Modeling – The Value Proposition

Jim Streed
Motorola ASTRO® System Design

mod' el *n.* A schematic description of a system, theory, or phenomenon that accounts for its known or inferred properties and may be used for further study of its characteristics

American Heritage Dictionary, 4th Ed.

Modeling: Return on Investment?

Justification of a process change is elusive*:

- There are no "hard numbers"
- There will be no hard numbers in the foreseeable future
- If there were hard numbers, there wouldn't be a way to apply them to our situation
- If we did use such numbers, no one would believe us anyway

What is the value proposition of any modeling activity?

Given a project situation, how do we decide if modeling is worthwhile?

* Sheard, Sarah A. and Miller, Christopher L., "The Shangri-La of ROI", Proceedings of INCOSE, 2000

I was tasked with demonstrating the return on investment for UML modeling. Given my experience with process change, and Sarah Sheard's concise summary of the challenge of process change management, I realized that computing the ROI of UML was not particularly useful. At worst, it was a fool's errand.

I changed the problem statement from "What value does UML have?" to "How do I decide when modeling has value?"

Executive Summary

The value proposition in modeling is comparing the value of the answer to the cost to find the answer.

- The primary driver for modeling is risk reduction.
  - The value of the answer is often the value of reducing the risk
- Modeling benefits contributing to risk reduction:
  - complexity kept under control
  - improved requirements quality
  - rigorous design verification
  - increased formal review effectiveness
- There is **no global answer to the need for modeling**; each development situation requires a modeling choice.

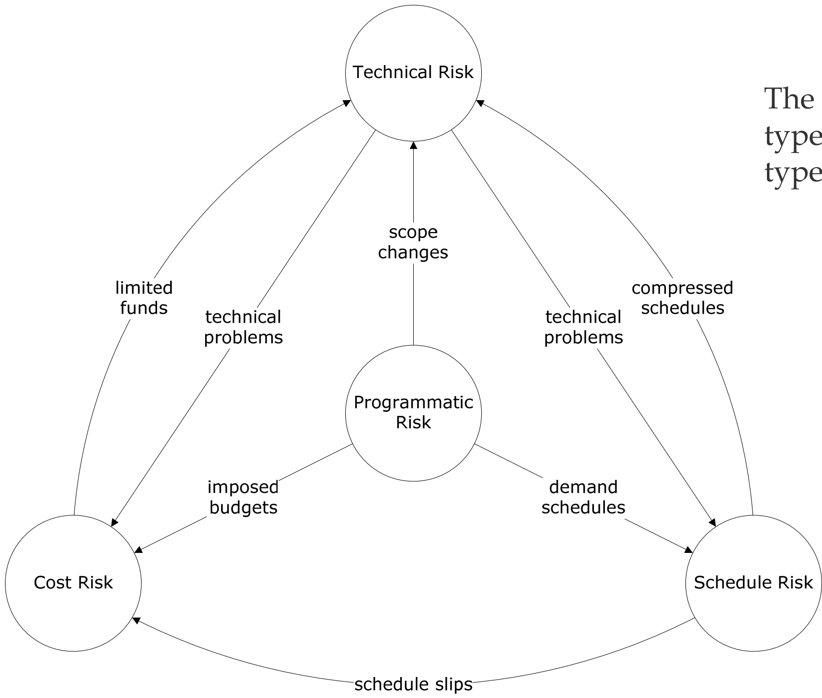
Modeling is a risk reduction activity. The value of modeling is tied to the value of reducing the risk. The value of reducing risk is dependent on situation and an organization's risk tolerance level.

The rest of this presentation is a tour of risk and modeling. Modeling is a very tailor-able activity that can be shaped to any combination of risk and time horizon.

Agenda

- Risk
- The Nature of Models
- Motivation
- Modeling in Practice

The Risk Environment



(redrawn from the INCOSE Systems Engineering Handbook)

The presence or mitigation of one type of risk can give rise to another type of risk.

A program is subjected to a number of types of risk.

The appearance or mitigation of one type of risk can give rise to risks in other areas. For example, the activity of mitigating a technical risk (a requirement might not be met) can give rise to schedule risk.

Risk Quantification

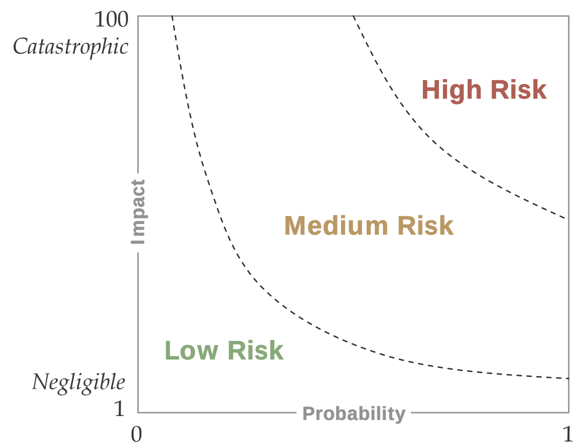
Risk Statement Form

If <event occurs>,
then <consequence to customer, business, or program>.

The Risk Priority Number (RPN)

is the product of

the **probability** of the event occurring
and
the **impact** to the customer, business, or program.



Risk is the combination of probability and impact.

