V&V Concepts

Wednesday, September 23, 2015
10th Annual FAA V&V Summit

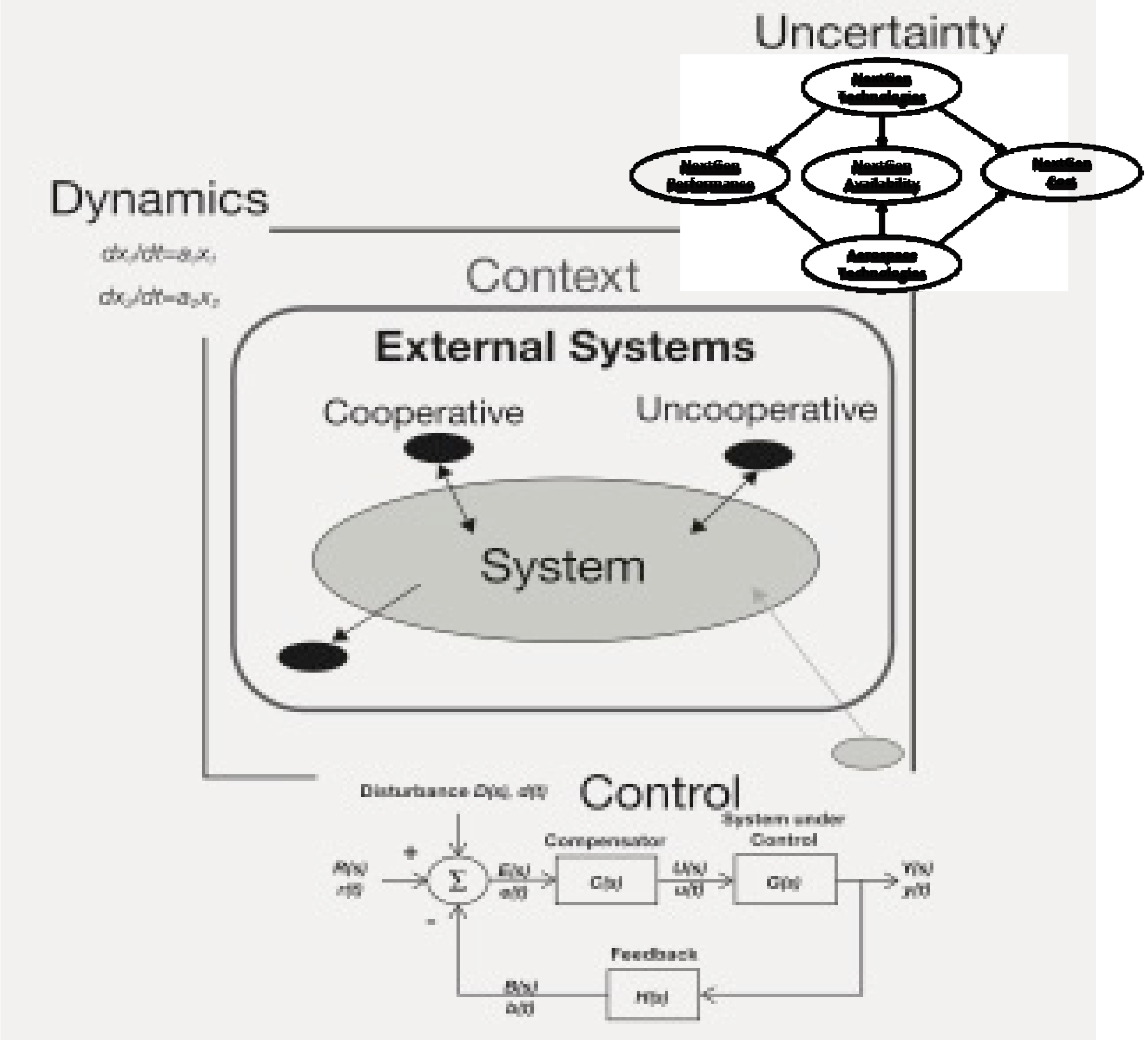
William D. Miller
Stevens Institute of Technology
INCOSE *INSIGHT* Magazine Editor-in-Chief
Former INCOSE Technical Director

wdmiller220@gmail.com

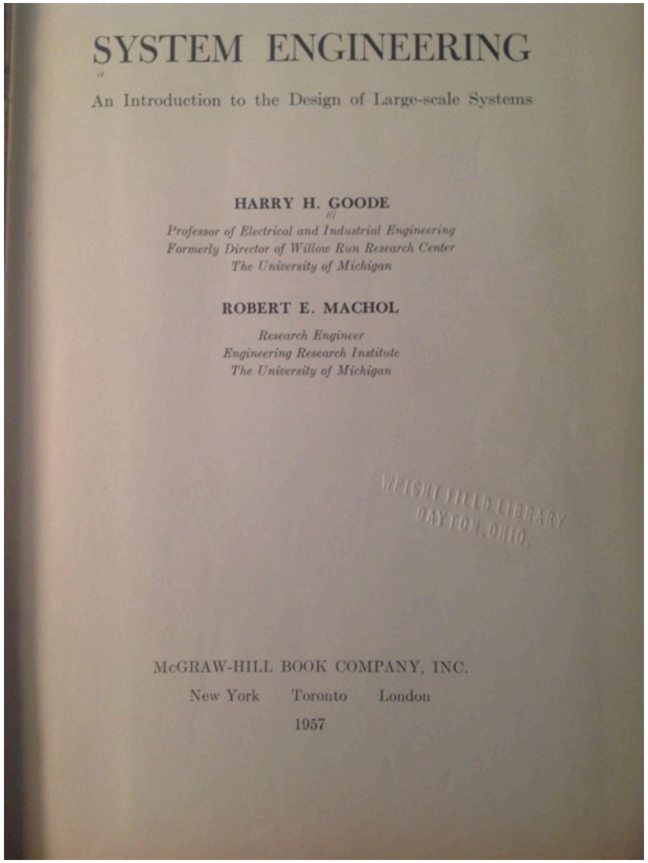
Focus

- Objective: Resilient Systems
- Interplay of Systems Engineering & Test for V&V
- Extending the Model-Based Systems Engineering Paradigm
  - to Model-Based Integration & Test
  - and to Model-Based V&V
- State of Practice, Successes and Challenges

