SPRINGER NATURE
Reference

Azad M. Madni
Norman Augustine
Editors-in-Chief

Michael Sievers
Associate Editor

Handbook of Model-Based Systems Engineering

Foreword by
Charles Bolden Jr.

Springer

Overarching Process for Systems Engineering and Design

7

A. Terry Bahill and Azad M. Madni

Contents

Introduction to Modeling 197
Uncertainty Is Ubiquitous 197
Model-Based System Engineering 198
Purpose of Models 199
Kinds of Models 199
Types of Models 199
Tasks in the Modeling Process 200
Model for a Baseball-Bat Collision 200
Checklist for Tasks Necessary in a Modeling Project 202
Requirements Discovery Process 208
Where Do Requirements Come From? 209
A Use Case Template 209
A Use Case Example from a Chocolate Chip Cookie-Making System 211
A Test Plan for This System 213
Tradeoff Study Process 214
Tradeoff Study Example from a Chocolate Chip Cookie Acquisition System 216
Risk Analysis Process 220
Risk Analysis Example from a Chocolate Chip Cookie Making System 220
Comparing the Requirements, Tradeoff, and Risk Processes 222
Comparing the *Activities* of the Requirements, Tradeoff, and Risk Processes 222
Comparing the *Products* of the Requirements, Tradeoff, and Risk Processes 223

A. T. Bahill (✉)
Systems and Industrial Engineering, University of Arizona, Tucson, AZ, USA
e-mail: terry@sie.arizona.edu

A. M. Madni
Systems Architecting and Engineering, Astronautical Engineering Department, University of Southern California, Los Angeles, CA, USA
e-mail: azad.madni@usc.edu

A. M. Madni et al. (eds.), *Handbook of Model-Based Systems Engineering*,
https://doi.org/10.1007/978-3-030-93582-5_16

A Requirement from the Chocolate Chip Cookie Acquisition System 224
An Evaluation Criterion from the Chocolate Chip Cookie Acquisition System 226
A Risk from the Chocolate Chip Cookie Acquisition System 227
Comparing a Requirement, an Evaluation Criterion, and a Risk 228
The SIMILAR Process 229
State the Problem 230
Investigate Alternatives 232
Model the System 232
Integrate 233
Launch the System 233
Assess Performance 234
Reevaluate 234
The Overarching Process 235
Effects of Human Decision-Making on the Overarching Process 237
Confirmation Bias 237
Severity Amplifiers 239
Framing 240
The Overarching Process 242
Uncertainty in Stating the Problem for the Overarching Process 244
A System for Handling Uncertainty in Models and Documentation 244
Chapter Summary 249
Cross-References 253
References 253

Abstract

This chapter presents three key processes central to systems engineering: requirements discovery, tradeoff studies, and risk analysis. It compares and contrasts these three processes and then combines them into a single *Overarching Process*. The three original processes can then be viewed as specific tailorings of the Overarching (superset) Process. Similarly, the Overarching Process can be viewed as a top-level process (a superset) for model-based system engineering (MBSE) implementations. The Overarching Process itself is not an example of model-based systems engineering, except at a high level. This chapter also identifies the activities in the Overarching Process that contribute to uncertainty. All of these activities involve human decision-making. Therefore, most mistakes caused by uncertainty are found in the system models and documentation. These mistakes often arise from confirmation bias, severity amplifiers, and framing. The two key examples used in this chapter are the Cookie Acquisition System and the BaConLaws model for baseball-bat collisions.

Keywords

Uncertainty · Requirements · Tradeoff study · Trade study · Risk analysis · Baseball · Decision analysis and resolution · Sensitivity analyses · Multi-objective decision-making

Introduction to Modeling

A model is a simplified representation of some aspect of a real system. Models are ephemeral: They are created, they explain a phenomenon, they stimulate discussion, they foment alternatives, and then they are replaced by newer models. Engineers know how to construct a model, but quite frequently, they miss a few steps. This recognition provides the impetus for this chapter that presents a succinct description of the modeling process.