An event without consequence does not constitute a risk.

An event whose probability is 1 is not a risk, it's an issue.

A statement of pure probability is an incomplete characterization of the risk.

Finding Risks

Structured Risk Assessments

- Design Failure Modes Effects Analysis (DFMEA)
- Fault Tree Analysis (FTA)
- Performance Risk Assessment

Examine the successes, failures, problems, and solutions of similar prior projects

- Post mortem lessons learned
- Field failure data
- Risk database – patterns of success and failure

Ask experts and those with experience

- What scares you? What keeps you awake at night?
- What assumptions have you made?

Assumptions are very important indicator of risk. If the assumption doesn't come true, then there is a consequence. How safe is the assumption, and how significant is the consequence?

Agenda

- Risk
- The Nature of Models
- Motivation
- Modeling in Practice

"The genius of human understanding is to choose what to study and what to set aside."

*- Guy Consolmagno, S.J.**

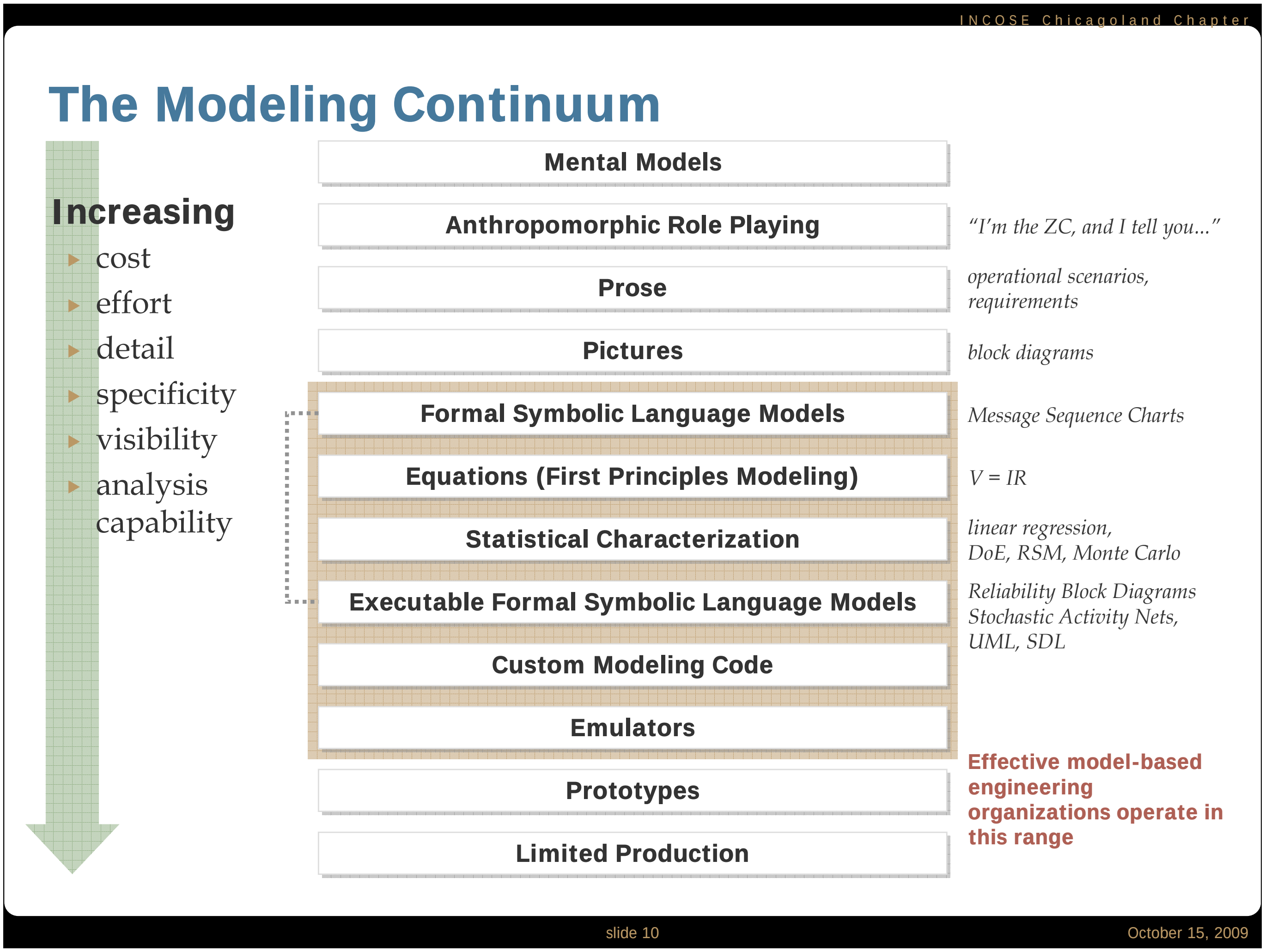
A model "concentrates on describing a selected aspect of the world, setting aside other aspects as incidental."

- John H. Holland†

* Consolmagno, Guy, *God's Mechanics: How Scientists and Engineers Make Sense of Religion*, 2008

† Holland, John H., *Emergence: From Chaos to Order*, 1998

Modeling is not the real thing. Modeling is the representation of a real (past, present, or future) thing.



The modeling types represented on this slide are all valid and have useful applications.