Objective: Systems Resilient to Stochastic Inputs and Uncooperative Behavior



Back to the Future



SYSTEM ENGINEERING

An Introduction to the Design of Large-scale Systems

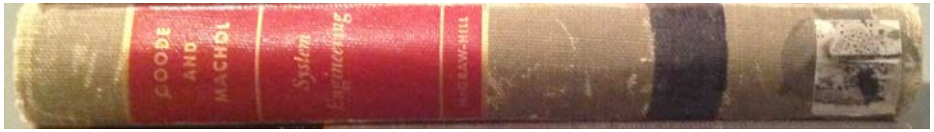
HARRY H. GOODE

Professor of Electrical and Industrial Engineering
Formerly Director of Willow Run Research Center
The University of Michigan

ROBERT E. MACHOL

Research Engineer
Engineering Research Institute
The University of Michigan

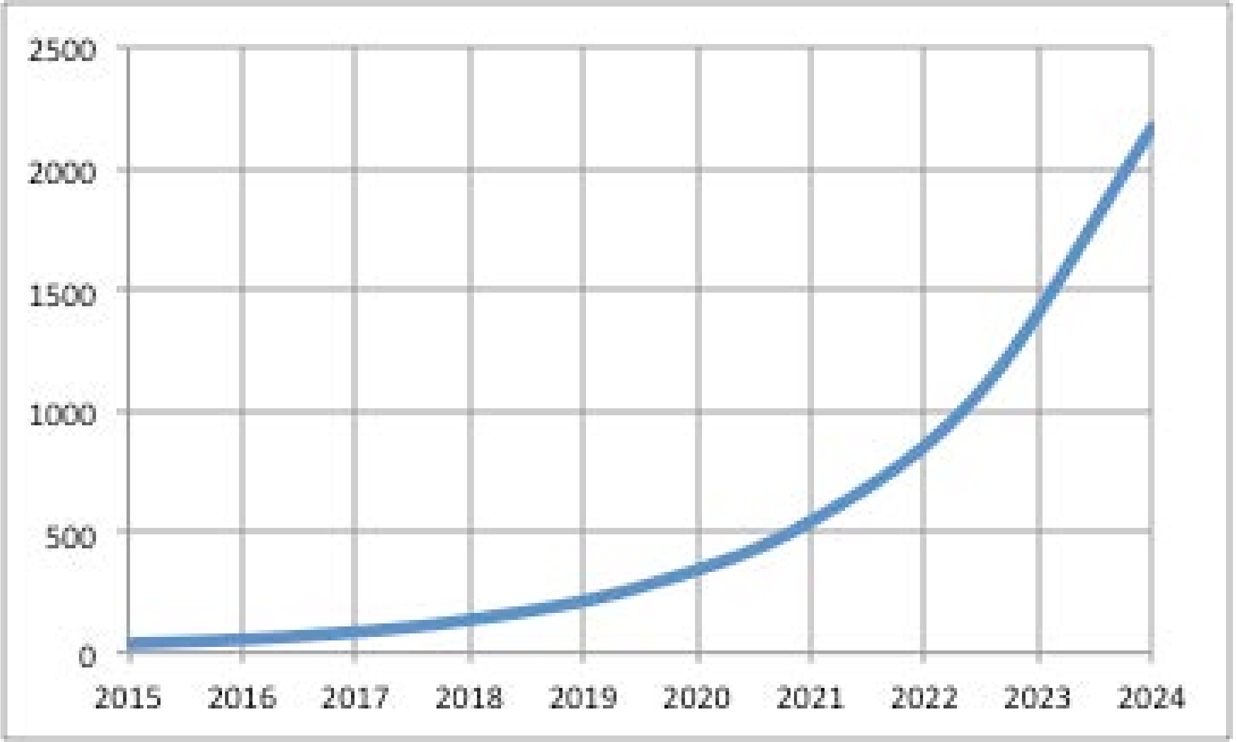
McGRAW-HILL BOOK COMPANY, INC.
New York Toronto London
1957



1. Introduction
2. Probability – The Basic Tool of Exterior System Design
3. Exterior System Design
4. Computers – The Basic Tool of Interior System Design
5. Interior System Design
    1. Inputs
    2. Classification of Systems
    3. The Single Thread
    4. High Traffic
    5. Competition
    6. Some Principles of System Design
6. Epilogue