Requirements discovery, tradeoff studies, and risk analyses are three distinct systems engineering activities. Even though they have the same underlying process structure, they appear different because they employ different vocabularies, inputs, and outputs. To convey the similarity of these processes to the reader, we abstracted and grouped the common activities in these three processes. The incipient development of this approach was presented in *Tradeoff Decisions in System Design* [8]. In the interest of clarity, the processes, shown in our figures, suppress the explicit representation of temporal sequences. Also, in the interest of clarity, we suppress the multitude of feedback loops that arise when several of these activities are performed in parallel.

In this chapter, we discuss these three processes, along with their sources of uncertainty, and present existing ad hoc methods and mechanisms for *identifying* uncertainties. We dealt with *handling* uncertainties in Madni and Bahill [38]. Next, we present the Overarching Process as a superset of these three processes. Finally, we present a system for ameliorating uncertainty in the Overarching Process. In the approach presented, uncertainty is consistently and uniformly addressed using the Overarching Process as a reference model.

Uncertainty Is Ubiquitous

Uncertainty is ubiquitous in our environment, and occasionally, people deliberately create uncertainty, for example, in games of chance such as playing card games. In card games, one or more decks of cards are shuffled to create a random ordering of cards. Thus, when a card facing down is about to be turned over, neither its suit nor its rank is known for sure (ideally).

In the field of metrology, measuring uncertainty is a core concept that quantifies the measurement error that one should reasonably expect. Uncertainty is involved in every measurement and is represented as significant figures using the number system. Numbers are restricted to only the physically meaningful digits. This quantification of uncertainty is then propagated throughout the calculations so that the uncertainty in the calculated values depends on the uncertainties associated with the measured values and the calculation algorithm.

As important, our understanding of nature is incomplete. Therefore, our models of natural phenomena have uncertainty. The Heisenberg uncertainty principle states that we cannot simultaneously measure both the position and velocity of an electron (or any number of other subatomic particles). As we measure the position ever more accurately, the estimate of the velocity becomes more inexact. Astrophysicists have proposed dark matter and dark energy to patch the holes in our laws of physics that are used to model nature. Even so, our models remain incomplete. Therefore, we still cannot predict the future. ("It's tough to make predictions, especially about the future." Impishly attributed to Yogi Berra.)

In optimization models, uncertainty is used to describe situations where the user does not have full control. Likewise, it is common to include estimates of uncertainty in economic and weather forecasts. Similarly, pollsters employ uncertainty in their models for polling and predicting political elections. In decision science, we employ probabilistic models for human decision-making under uncertainty.

In a systems engineering process, there cannot be a block that says "manage uncertainty" because uncertainty is ubiquitous: It is everywhere. Uncertainty must be managed where and when it occurs. Handling uncertainty is like making a peanut butter and jelly sandwich – just as the peanut butter must be spread over the surface of a slice of bread, so also uncertainty needs to be spread across the whole systems engineering process.

Model-Based System Engineering

System design can be component-based (e.g., WWII battleships), function-based (e.g., 1970s, MIL-STD 499A), requirements-based (e.g., 1990s, MIL-STD 499B), or model-based (e.g., 2000s, OMG, and Estefan [23]). Model-based system engineering and design has the advantage of executable models that improve efficiency and rigor. It also provides a common terminology (ontology), explicit representations, and a central source of truth from which views can be extracted for the needs of particular stakeholders [40]. The earliest development of this technique was in Wayne Wymore's [54] book entitled *Model-Based System Engineering*, although the phrase *Model-Based System Design* was in the title and topics of Jerzy Rozenblit's [50] PhD dissertation. One of the first model-based systems engineering process models was that of Bahill and Gissing [7]. A good summary of the Wymorian process is given in Estefan [23]. Model-based systems engineering depends on having and using well-structured models that are appropriate for the given problem domain [9, 40]. An ancient Chinese proverb that was invented by a New York City journalist a century ago says, "A picture is worth a thousand words." In engineering design, this phrase has morphed into "a model is worth a thousand pictures." This means that models greatly reduce the complexity of a system description. This is akin to design elegance. The complexity is there, but the ability to create views for specific viewpoints enables focusing on issues relevant to different stakeholders. Model elements that are not important to a particular stakeholder can be abstracted or elided.