Moving on the continuum from top to bottom,

- the models cost more to create
- the models require more effort to create
- the models contain more detail (more decomposition, *e.g.*, more moving parts)
- the models become more specific (less abstraction, *e.g.*, types of parts)
- the models become easier to comprehend for outsiders
- the models have higher fidelity in their analysis – they are closer to reality

The Yin Yang of Mental Models

Mental Models are

- Unreviewable
- Unverifiable
- Unique for each engineer
  - difficult to identify conflicts
  - source of debate in many technical meetings

Mental Models are

- how an engineer understands a problem
- an essential first step in creating any explicit model

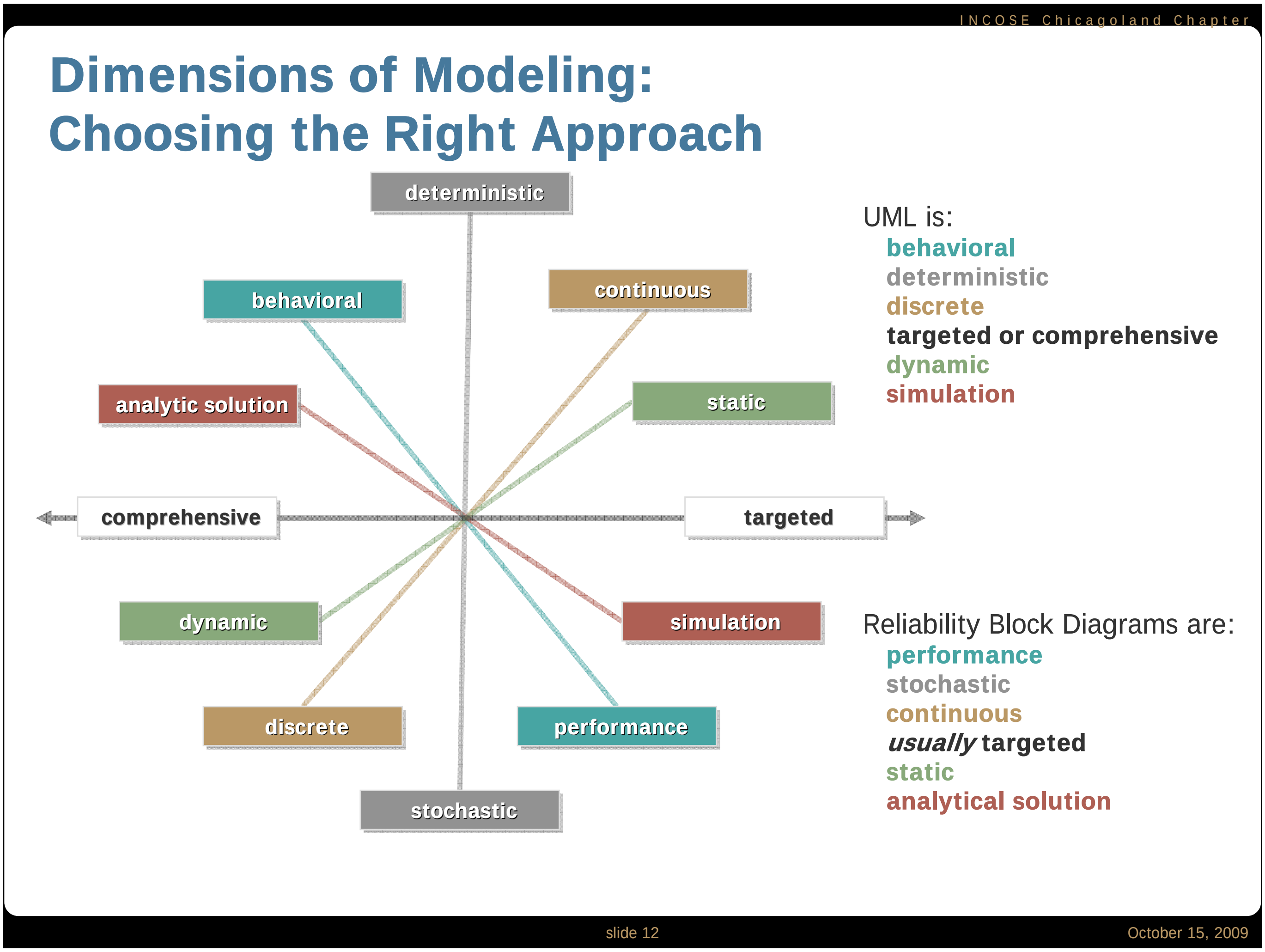
Reverse engineering a mental model

When requirements are written from a mental model, requirements reviewers must create their own mental model to assess the completeness and accuracy of the requirements.

Do the mental models match?

Mental models are absolutely necessary.

It more dangerous to rely on mental models the larger the development organization becomes.



Models come in many forms. Each modeling approach has a unique set of characteristics that make it optimal to solve a particular problem.

Comprehensive models cover most or all of the problem space; targeted models cover a judiciously-chosen segment of the problem space.

Analytic solutions are closed form (solving equations); simulations provide a result through statistical sampling.

Behavioral models focus on functional interaction; performance models focus on quantifying the quality of the behavior. (All performance models start with a behavior model.)