Projected Growth in Computing Capability

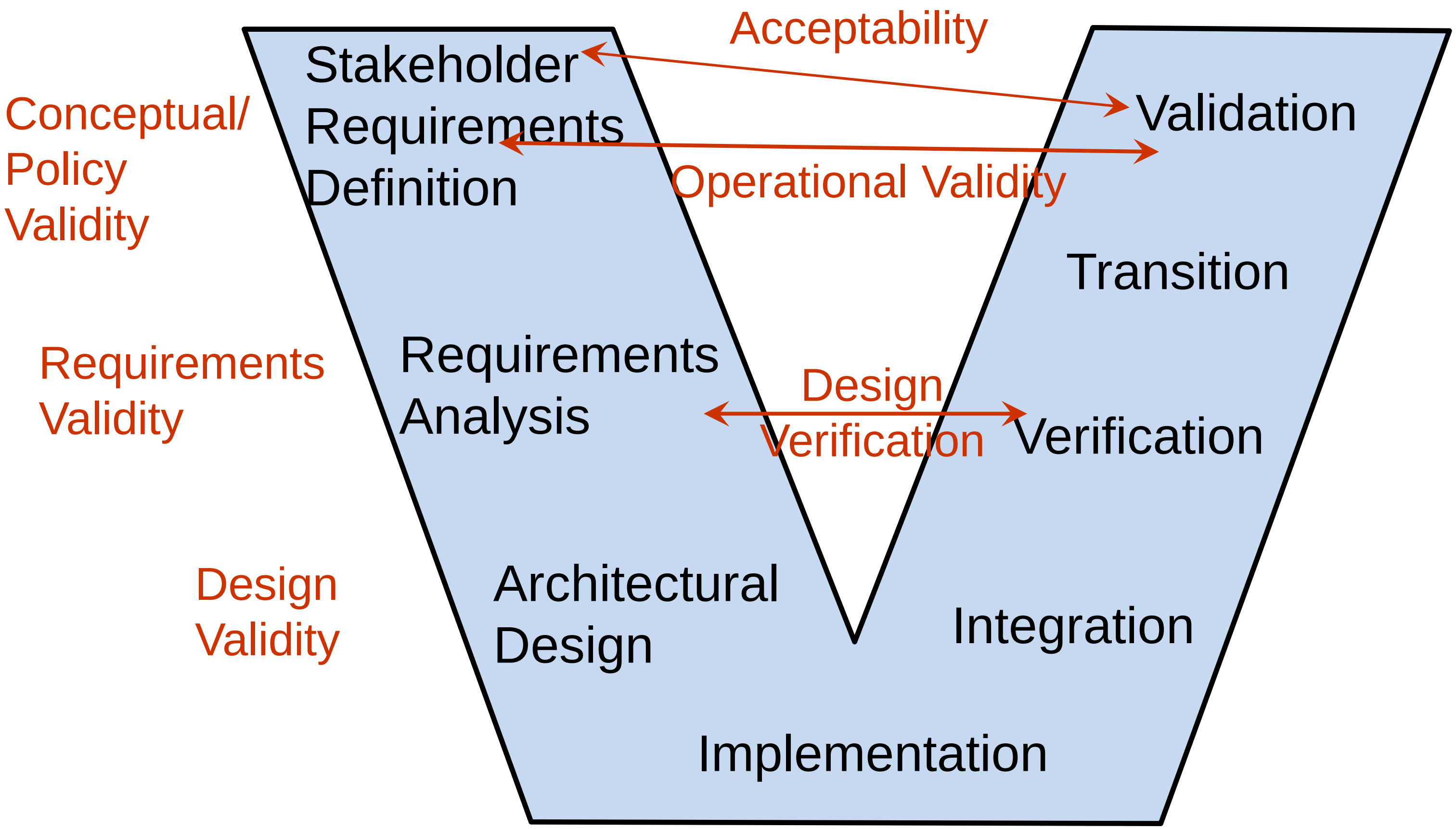
Petaflops



Baseline: China's Tianhe-2 computer rated at 33.86 petaflops
Assumption: Moore's Law holds up for the next 9 years

Extreme Model Checking Algorithmic Improvements: from 10^4 to 10^{15}

Verification and Validation in the Context of the *Vee* Model



Can break up development into multiple Vees

Buede

Early Validation 4-Step Process

1. Develop an operational concept
   a) Development of operational concept serves purpose of obtaining consensus in the written language of the stakeholders about the ways in which the system will be used
   b) Operational concept defines the vision for what the system is, the mission requirements (or the measures of effectiveness that the system must satisfy), and how the system will be used with other elements in the system's environment
2. Develop an objectives hierarchy
   a) Represents the value-drivers of stakeholder satisfaction for the system
   b) Stakeholders should be willing to pay to obtain improved performance (or decreased cost) in any one of these objectives
3. Develop an external systems diagram
   a) External systems of a system are those systems that are impacted by the system
   b) External systems are responsible for many of the system's requirements
4. The Continuous Early Validation (CEaVa) filtering module has four components
   a) Conceptual validity
   b) Requirements validity
   c) Design validity
   d) Policy validity

Note: Early validation needs rich traceability, both downwards and upwards.

Larsen and Buede 2002

Characteristics of Good Requirements

Attributes of Individual Requirement

1. **Unambiguous** – every requirement has only one interpretation
2. **Understandable** – the interpretation of each requirement is clear
3. **Valid** – make sure the specified requirement is correct
4. **Concise** – no unnecessary information is included in the requirement
5. **Traceable** – each requirement traces to an authoritative source (document or statement of the stakeholders)
6. **Design independent** – specify required functionality and performance rather than a particular solution or a portion of a particular solution
7. **Feasible** – the requirement can be implemented within the project's constraints (technical, cost, schedule)
8. **Verifiable** – a finite, cost-effective process has been defined to check the requirement
9. **Singular** – each requirement specifies only one function and only one performance parameter

Attributes of the Set of Requirements

1. **Unique** – requirements are not overlapping and/or redundant with other requirements
2. **Complete** – (a) everything the system is required to do throughout the system's life cycle is included, (b) responses to all possible (realizable) inputs throughout the system's life cycle are defined, (c) the document is defined clearly and self-contained, (d) there are neither to be defined (TBD) nor to be revised (TBR) statements; completeness is a desired property but can not be proven at the time of requirements development
3. **Consistent** – (a) internal, no two subsets of requirements conflict, (b) external, no subset of requirements conflict with external documents from which the requirements are traced
4. **Comparable** – the relative priority of the requirements is included
5. **Modifiable** – changes to the requirements can be made easily, consistently (free of redundancy), completely and traceable
6. **Attainable/Feasible** – solutions exist within performance, cost and schedule constraints