In this chapter, we derive the Overarching Process that can be used as an enveloping process on top of a model-based system engineering (MBSE) methodology. Other chapters in this handbook (e.g., [23]) show various implementations of

MBSE processes. But this chapter is at a higher level of abstraction. It shows an Overarching Process for MBSE.

The Overarching Process is a top-level, front-end process. Rather than sitting down with the customer on the first day of the project and filling out SysML diagrams, the customer would be better served by starting with the Overarching Process.

> "All the really important mistakes are made the first day," Eb Rechtin, *The Art of System Architecting*, p. 28, [49].

Purpose of Models

Models can be used for many reasons, such as guiding decisions, understanding an existing system, improving a system, creating a new design or system, controlling a system, improving operator performance, suggesting new experiments, guiding data collection activities, allocating resources, identifying cost drivers, increasing return on investment, helping to sell the product, and reducing risk [34]. Running business process models clarifies requirements, reveals bottlenecks, reduces cost, identifies fragmented activities, and exposes duplication of efforts [35].

Kinds of Models

There are different kinds of models in systems engineering. These models address different system perspectives: behavioral, structural, performance, and analysis. *Behavioral models* describe how the system responds to external excitation, that is, how the system functions transform the inputs into outputs. The BaConLaws model [3] is a model of behavior. It describes the linear and angular velocity of baseballs and softballs and baseball and softball bats after the collision in terms of these same parameters before the collision. *Structural models* describe the components and their interactions. Three-dimensional CAD/CAM images check the buildability of structures. *Performance models* describe units, values, and tolerances for properties such as weight, speed of response, available power, etc. These might be captured in requirements. Typical baseball performance measures include batting average, slugging average, and On-base Plus Slugging (OPS). *Analysis models* are used to calculate the properties of the whole system from the properties of its parts. For example, the time for a car to accelerate from 0 to 60 mph can be calculated from the mass of the car, the torque transmitted through the drive train, the aerodynamic drag coefficients, and the friction between the tires and the pavement.

Types of Models

There are many types of models [36]. People generally use only a few and erroneously believe them to be the totality of models because they tend to think of models from their narrow perspectives. Examples of most commonly used types of models include models based on physiological and physical laws and principles, differential equations, difference equations, algebraic equations, geometric representations of physical structure, computer simulations and animations, Laplace transforms, transfer functions, linear systems theory, state-space models (e.g., $\dot{x} = \mathbf{A}x + \mathbf{B}u$), state machine diagrams, charts, graphs, drawings, pictures, functional flow block diagrams, object-oriented models, UML and SysML diagrams, Markov processes, time-series models, physical analogs, Monte Carlo simulations, optimization algorithms, statistical distributions, mathematical programming, financial models, PERT charts, Gantt charts, risk analyses models, tradeoff analyses models, mental models, computer-based story representations, scenario models, and use case models. The appropriate type of model depends on the particular system being studied, the question being asked of the model, the operational context, and the modelers' background.

For example, to understand how people make decisions, at least three phenomena should be accounted for: confirmation bias, attribute substitution, and representativeness. For the biological domain, we must first choose the subject, that is, a virus, a bacterium, a plant, or an animal. Once we have chosen our subject, we can then derive its genome. To model something in the social domain, we might use a novel, an encyclical, a song, a poem, or possibly even a joke.