Deterministic models do not account for any random variation; stochastic models include random variation.

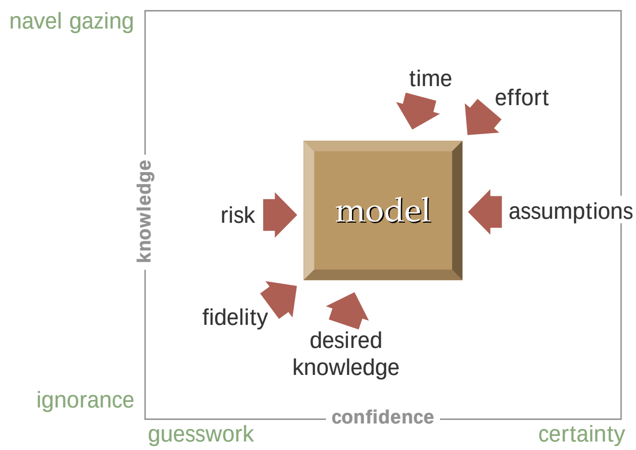
Continuous models accept continuous inputs and/or produce continuous outputs; discrete models are typically event-driven.

Static models are generally not time-dependent or do not include transient effects; dynamic models show the changeable nature of the system modeled.

Modeling Forces and Productive Tension

"All models are wrong, some models are useful."

- George Box



Models are never finished. A modeler finds the right balance of forces to determine when the model is "done enough".

Conflicts over modeling are often rooted in a disagreement on how these forces are to be balanced. Typically, the non-modeler under-estimates the risk, and the modeler over-estimates the fidelity necessary.

Agenda

- Risk
- The Nature of Models
- Motivation
- Modeling in Practice

“Models, above all, make anticipation and prediction possible.”

*- John H. Holland**

* Holland, John H., *Emergence: From Chaos to Order*, 1998

Why Model?

Reduce Risk
any combination of technical, cost, or schedule risk

- Mitigate risks before implementation and verification phases
- Break apart one nebulous risk into many smaller specific risks

Manage Complexity

- Comprehend the inherent complexity in the system
- Drive out unnecessary complexity (over engineering, gold plating)
- Reduce effort to review, implement, and verify

Make Decisions

- Converge to a decision
- Make the right decision
- Make the decision stick

Investigate

- Provide insight into a problem area
- Get quick, approximate answers that can be used immediately

Understand

- Give customers useful advice
  on how to get the most from their system

Modeling increases the probability of "doing it right the first time."

The last 4 reasons are all contributors to the risk level.

Modeling Enhances Engineering Capability

Requirements identification (requirements modeling)
- complete, non-conflicting requirement set

Quantification
- write measurable requirements

Requirements verification (for Verification Method of Analysis)
- Availability
- Capacity

Design verification
- rigorous design verification replaces expert blitz (of the implementation) in system test
- modeling is more effective than formal reviews at verifying the design

Critical Parameter Management
- quantify
- allocate
- predict

An engineering organization that models is more effective at other engineering tasks.

When to Model?

Factor	Considerations
Risk	How much risk is present, and how much are you willing to spend to mitigate that risk? What is the consequence if the wrong decision is made?
Financial Stake	Is this a $20 decision, or a $20,000,000 decision? What is your exposure to engineering rework?
Effort	Can you scale the modeling effort to match the risk or the financial stake? How much modeling is enough? How much modeling is too much? Can you get the fidelity needed given the time and effort available?
Complexity	*A complex system is one where no single person can comprehend and manage the entire system.* Can you analyze the problem in your head? Can you effectively communicate the reasoning behind your decisions?

Factors to consider when deciding whether modeling might have value.

The last 3 factors are all components of risk.

Complexity*

Detail Complexity

- Many combinations,
  many components,
  many variables,
  many steps
- Examples:
  - assembling a bicycle
  - call processing flows
  - system upgrade procedure
- Begets brittleness
  - A rigorous specification of the design is essential; it prevents important detail from being "pushed to the margins"

Dynamic Complexity

- Doing the obvious thing does not produce the obvious, desired outcome
  - the same action has dramatically different effects in the short term and the long term
  - an action has one set of consequences locally and a very different set of consequences elsewhere in the system
  - obvious interventions produce non-obvious consequences
- Begets surprises
  - Understanding dynamic complexity helps to reveal patterns and major interrelationships

* Senge, Peter M., *The Fifth Discipline*, 1994, *pp*. 71-72

Complexity comes in two flavors: detail complexity and dynamic complexity.

Modeling has a role in handling both types of complexity, but the modeling approach may be different for the two types of complexity.

A common mechanism for dynamic complexity is a component that correlates a number of inputs from a number of different sources to determine behavior. Changing one stimulus can produce unexpected behavior in the target component if the other inputs are in unanticipated states.