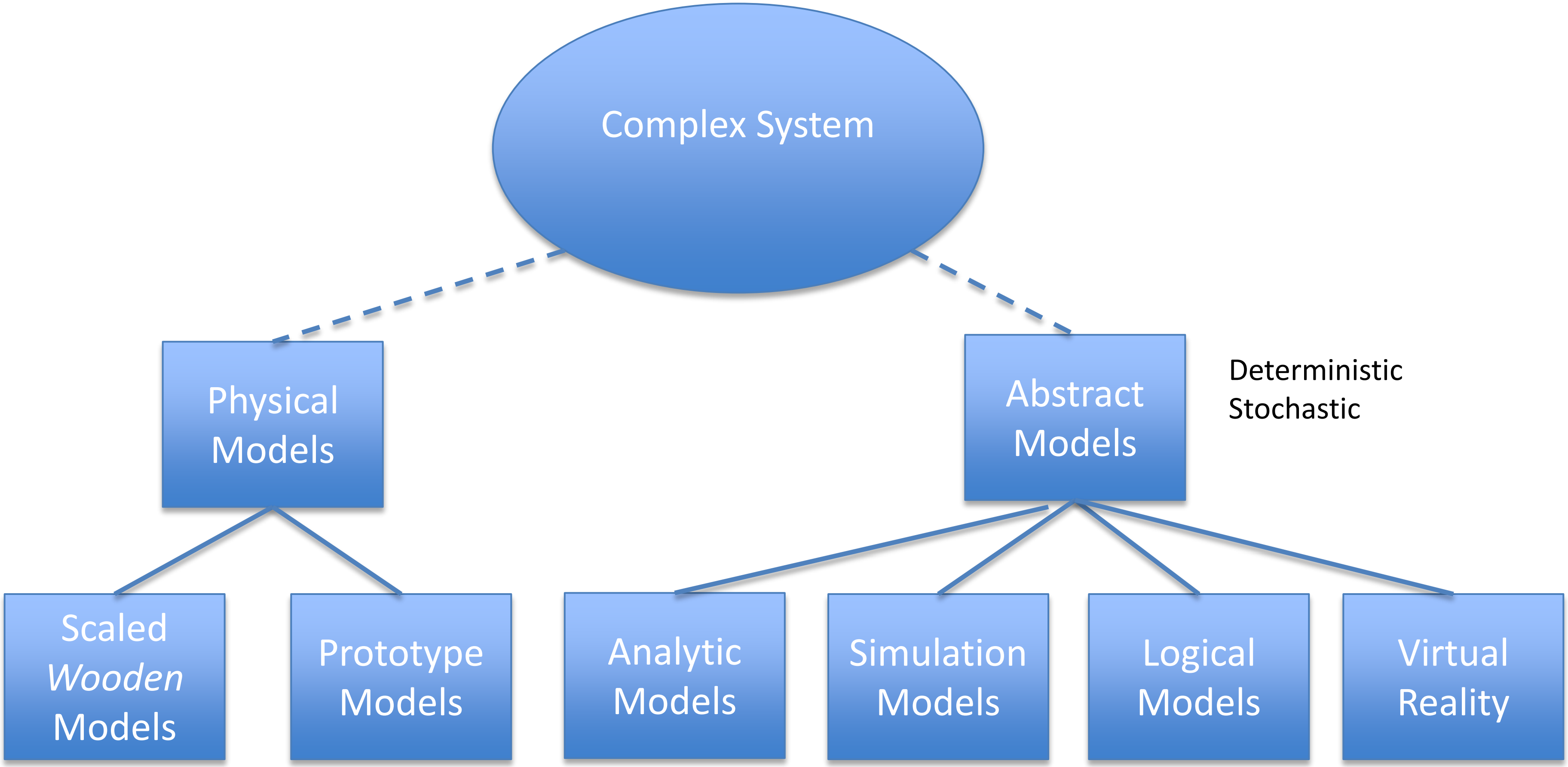
Adapted from Buede 2000 & 2009

Engineers responsible for system integration, verification, transition, and validation should sign off on the unambiguity, understandability and verifiability of the requirements and be members of the technical review panels

Complete Requirements Include Identifying Unintended Inputs and Undesired Outputs

	Inputs		**Outputs**	
	Intended	**Unintended**	**Desired**	**Undesired**
Signal	Pulse shape, data rate, signal to noise ratio	Electrical noise	Data rate, accuracy	Error rate, false alarm rate
Electrical	Nominal voltage	Surge voltages and timing	Voltage, current, frequency stability	Electromagnetic interference, electric shock
Mechanical	Activation force	Shock and vibration	Movement, resistance	Acoustic noise levels
Environmental	Normal temperature range	Temperature and humidity extremes	Particle density, air flow	Heat, effluents

Feasibility Assessment Critically Dependent on Modeling



Deterministic
Stochastic

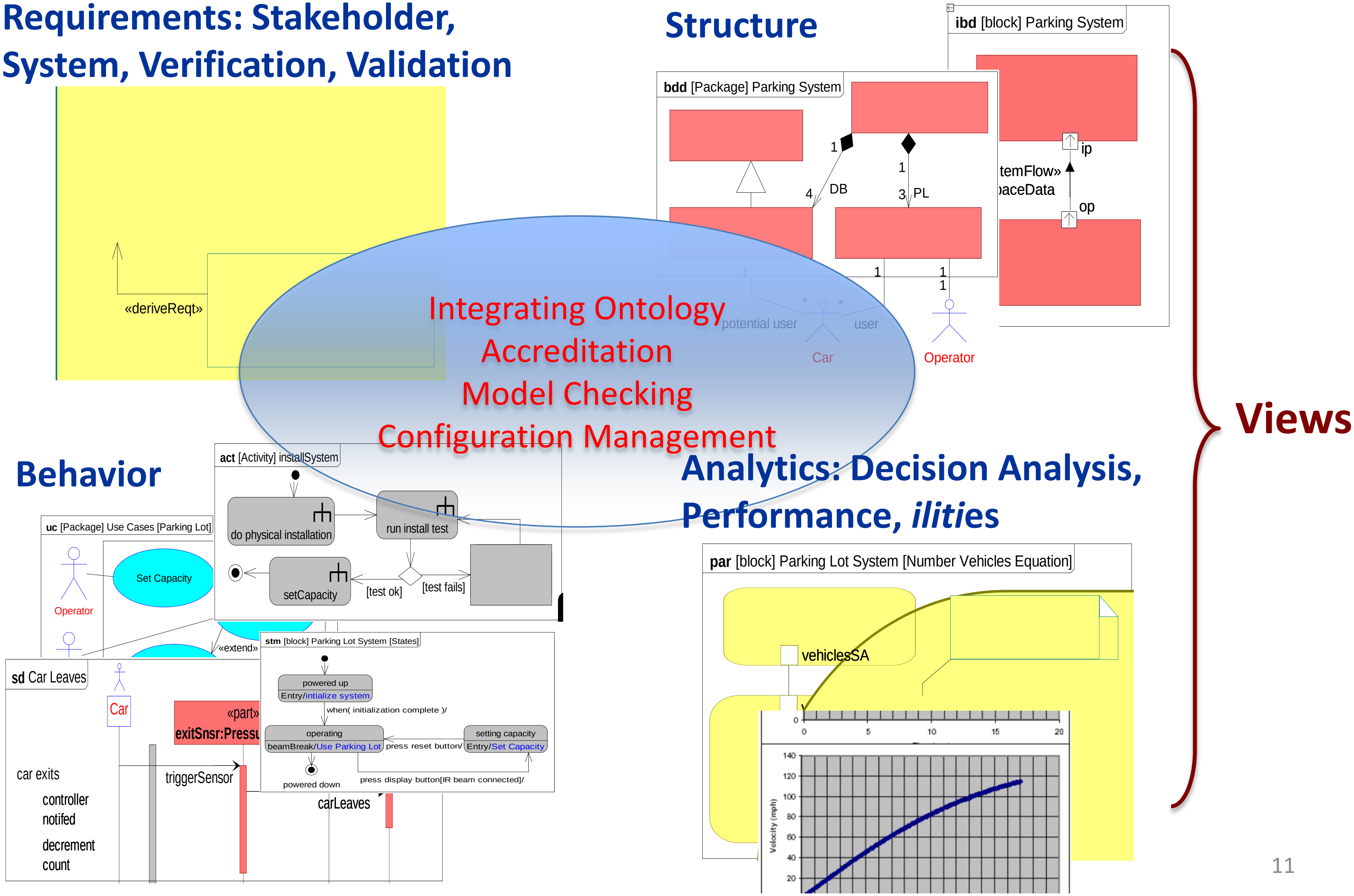
e.g., Systems of
Equations

Event Driven
System Dynamics
Monte Carlo

Requirements
Structure
Behavior
Parametrics

Human centric

Systems Modeling



Requirements: Stakeholder, System, Verification, Validation

Structure

Behavior

Analytics: Decision Analysis, Performance, *ilities*

Integrating Ontology
Accreditation
Model Checking
Configuration Management

Views

Leveraging Model-Based Systems Engineering for Model-Based V&V

Main Concept of MBSE

Replacing document-based systems engineering with an integral set of models, including