Most models of real-world phenomena require a combination of these types. For example, Bahill [3] uses Newton's principles, the conservation laws of physics, algebraic equations, spreadsheets, figures, tables, simulations, an optimization package, design of experiments, and statistics. Hence, the BaConLaws model comprises many different types of models.

Tasks in the Modeling Process

In this section, we provide a checklist of the principal tasks or steps that should be performed in a modeling study [2]. Modelers should look at each item on the list to determine if they have done that task. If not, then they should explain why they did not do it. But before explaining our checklist, we must present an example that can be used to illustrate the items on the checklist.

Model for a Baseball-Bat Collision

An effective way to understand these tasks in the modeling process is through an example. A suitable example is one in which the details are quantitative, publicly available, and readily accessible and whose principles are commonly understood by engineers. We decided to use the BaConLaws model for baseball from Bahill [3], Chap. 4. A *very* brief synopsis of this model is presented below. The reader may skip the equations without loss of continuity.

The following equations comprise the BaConLaws model for bat-ball collisions. First, the kinetic energy lost (transformed into heat) during the collision is

$$KE_{\text{lost}} = \frac{1}{2}\frac{m_{\text{ball}} m_{\text{bat}} I_{\text{bat}} \left(v_{\text{ball-before}} - v_{\text{bat-cm-before}} - \omega_{\text{bat-before}} d_{\text{cm-ip}}\right)^2 \left(1 - CoR^2\right)}{m_{\text{ball}} I_{\text{bat}} + m_{\text{bat}} I_{\text{bat}} + m_{\text{ball}} m_{\text{bat}} d_{\text{cm-ip}}^2}, \tag{1}$$

where $d_{\text{cm-ip}}$ is the distance between the bat's center of mass and the impact point, and CoR is the coefficient of restitution, which also models the energy lost.

The linear velocity of the ball after the collision is

$$v_{\text{ball-after}} = v_{\text{ball-before}} - \frac{\left(v_{\text{ball-before}} - v_{\text{bat-cm-before}} - \omega_{\text{bat-before}} d_{\text{cm-ip}}\right)(1 + CoR) m_{\text{bat}} I_{\text{bat}}}{m_{\text{ball}} I_{\text{bat}} + m_{\text{bat}} I_{\text{bat}} + m_{\text{ball}} m_{\text{bat}} d_{\text{cm-ip}}^2}$$
$$\text{where } v_{\text{ball-before}} < 0. \tag{2}$$

The linear velocity of the bat after the collision is

$$v_{\text{bat-cm-after}} = v_{\text{bat-cm-before}} + \frac{\left(v_{\text{ball-before}} - v_{\text{bat-cm-before}} - \omega_{\text{bat-before}} d_{\text{cm-ip}}\right)(1 + CoR) m_{\text{ball}} I_{\text{bat}}}{m_{\text{ball}} I_{\text{bat}} + m_{\text{bat}} I_{\text{bat}} + m_{\text{ball}} m_{\text{bat}} d_{\text{cm-ip}}^2} \tag{3}$$

The angular velocity of the bat after the collision is

$$\omega_{\text{bat-after}} = \omega_{\text{bat-before}} + \frac{\left(v_{\text{ball-before}} - v_{\text{bat-cm-before}} - d_{\text{cm-ip}} \omega_{\text{bat-before}}\right)(1 + CoR) m_{\text{ball}} m_{\text{bat}} d_{\text{cm-ip}}}{m_{\text{ball}} I_{\text{bat}} + m_{\text{bat}} I_{\text{bat}} + m_{\text{ball}} m_{\text{bat}} d_{\text{cm-ip}}^2} \tag{4}$$

