the requirements are satisfied

Codifying Systems Engineering Practice Through Standards

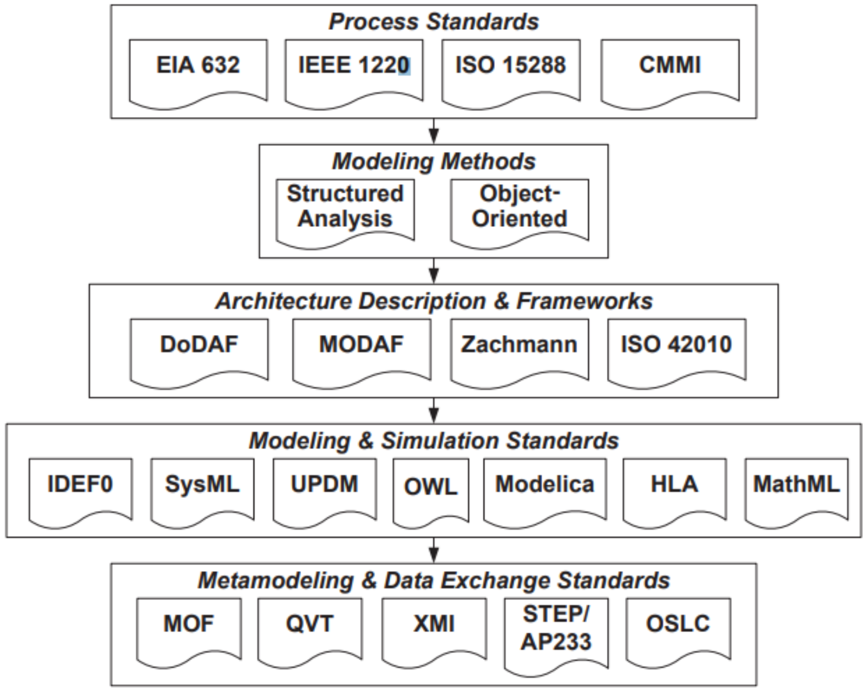

FIGURE 1.9

A partial systems engineering standards taxonomy.

-
- A particular systems engineering approach may implement one or more standards from each layer of this taxonomy
- The systems engineering process defines what activities are performed but does not generally give details on how they are performed.
- A systems engineering method describes how the activities are performed and the kinds of systems engineering artifacts that are produced
- The concept of operations defines what the system is intended to do from the user's perspective.
- Criteria for selecting a method include its ease of use, its ability to address the relevant systems engineering concerns, and the level of tool support.
- The C4ISR framework [16] was introduced in 1996 to provide a framework for architecting information systems for the US Department of Defense
- The OMG SysML specification—the subject of this book—was adopted in 2006 by the Object Management Group as a general-purpose graphical systems modeling language that extends the Unified Modeling Language (UML).
- Model and data exchange standards is a critical class of modeling standards that supports model and data exchange among tools.

Chapter 2 Model-Based Systems Engineering

- applies systems modeling as part of the systems engineering process described in Chapter 1 to support analysis, specification, design, and verification of the system being developed

Document-Based Systems Engineering Approach

- System Function Decomposition: Utilize drawing tools such as functional flow diagrams and schematic block diagrams to decompose system functions and allocate them to respective components.
- Engineering Trade Studies: Conduct analyses and trade studies across various disciplines to evaluate and optimize alternative designs, considering aspects like performance, reliability, safety, and mass properties.
- Requirements Traceability: Establish and maintain requirements traceability by linking specifications at different levels of the hierarchy. Utilize requirements management tools to parse and capture requirements in a database, ensuring traceability between requirements and design elements.
- Document-Based Approach Limitations: While document-based approaches can be rigorous, they suffer from limitations such as difficulties in assessing completeness, consistency, and relationships between different aspects (requirements, design, analysis). This can lead to poor synchronization, hindering system understanding, traceability, and change impact assessments. Additionally, maintaining and reusing information for evolving systems is challenging, potentially impacting cost, schedule, and system quality.

Model-Based Systems Engineering Approach

- MBSE is the formalized application of modeling to support system activities from conceptual design through development and later lifecycle phases.
- It utilizes models for system requirements, design, analysis, verification, and validation.
- Model elements represent requirements, design, test cases, design rationale, and their interrelationships.
- It specifies hardware and software components, their interconnections, interfaces, interactions, functions, and performance characteristics.
- Model elements are stored in a model repository and presented through diagrams with graphical symbols.
    - The repository enables querying, analysis, and document generation from the system model.