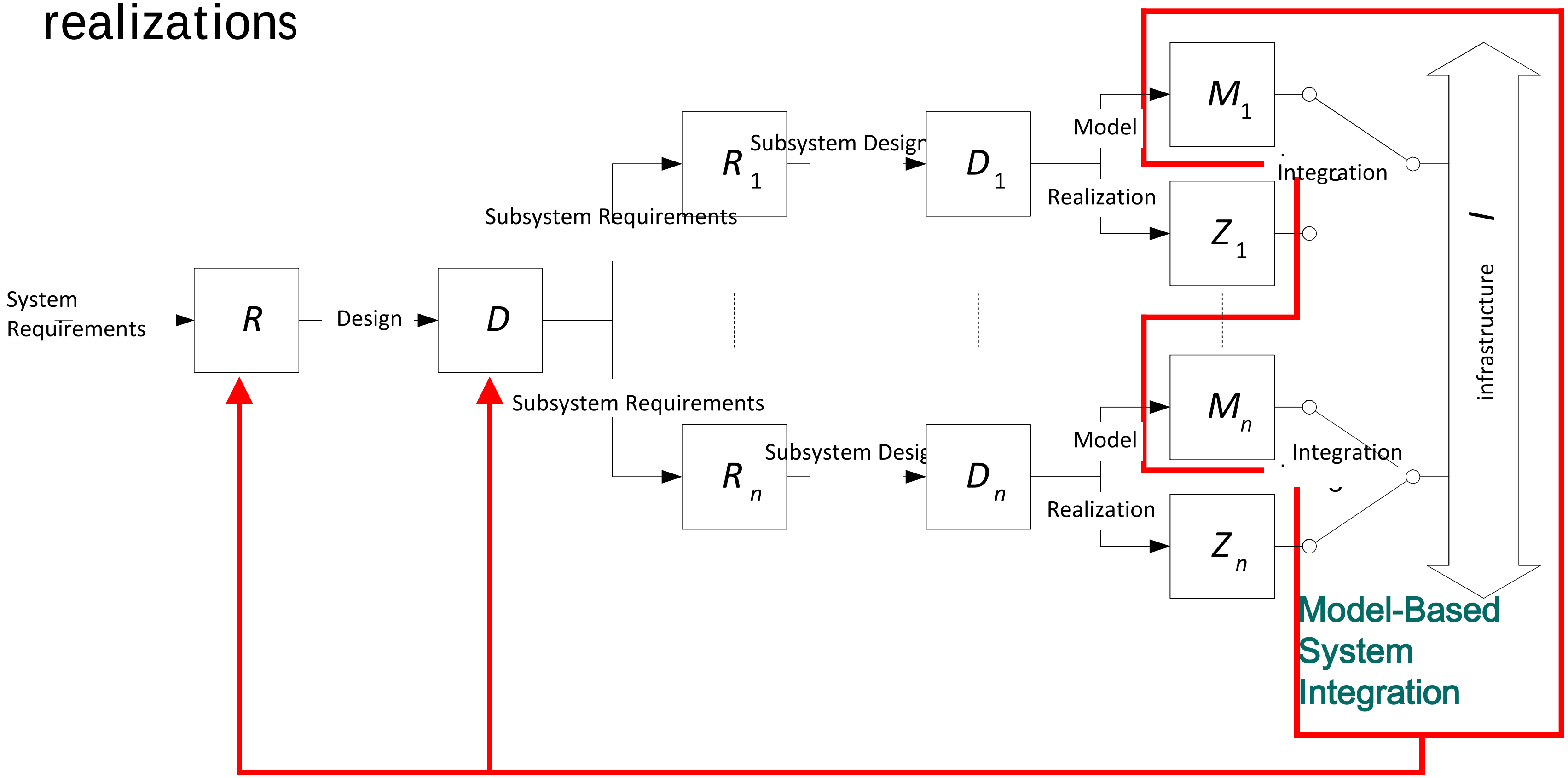
- Text-based models [entity-relationship diagrams]
- Graphical models for architecture definition, both structural and behavior (control flow, data flow, state transition)
- Mathematical analysis and simulation models for performance-cost-schedule, decision support and risk
- Error-checking models
- Visualization models

Levels of MBSE

- Partial replacement has existed since the late 1970s with automated support for N2 diagrams, block diagrams, function flow block diagrams
- Current goal is to have sufficient integration of the models to produce the current set of documents
  - Requirements documents
  - Architecture documents
  - Interface control documents
- Future goal is to conduct business via a completely integrated set of models and no documents

Model-Based Integration and Test

Requirements R, designs D, models M, realizations Z of a system with n components and infrastructure that allows integration of models and realizations



Model-Based System Testing

ESI

Model-Based Integration and Test

Integration Sequence Modeling and Optimization

Integration Modeling

- Tests depend on modules that are integrated
- Thus test sequence depends on integration sequence
- By optimizing integration sequence, more tests can be performed in parallel
- Result is reduction in integration & test time
- Risks are uncertainty of availability of modules and possible fault states

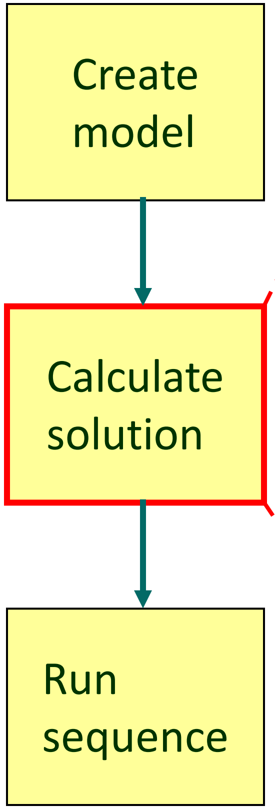
Integration Optimization

- Assembly by disassembly approach using AND/OR graph
- NP hard problem with AND/OR graph search size $\sim 2^{|M|} - I$, e.g., 20 modules need on the order of 10^6 investigations
- “Early time” & “parallel time” heuristics reduce computation time (cost for test and integration spent as early as possible in integration sequence to make sequence as parallel as possible)
- Automatic recalculation of sequence as risks encountered