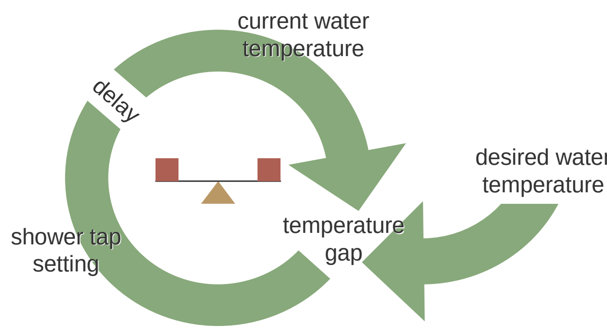
Dynamic Complexity

Indicators of dynamic complexity

- Circular relationships
  - positive or negative feedback
  - balancing interactions
- Delays in response
- External influences
- Adaptive agents

Examples

- An interface with state machines at both terminations
- Stock Market: existence of a model to **predict** market behavior **influences** market behavior
- Setting the shower for ideal temperature*

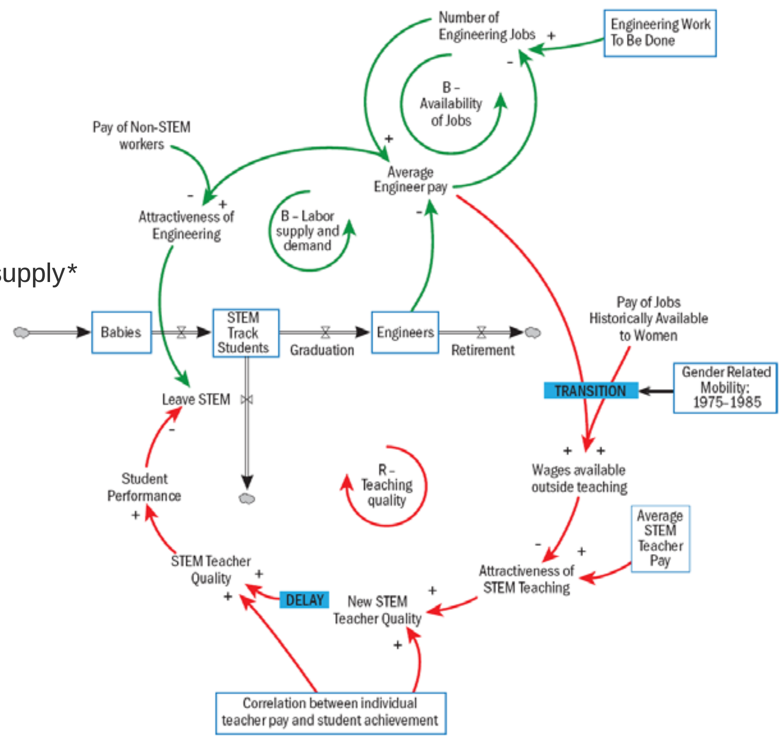


* Senge, Peter M., *The Fifth Discipline*, 1994

Dynamic complexity is particularly difficult to comprehend and represent. The discipline known as “systems thinking” helps to understand and characterize dynamic complexity.

A Dynamic Complexity Example

How the Science, Technology, Engineering, and Mathematics (STEM) teaching system affects the engineering labor supply*



* Sturtevant, D. J. 2008. *America disrupted: Dynamics of the technical capability crisis.* MS thesis, MIT. Available at http://www.clexchange.org/ftp/documents/Research/RS2008-09AmericaDisrupted.pdf.

This example is included for interested readers to study after the presentation.

Emergence: much coming from little

Dynamic complexity can produce emergent behavior

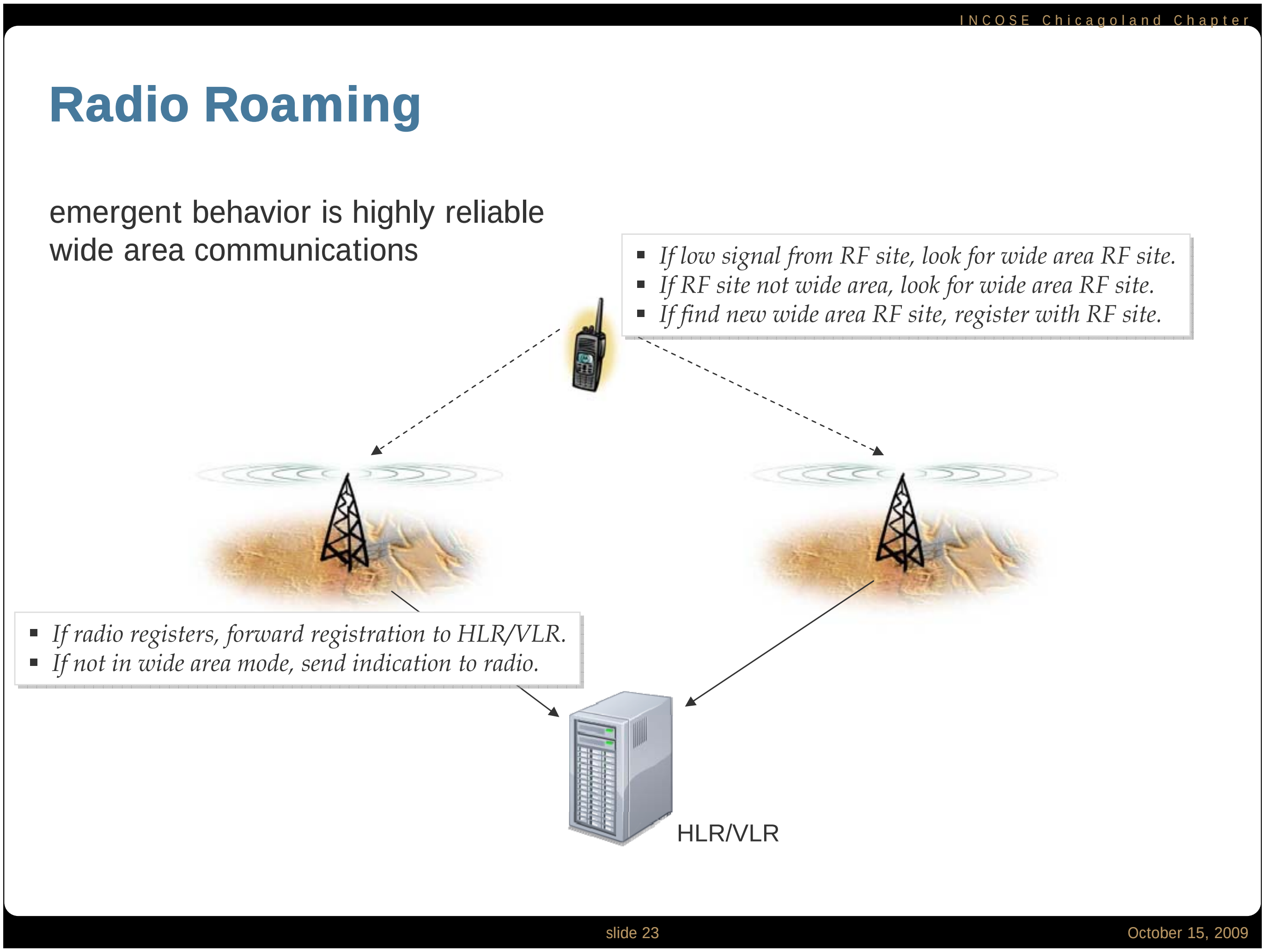
- Emergent behavior arises when building blocks **interact**, often following simple **rules** based on **local context**. The system behavior is not evident until the building blocks interact.
- Emergence enables a system to become greater than the sum of its parts.