Our most succinct presentation of this BaConLaws model is

$$\boxed{\begin{aligned}
&CoR = -\frac{v_{\text{ball-after}} - v_{\text{bat-cm-after}} - d_{\text{cm-ip}} \omega_{\text{bat-after}}}{v_{\text{ball-before}} - v_{\text{bat-cm-before}} - d_{\text{cm-ip}} \omega_{\text{bat-before}}}\\
&\text{where } 0 < CoR < 1\\
&A = \frac{\left(v_{\text{ball-before}} - v_{\text{bat-cm-before}} - d_{\text{cm-ip}} \omega_{\text{bat-before}}\right)(1 + CoR)}{m_{\text{ball}} I_{\text{bat}} + m_{\text{bat}} I_{\text{bat}} + m_{\text{ball}} m_{\text{bat}} d_{\text{cm-ip}}^2}\\
&\text{and } A < 0\\
&v_{\text{ball-after}} = v_{\text{ball-before}} - A m_{\text{bat}} I_{\text{bat}}\\
&v_{\text{bat-after}} = v_{\text{bat-before}} + A m_{\text{ball}} I_{\text{bat}}\\
&\omega_{\text{bat-after}} = \omega_{\text{bat-before}} + A m_{\text{ball}} m_{\text{bat}} d_{\text{cm-ip}}\\
&\omega_{\text{ball-after}} = \omega_{\text{ball-before}}
\end{aligned}} \tag{5}$$

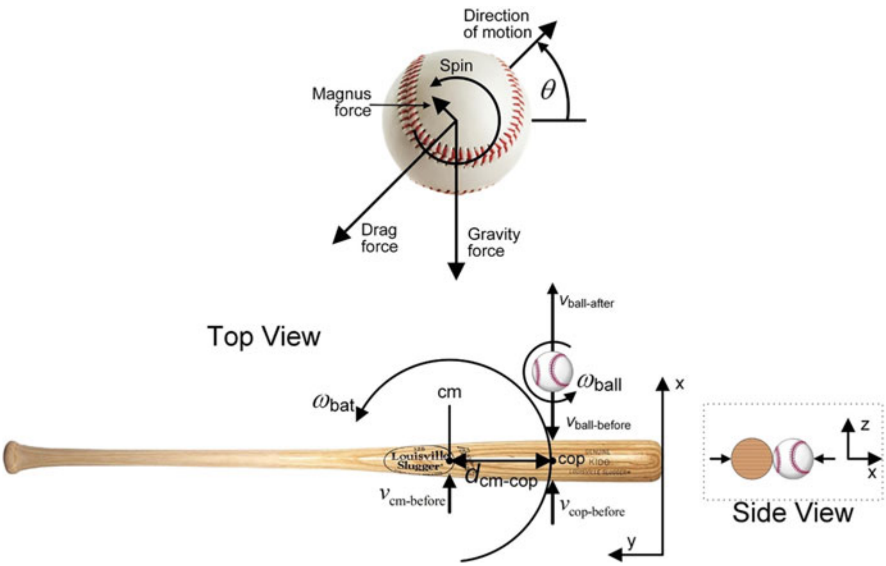


Fig. 1 Forces on a ball-in-flight (top) and a schematic of a head-on bat-ball collision (bottom). The center of mass is cm, and the center of percussion is cop. (© 2019, Bahill. Used with permission)

The numerical value for A is unique for each bat-ball collision. Of course, more complicated models exist, for example, those not described by Fig. 1 and therefore where $\omega_{\text{ball-after}} \neq \omega_{\text{ball-before}}$.

If you do not want to follow the equations of this model, then just imagine watching a baseball game. Your thoughts should fill the squiggly braces in the following checklist. We describe {in squiggly braces} the parts of the BaConLaws model that implement the individual tasks.