Optimization moves away from pure bottom up, top down, etc. approaches

Model-Based Integration and Test

Test Sequence Modeling and Optimization

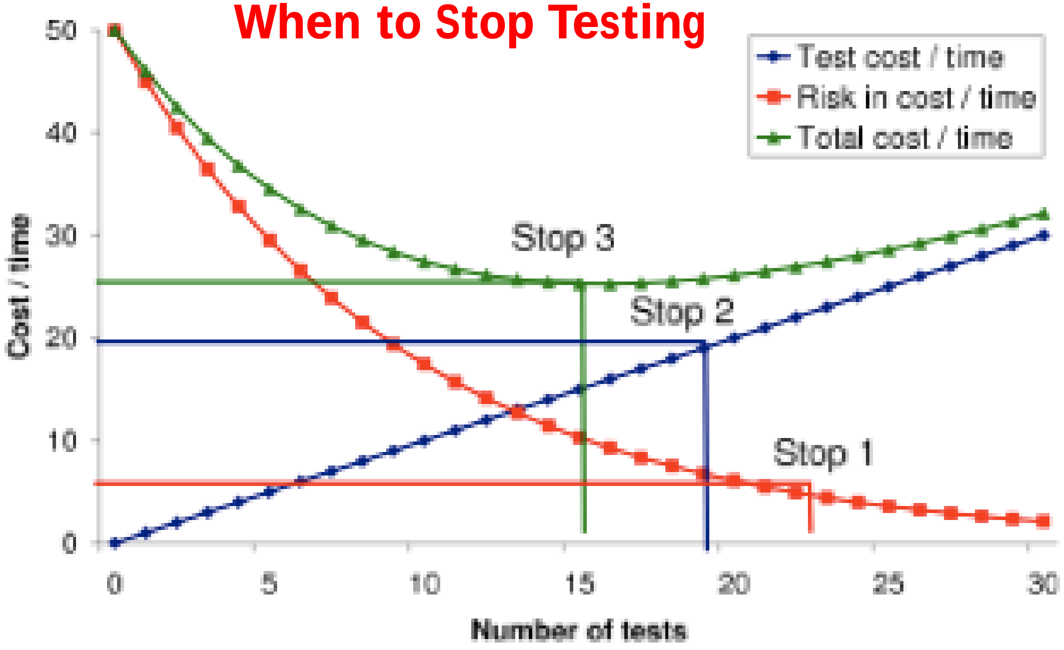


Create model

Calculate solution

Run sequence

When to Stop Testing



1 Stop when a certain system quality defined in risk is reached where risk is defined as the sum of the risk per fault state, which is defined as the impact of the fault state times the probability that the fault state is present, or

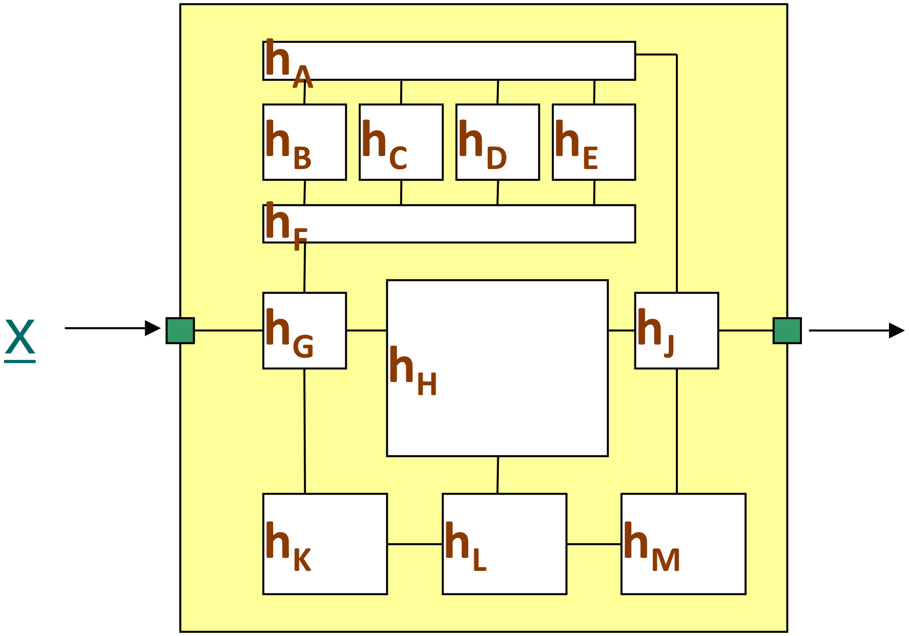
2 Stop when time or money runs out, or

3 Stop when the (weighted) sum of the time “cost” or money “cost” and the remaining risk defined as a “cost” is the lowest.

ESI Tangram Book

Model-Based Integration and Test

Model-Based Diagnosis



$\underline{y} = \underline{f}(\underline{x},\underline{h})$

Note:

If $\underline{y} = \underline{f}(\underline{x},\underline{h})$ then $\underline{h} = \underline{f}^{-1}(\underline{x},\underline{y})$

$h_i = 1$ means f_i is healthy,

$h_i = 0$ means f_i is at fault

Process:

1. map f to propositional logic
2. Identify $\underline{x}$ and $\underline{y}$ such that $\underline{y} \neq \underline{f}(\underline{x},\underline{1})$
3. Perform mathematical routines to compute the inverse function
4. The failing module or modules are indicated by $\underline{h} = \underline{f}^{-1}(\underline{x},\underline{y})$

Result is probabilistic estimation of module health

ESI Tangram Book

Model-Based V&V Considerations

- Challenge
  - ~ 10^n to 10^{n+m} increase in systems complexity over the next k years
- Techniques
  - Automated Testing
  - Test Optimization
  - Extreme Model Checking
  - Formal Methods
  - Statistical Verification
- V&V Technology Projections
  - Algorithmic Advances
  - Moore's Law
  - Automata (Parallel) Processors
- State of Practice
  - U.S. National Nuclear Security Administration (NNSA)
  - Semiconductor Industry
  - JPL Europa Mission (in development)