Emergent behavior can be desired or undesirable

- Teamwork
- Music
- Games: chess, checkers
- Flock of birds, ant colony
- Gaper's block on a highway
- An economic system
- Radio roaming and mobility

Emergence is a topic that can be a 1-hour presentation on its own.

Often, the rules based on a local context are incentive-based. An economy is an excellent example of an incentive-based system with emergent behavior.



This is an example where agents act upon their local knowledge of the environment. There is no central control for radio roaming. High reliability wide area communications is the emergent behavior.

This design has desirable and undesirable emergent behavior.

So What?

Dynamic complexity is an indicator of risk

- scenario sensitivity
- undesirable emergent behavior
- race conditions and timing interdependencies

Modeling is often the only way to gain the deep understanding of the system necessary to mitigate this type of risk.

Without awareness of dynamic complexity and emergent behavior, engineers can underestimate the amount of risk in a particular design.

Agenda

Risk

The Nature of Models

Motivation

Modeling in Practice

Modeling as a Skill

What are the characteristics of a good modeler?*

Enjoys Word Problems figures out what's important, what's not important, and what's missing

Ability to Abstract makes good assumptions, validates them, knows where they apply, and *remembers* them

chooses an appropriate level of detail

Able Communicator collects information from, and provides information to, many individuals and groups with varying levels of expertise

documents the model and assumptions for future reuse

Ability to Draw Pictures represents the problem space in means other than words; can translate a mental model to paper

Healthy Skepticism questions everything to discern the real problem to be solved, the real requirements, the implicit assumptions, the validity of the data, …

* Moore, S., "System Design Performance Roadmap", *pp.* 11-32, revision 1.1, July 5, 2005

How do we teach engineers to think like a modeler?

Many important elements of model building are intuitive rather than formal.

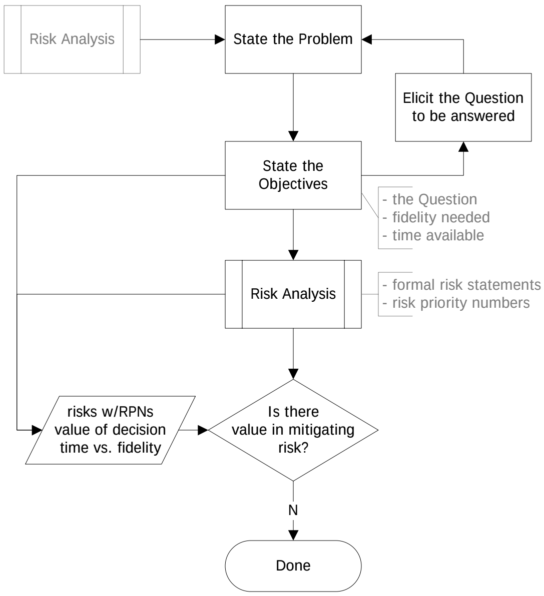
"Model building is all about the information you throw away." In real life… "you have far too much data and the real problem is in throwing it away, knowing what is and is not important."