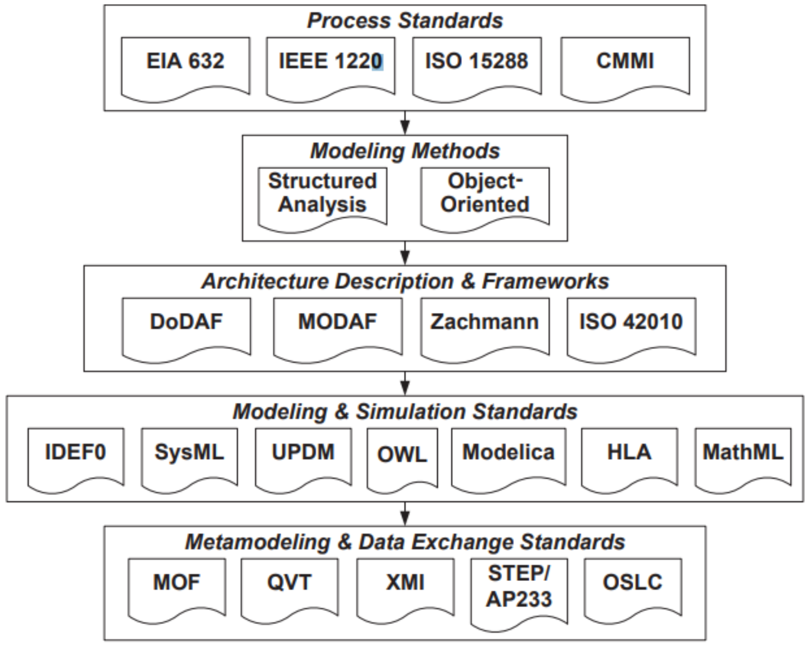

FIGURE 1.9

A partial systems engineering standards taxonomy.

- Model-Based Approach Benefits
    - Enhanced communication and shared understanding across development teams.
    - Reduced development risk through ongoing requirements validation and design verification.
    - Improved quality with more complete and traceable requirements.
- Increased Productivity with MBSE:
    - Faster and more comprehensive impact analysis of requirements and design changes.
    - Reuse of existing models for design evolution and automated document generation.
- Challenges and Considerations
    - Transitioning to MBSE requires upfront investment in processes, methods, tools, and training.
    - MBSE may coexist with document-based approaches during the transition, tailoring the approach based on project needs.
- Historical Use of Models in Document-Based Systems Engineering:
    - Various models and diagramming techniques have been employed in document-based systems engineering, including functional flow diagrams, behavior diagrams, and schematic block diagrams.
- Limitations of Traditional Model Use:
    - Models were typically used for specific analysis or selected aspects of system design, lacking integration into a cohesive overall system model.
    - Modeling activities were not fully integrated into the broader systems engineering process.

- Transition to Model-Based Systems Engineering (MBSE):
    - MBSE signifies a shift from controlling documentation to controlling the system model, integrating requirements, design, analysis, and verification into a unified framework.
- Improved Quality and Traceability:
    - More complete, unambiguous, and verifiable requirements.
    - Rigorous traceability between requirements, design, analysis, and testing.
    - Enhanced design integrity.
- Increased Productivity and Efficiency with MBSE:
    - Faster and more comprehensive impact analysis of requirements and design changes.
    - Exploration of trade-space, reuse of existing models, and automated document generation.
    - Reduction of errors and time during integration and testing.
- Challenges and Considerations in Transitioning to MBSE:
    - Need for upfront investment in processes, methods, tools, and training.
    - During the transition, a combination of MBSE and document-based approaches may be necessary.
    - Tailoring the approach and scoping the modeling effort is crucial based on project needs, especially for large and complex legacy systems.
- Modeling Principles
    - MODEL AND MBSE METHOD DEFINITION
        - Model Definition:
            - A model is a representation of concepts realized in the physical world, describing a domain of interest.
            - It is an abstraction that selectively includes details relevant to the intended use, expressed through various forms such as quantitative, logical, or geometric representations.
        - System Model in SysML:
            - A SysML system model is analogous to a building blueprint, specifying the behavior, structure, properties, constraints, and requirements of the system.
            - Model elements are stored in a repository and can be presented graphically or in other forms, integrating with analysis and design models for a comprehensive representation.
        - SysML as a Semantic Foundation:
            - SysML provides a semantic foundation defining model elements and their relationships within the system model.