Some Successes

- Manhattan Project (1940s) from Richard W. Hamming, *The Art of Doing Science and Engineering*
  - Design options modeled and simulated on IBM accounting machines until a design was chosen to test
  - Last minute assessment of probability that the first live test would ignite the atmosphere
- Boeing 777 from Karl Sabbagh, *21st Century Jet*
  - Computer-graphics Aided Three-dimensional Interactive Application (CATIA)
  - Electronic Preassembly in the Computer (EPIC) replaced mock-ups
  - Flight control system models
- Semiconductors
  - Formal methods to verify designs driven by Intel's Pentium chip design defect
- Lithographic Machines from Jan Tretmans, editor, Embedded Systems Institute, *Tangram: Model-based integration and testing of complex high-tech systems*
  - Reduction in testing interval for next gen type systems driven by Moore's Law

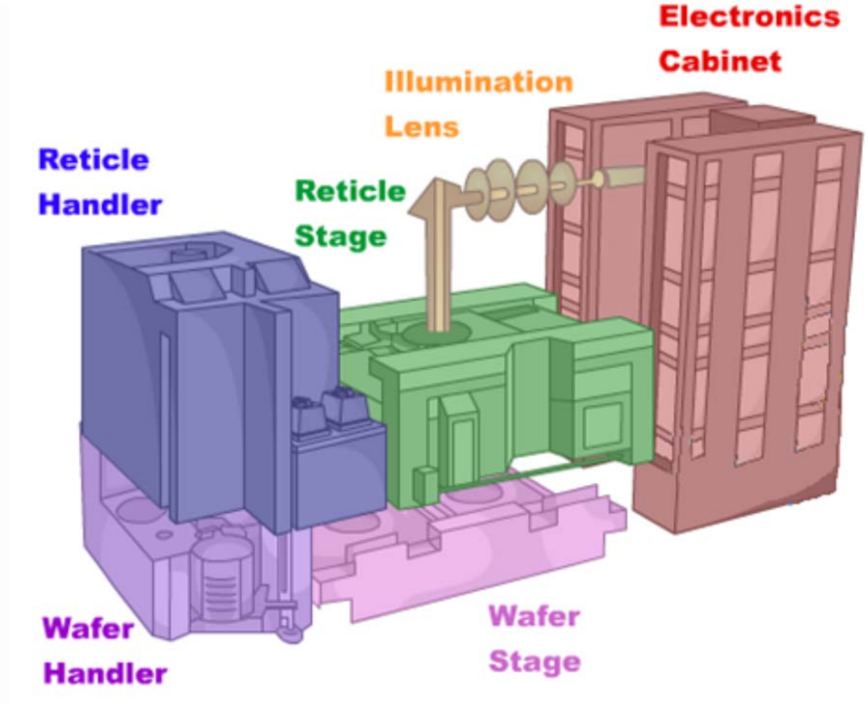
Reduction to Practice
ESI Tangram Project

MBI&T of lithographic systems

- High energy sources
- Mechanical motion
- Embedded electronics
- Embedded software
- Supplier-integrator model
- Technology cycle driven by Moore's Law

10% reduction in test time compared to manually generated plans

Model-based diagnosis time reduction from days to seconds/ milliseconds



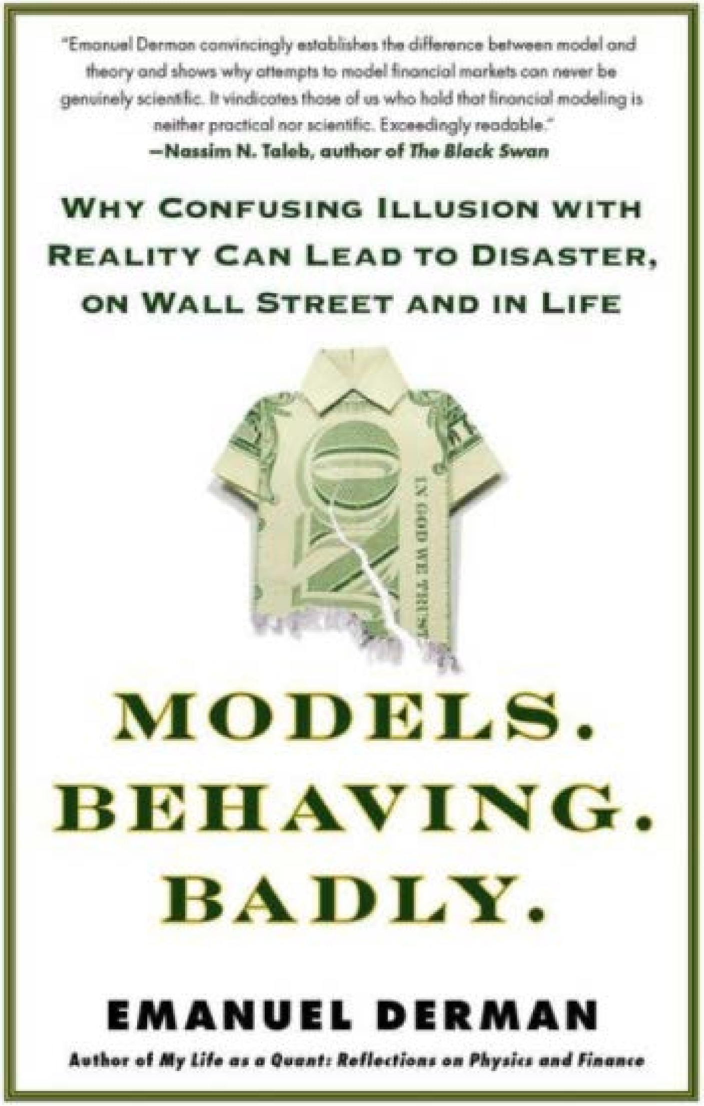
Embedded Systems Institute Tangram

Challenges of Modeling

George Box, published in proceedings of a 1978 statistics workshop:

- Now it would be very remarkable if any system existing in the real world could be *exactly* represented by any simple model. However, cunningly chosen parsimonious models often do provide remarkably useful approximations. For example, the law PV = RT relating pressure P, volume V and temperature T of an "ideal" gas via a constant R is not exactly true for any real gas, but it frequently provides a useful approximation and furthermore its structure is informative since it springs from a physical view of the behavior of gas molecules.
- For such a model there is no need to ask the question "Is the model true?". If "truth" is to be the "whole truth" the answer must be "No". The only question of interest is "Is the model illuminating and useful?".

https://en.wikipedia.org/wiki/All_models_are_wrong



"Emanuel Derman convincingly establishes the difference between model and theory and shows why attempts to model financial markets can never be genuinely scientific. It vindicates those of us who hold that financial modeling is neither practical nor scientific. Exceedingly readable."
—Nassim N. Taleb, author of *The Black Swan*

WHY CONFUSING ILLUSION WITH REALITY CAN LEAD TO DISASTER, ON WALL STREET AND IN LIFE

MODELS. BEHAVING. BADLY.