Checklist for Tasks Necessary in a Modeling Project

- Describe the system to be modeled. {The BaConLaws model describes head-on collisions between bats and balls, that is, when the bat is going upward at about 10° and the ball is coming downward at about 10° and there is no offset between the bat displacement vector and the ball displacement vector. It gives the velocity and spin of the bat and ball before and after collisions. It does not describe the dynamics *during* the collision nor the swing of the bat.}
- State the purpose of the model. {The purpose of the BaConLaws model was to explain bat-ball collisions with precise, correct equations, without jargon. This included defining the performance criterion function. If the model were being

asked a different question, say about a player's salary contract, an entirely different type of model would have been used, for example, a financial investment model [18, 19].} Here are some baseball models created by physic professors that are not equation-based:
Al Nathan
https://www.scientificamerican.com/article/baseball-physics-opening-day/
David Kagan
https://physics.csuchico.edu/baseball/talks/AAPT(Nov-2012)/slides.pdf
Rod Cross
http://www.physics.usyd.edu.au/~cross/baseball.html
Bob Adair
https://www.amazon.com/Physics-Baseball-3rd-Robert-Adair/dp/0060084367

- Determine the level of the model [11]. {The level for the BaConLaws model encompasses the ball velocity, the bat velocity, and the bat angular velocity after the collision in terms of those same parameters before the collision. The timescale is in milliseconds.}
- State the assumptions and, at every review, reassess the assumptions. {Our assumptions were stated (on pages 24 and 36 of Bahill [3]), and they were reviewed repeatedly.}
- Investigate alternative models. {Many bat swing models were presented in Chap. 1 [3]. Alternative collision configurations were explained in Chaps. 2 and 3. Chapter 3 also presented nine alternative definitions for the sweet spot of the bat. The BaConLaws model was given in Chap. 4, and alternative models were given in Chaps. 5 and 9. Having alternative models helps ensure that you understand the physical system. No model is more correct than another. Alternative models just emphasize different views of the physical system. They are not competing models; they are synergetic.}
- Select tools for the model and simulation. {We used the What'sBest! optimizer, the Pascal language, the Excel spreadsheets, the Math Type equation editor, and MS Word.} This should not be a casual decision. One should not merely accept the default. Tradeoff studies should be used to help select the best tools.
- Make the model. {The BaConLaws model is shown in Eqs. (1) to (5).}
- Integrate with models for other systems. {The outputs of the BaConLaws model became inputs to the ball-in-flight model of Chap. 7 and the Probability of Success model in Chap. 9 of Bahill [3].}
- Gather data describing system behavior. {We used data from our internal databases, peer-reviewed journal papers, and the following online databases:
http://mlb.com/statcast/
https://baseballsavant.mlb.com/statcast_search
https://www.baseball-reference.com/.}
- Show that the model behaves like the real system. {The outputs of the simulations were compared to the data listed in the above paragraph.}

- Verify and validate the model. {*Verification* means, Did you build the system right? For the BaConLaws model, the outputs of the simulations agree with the data listed in the above paragraph. The double checks in the simulation ensured the correctness of the spreadsheets. For example, the kinetic energy lost was computed with equations and also by summing individual kinetic energy components. The conservation laws were used in the derivations, and the final outputs of the simulation were inserted into the conservation law equations to ensure consistency of the spreadsheet. The main output of the BaConLaws model was compared to the output of the Effective Mass model in Chap. 5 of Bahill [3]. The physics was peer-reviewed by two anonymous physics professors. Each of the main BaConLaws equations was derived using at least two techniques. Finally, the equations were checked by an independent mathematician. *Validation* means, Did you build the right system? Our customer wanted a system that described head-on collisions between bats and balls. They wanted a system that would give bat and ball velocity and the bat angular velocity after the collision in terms of those same parameters before the collision. This is what our system does: See Eqs. (2) to (5) above. Finally, we performed a sensitivity analysis, which is a powerful validation tool [30, 53]. It warns if something is wrong with the model. It might also define the boundary conditions for parameters, discover potential brittleness, impact recommended operating procedures, find quirks in how the system must be used, etc.} Enough details should be given to allow other users to replicate your results. If other people cannot replicate your experiments and analysis, then your model fails validation.
- Perform a sensitivity analysis of the model as follows.