        - It allows for graphical and tabular presentations, facilitating a holistic view of the system's attributes.
      - Model-Based Systems Engineering (MBSE) Method:
        - A method is a set of activities, techniques, conventions, and tools supporting processes.
        - An MBSE method implements systems engineering processes, producing a system model as a primary artifact.
  - THE PURPOSE FOR MODELING A SYSTEM
    - Clear Definition of Modeling Purpose:
      - The purpose of modeling a system must be precisely defined, considering the varied needs of stakeholders throughout the system's lifecycle.
    - Evolution of Intended Uses Across Lifecycle Phases:
      - Stakeholder needs and intended uses of the model evolve throughout the system development life cycle.
      - Early phases may focus on conceptual design and evaluation, while later phases emphasize component integration, verification, and detailed design.
    - Diverse Uses of System Modeling:
      - The intended use of system modeling aligns with specific systems engineering activities.
      - Examples include characterizing existing systems, specifying and designing new systems, validating requirements, conducting trade-offs, estimating costs, and supporting maintenance and diagnostics.
    - Alignment with Systems Engineering Activities:
      - The purpose of modeling is closely associated with systems engineering activities that the model is intended to support across the system's lifecycle.
      - The diverse uses of system modeling address various aspects, from concept representation to system verification and maintenance support.
- Model Validation:
  - Model Validation Process:
    - Model validation involves determining the accuracy of the model in representing the domain of interest for its intended use.
    - For analysis models, validation includes static checks, expert reviews of input data and assumptions, and comparison of model-generated results with real-world data.

    - Factors Influencing Model Accuracy:
        - The accuracy of a SysML system model depends on the quality of source information, validity of assumptions, and proper capture of information in the model.
        - Validation can be performed through model checks, expert reviews, and by using the system model as input for other analyses and simulations.
- Establishing Model Quality Criteria
    - Distinguishing Good Model and Design:
        - Distinguishing between a good model and a good design is crucial; a good model meets its intended use, while a good design satisfies requirements and incorporates quality design principles.
    - Assessment Questions for Model Quality:
        - Questions to assess a model's quality include defining its purpose, ensuring its scope is sufficient, and confirming its completeness, well-formedness, and consistency.
    - Model Scope Considerations:
        - Model scope involves balancing breadth, depth, and fidelity across different development phases, considering available resources, and aligning with the system's requirements.
    - Model Understandability:
        - Models should be interpretable by both humans and computers. Factors affecting understandability include information presentation, eliding nonessential details, layout, icon usage, and adherence to modeling conventions.
    - Consistency and Documentation:
        - Consistency is ensured by enforcing modeling language rules and constraints. Documentation of modeling conventions and standards, along with self-documenting practices, contributes to consistency.
    - Accuracy and Validation:
        - Accuracy is determined by model validation, relying on the quality of source information, validity of assumptions, and proper capture in the model. Assessments include expert reviews and consideration of other quality attributes.
    - Integration with Other Models:
        - The system model may need to integrate with various engineering models, necessitating a clear understanding of the modeling languages, tools, and methods used for information exchange.

- Model Based Metrics
    - Model Quality vs. Design Quality:
        - Distinction between a good model and good design; model quality criteria help meet intended use but may not explicitly reflect design quality.
    - Use of Model-Based Metrics:
        - Model-based metrics aid in assessing design quality, progress, and estimating effort throughout the development process.
    - Quality of the Design:
        - Metrics, including requirements satisfaction, verification, and technical performance measurement, assess the quality of a model-based system design.
    - Design Progress Assessment:
        - Metrics are defined to assess progress by considering design completeness, allocation of logical components, verification status, and integration into the system
    - Effort Estimation with COSYSMO:
        - The Constructive Systems Engineering Cost Model (COSYSMO) is employed for estimating the cost and effort in model-based systems engineering activities.
    - Identifying Sizing Parameters:
        - Sizing parameters, such as model elements, requirements, use cases, and constraints, are identified in the model for integration into the cost model.
    - Validation and Continuous Improvement:
        - Collection and validation of sizing and productivity data over time are essential for establishing accurate cost estimates and supporting continuous improvement in productivity.