EMANUEL DERMAN

Author of *My Life as a Quant: Reflections on Physics and Finance*

Beware of emergent behaviors in socio-cyber-physical systems!

Back-ups

Conceptual Validity

- Purpose: establish a consistency check between
  - Stakeholders' Needs and
  - Operational Concept
- Operational Concept for each phase *k* of the system life cycle [Phase *k* is the particular phase of the development process, i.e., development, production, deployment, training, operation & maintenance, refinement, and retirement]
  - Vision Statement (high priority features)
  - Mission Requirements (Measures of Effectiveness)
  - Scenario Development (use case, sequence diagrams aka interaction diagrams, and dependency diagrams)
    - Identify other systems of interaction
    - Define inputs & outputs (black box)
    - Leads to External Systems Diagram
- Preliminary Objectives Hierarchy
  - Defines key performance and resource parameters at system boundary
  - Defines value system of stakeholders on performance and resource parameters

Larsen and Buede 2002

Operating Scenarios for Concept Validity

- Initialization of the system
- Normal steady state operation in standard operating modes of the system for all possible contexts (environments) in which the system may be placed, e.g., extreme cold, ocean depths.
- Extremes of operation due to high and low peaks of the external systems in each standard operating mode in each context
- Standard maintenance modes of the system
- Standard resupply modes of the system
- Reaction to failure modes of other systems
- Failure modes due to internal problems, providing as much graceful degradation of the meta-system as possible
- Shutdown of the system
- Termination (phase out) of the system

Note: The scenarios need to span the phases of the system life cycle; a rule of thumb for a common, relatively simple system would be 25 to 50 scenarios.

Larsen and Buede 2002

Operational Concepts Completeness Sanity Check

- Lifecycle phases as well as states and modes of operation
  - Development, manufacturing, verification, shipping, storage, installation, training, operations, maintenance, upgrading, disposal
- Viewpoints of all stakeholders
  - Developers, manufacturers, verifiers, purchasers, handlers (packing, storing, shipping, disposal), trainers, users (training, operations, upgrades), logistics personnel, maintenance personnel
- Nominal operations and environments
  - Who will use the system, why, where, when, under what conditions, how?
- "Off-nominal" operations and environments
  - Hazards to users, hazards to others, hazards to the system, hazards to other systems if the system fails, potential misuses of system, extreme conditions
- Interfaces
  - Inputs expected, outputs expected, input does not occur, output does not occur, wrong input occurs, wrong output occurs, unintended input occurs, undesired output occurs

Hooks and Farry 2001

Requirements Validity (1 of 3)

- Purpose: establish conformity between
  - Operational Concept
  - Originating Requirements Document (written in stakeholder language)
- Match system level requirements to operational concept
  - Mission requirements
  - Inputs and outputs of system
  - Objectives hierarchy
- Fine tune objectives hierarchy
  - Initiates Cost as Independent Variable
  - Establishes priorities of system view
  - Define thresholds and constraints of performance and resources
    - Provides guidance for trade studies and risk management
    - Provides structure for making technology insertion decisions
    - Establish trades between development and operational cost

Note: The 4 categories of requirement types are 1) behavioral, including inputs, outputs, functions and interfaces, 2) technology and system-wide, 3) trade-off, and 4) qualification. Note that very few requirements documents contain performance, cost and cost-performance tradeoff requirements.

Larsen and Buede 2002

Requirements Validity (2 of 3)

- Every originating input and output performance requirement of phase *k* must be traced to one and only one mission requirement of phase *k*. [Phase *k* is the particular phase of the development process, i.e., development, production, deployment, training, operation & maintenance, refinement, and retirement]
- Every mission requirement for phase *k* must be traced to at least one originating input and output performance requirement of phase *k*.
- For every external input (output) item in one of the scenarios of the operational concept for the system of concern, there should be at least one input (output) requirement in the Originating Requirements Document (ORD).
- For every input (output) requirement in the ORD, there must be an associated external input (output) item in one of the scenarios of the operational concept.
- There should be as many system-wide requirements as needed.
- There should always be at least one (and preferably only one) cost and one schedule requirement for the system of concern.

Larsen and Buede 2002

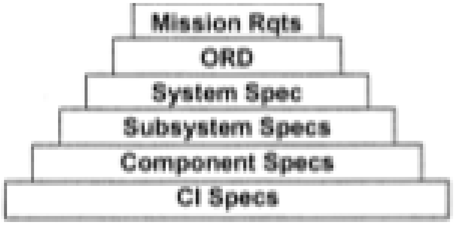
Requirements Validity (3 of 3)

- Requirements adhere to rules for syntax and structure
- Requirements comply with characteristics of good requirements
  - Individual Requirements: unambiguous, understandable, correct, concise, traced, traceable, design independent, and verifiable
  - Requirements Set: unique, complete, consistent, comparable, modifiable, attainable, and organized

Buede 2000 and 2009

Design Validity

- Purpose: establish congruence between
  - Originating requirements (ORD) in stakeholders' language
  - Derived requirements in engineers' language
- Derive requirements hierarchy (similar to Egyptian pyramids), e.g.,
  - System
  - Subsystems
  - Components
  - Configuration Items (CIs)



- Ensure consistency between derived and originating requirements
  - No new requirements
  - No forgotten requirements
  - No meaning changes

Larsen and Buede 2002

Policy Validity

- Purpose: analyze potential problem solutions and policies relating to product implementation
- Addresses policy consistency to identify and close gaps between policies and procedures
  - Scope (roles and obligations of stakeholders)
  - Domain
  - Procedures of organization
- Incorporates trade-off processes to 1) identify cost drivers and conduct cost-performance trade-offs and 2) establish cost-performance targets
  - Ensure timely cost versus performance trades
  - Sets realistic cost and performance thresholds and objectives ("best bang for the buck")
  - Scrub design for high cost – low performance features
  - Measure progress of achieving cost and performance goals

Larsen and Buede 2002