Modeling as a Skill

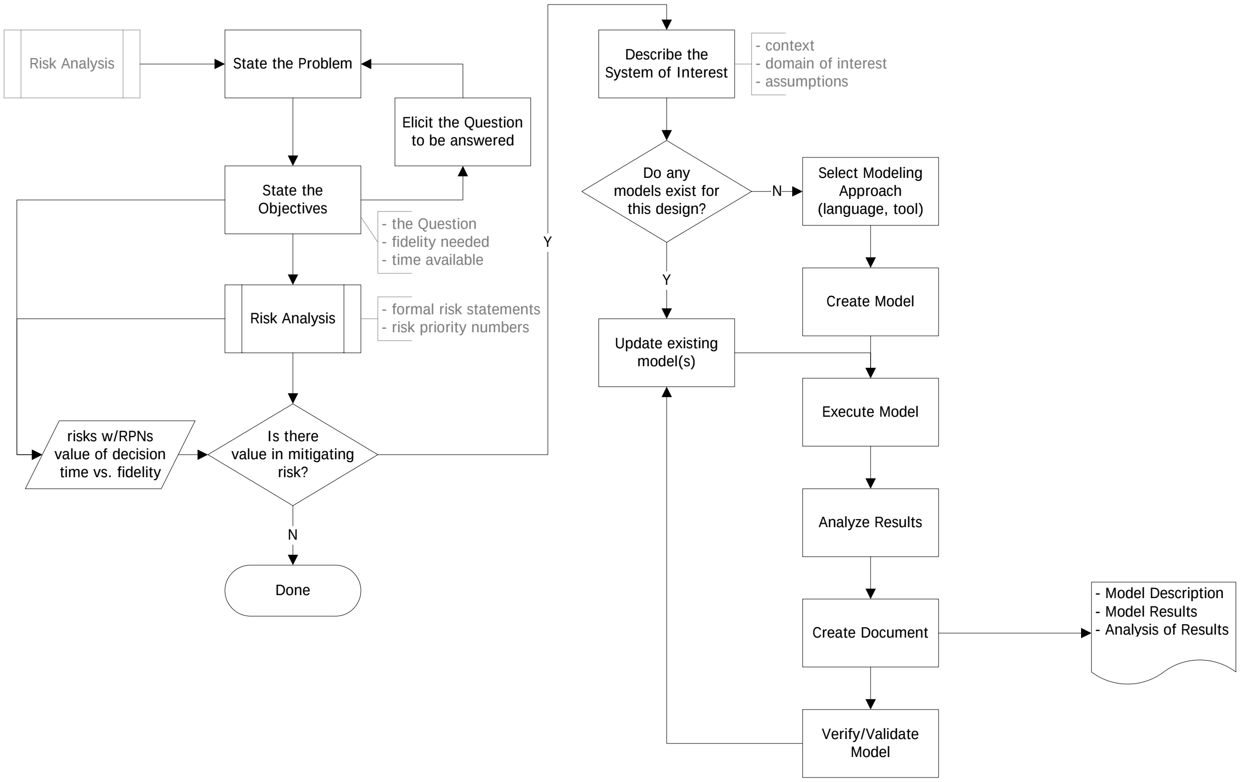
Building modeling skills is a mentoring / apprenticeship opportunity…

…but a process can serve as a starting framework.

The Modeling Process, Part 1



The Modeling Process, Parts 1 & 2

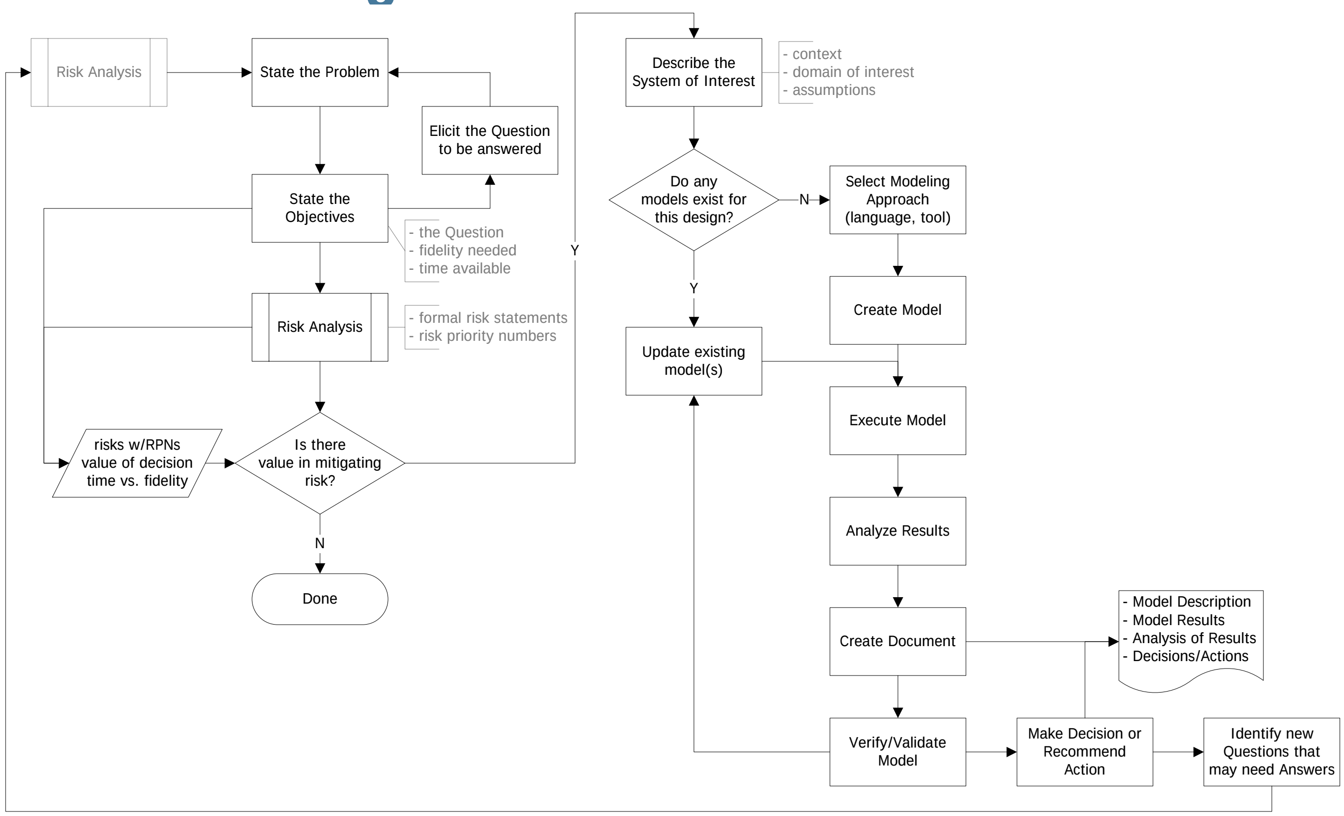


The distinction between model verification and model validation is purposely blurred in this flow chart to keep it simple (it was a modeling decision :-)).

Model verification is the act of making sure the model represents the system that was intended to be represented. (Did I model the thing right?)

Model validation is the act of making sure the model is reasonably representative of the realized system. (Did I model the right thing?) How safe is it to act on the results of the model?

The Modeling Process



Living in the Modeling Mindset

Is it worth it to model?

- How big and scary are your risks?
  - Do you have a well-defined project plan?
    Are you confident it will work?
- What is the financial stake in your decision?
- Can you arrive at an answer of acceptable fidelity in the time available?
- What expertise is available?
- What skill sets are available?

Risk Identification and Quantification is key to employing modeling wisely

- RPNs show what should be addressed first, or addressed at all

Living in the Modeling Mindset

Questions for program reviews:

- Where is your list of technical risks?
  - There are never *no* technical risks!
  - Make sure the risks are actually risks

    Can they be stated in formal "IF event, THEN consequence" format?
- What specific question(s) are you trying to answer with modeling?
- What might you change as a result of the modeling?
- What modeling are you using to mitigate your highest risks?
  - At a minimum, engineers always use mental models
- How will your chosen modeling approach mitigate the risks?
- How will the RPN change, and when will it change?

Questions?

Supplemental Information

Modeling and Deliverables

Modeling enhances the creation of mainline deliverables

- Provides concrete data for feasibility studies
  to enable early decisions
- Provides numerous requirements improvements
  - facilitates requirements identification (requirements modeling)
  - (probably) reduces number of requirements
    no need to repeat description of design if it's in the model
  - facilitates creation of measurable, verifiable requirements
  - enables verification of requirements with method of Analysis
    enables new requirements previously useless to specify
- Identifies test scenarios
  - can create test scripts from the model

Modeling and Deliverables

A Behavioral Model can serve as design documentation

- A comprehensive model acts as a blueprint for feature implementation
- Fewer artifacts are needed when the model is the design; the design documentation can't get out of sync with the model

Modeling increases the effectiveness of formal reviews

- Rigorous design verification with executable models
- Enables evaluation of completeness and correctness
- Reduces the need to reconstruct the design with a mental model when reviewing requirements
- Abstraction helps more people understand how it works

Modeling and Deliverables

Abstraction increases reusability

- identifies patterns that facilitate reuse

Modeling facilitates remote site development

- it breaks down language barriers and experience barriers

Modeling moves Expert Blitz from Test Phase to Design Phase

- A comprehensive behavioral model can be stressed *before* implementation

Modeling increases the reliability of project estimation data

- Reduces the variability of number of operational scenarios, requirements, message sequence charts

Modeling Strategy

Consider two possible approaches to implementing an organizational Model-Based Engineering strategy

- Unified System Model
  - one large behavioral model
  - new functionality integrated into existing model
  - performance models started from functional model
- Coordinated, Targeted Models
  - multiple models created for specific purposes
  - models created and stored according to rules set by system's functional architecture
  - no single model of system, but individual models can be integrated (sometimes with significant effort) to address a larger scope

Modeling Strategy Considerations

Unified System Model	Coordinated, Targeted Models
needs dedicated staff to maintain	need oversight to maintain model database schema
fewer, expert modelers work on the Model (used as department resources, or set up as domain experts, *e.g.*, call processing)	all engineers need some modeling proficiency; all engineers need skill in knowing when and how to apply modeling
can reveal non-obvious system interactions	will not identify interactions outside the model's scope
potentially rapid cycle time once Model is largely in place (step function)	moderate improvement in cycle time (doing it right the first time)
single modeling language (*e.g.*, UML), or model integration environment (*e.g.*, ML Designer)	potentially varied modeling approaches