Sensitivity Analysis of a Bat-Ball Collision Model

The batter in a game of baseball or softball would like to obtain the maximum batted ball velocity. The larger the batted ball velocity, the more likely the batter will get on base safely. Therefore, we made the batted ball velocity our performance criterion. (Our equations are vector equations. In our analysis, we represented both the magnitudes and directions of the vectors. However, in the book and in this chapter, we only present the magnitudes.)

The linear velocity of the ball after the collision is

$$v_{\text{ball-after}} = v_{\text{ball-before}} - \frac{\left(v_{\text{ball-before}} - v_{\text{bat-cm-before}} - \omega_{\text{bat-before}} d_{\text{cm-ip}}\right)(1 + CoR) m_{\text{bat}} I_{\text{bat}}}{m_{\text{ball}} I_{\text{bat}} + m_{\text{bat}} I_{\text{bat}} + m_{\text{ball}} m_{\text{bat}} d_{\text{cm-ip}}^2} \tag{6}$$

In a simple sensitivity analysis, an input is changed by a small amount, and the resulting change in the output is recorded. For example, in Table 1, when $v_{\text{bat-cm-before}}$ (the velocity of the center of mass of the bat before the collision) was increased by 1%, $v_{\text{ball-after}}$ increased by 0.62%. This was the most sensitive input listed in Table 1.

Table 1 Relative sensitivity analysis

Inputs	Nominal values, SI units	Nominal values, baseball units	$v_{\text{ball-after}}$ when the input was increased by 1%. The nominal value was 91.894 mph	Percent change in $v_{\text{ball-after}}$
$v_{\text{ball-before}}$	−37 m/s	−83 mph	92.066	0.19
$\omega_{\text{ball-before}}$	209 rad/s	2000 rpm	91.894	0
$v_{\text{bat-cm-before}}$	23 m/s	52 mph	92.463	0.62
$\omega_{\text{bat-before}}$	32 rad/s	309 rpm	92.000	0.12
CoR	0.465	0.465	92.450	0.60

Next, the coefficient of restitution, CoR, was set as a constant, and we computed the partial derivatives of the batted ball velocity, $v_{\text{ball-after}}$, with respect to the eight model inputs and parameters. Finally, we used partial derivatives and computed the semirelative sensitivity functions [53].

Table 2 gives the nominal values, along with the range of physically realistic values for collegiate and professional baseball batters, and the semirelative sensitivity values computed analytically. The bigger the sensitivity value, the more important the variable or parameter is for maximizing batted ball velocity.

The equations of the BaConLaws model have the variable $d_{\text{cm-ip}}$ for the distance between the bat center of mass and the impact point. To get numerical values for Table 2, we needed a particular impact point. For this, we used the center of percussion, hence $d_{\text{cm-cop}}$. The variable $vt_{\text{bat-cop-before}}$ is the total velocity, meaning the sum of the linear and angular velocity, of the center of percussion of the bat before the collision.

The right column of Table 2 shows that the most important property (the largest absolute value), in terms of maximizing batted ball velocity, is the linear velocity of the center of mass of the bat before the collision, $v_{\text{bat-cm-before}}$. This is certainly no surprise. The second most important property is the coefficient of restitution, CoR. The least important properties are the angular velocity of the ball, $\omega_{\text{ball-before}}$; the distance between the center of mass and the center of percussion, $d_{\text{cm-cop}}$; and the moment of inertia of the bat, I_{bat}. For our analysis, the sensitivities to the distance between the center of mass and the center of percussion, $d_{\text{cm-cop}}$, and the mass of the ball, m_{ball}, are negative, which merely means that as they increase, the batted ball velocity decreases. The second-order interaction terms, which are not shown, are small, which is good. The results shown in Tables 1 and 2, for two different sensitivity analysis techniques, agree.

{The most important parameters, in terms of maximizing batted ball speed, are the velocity of the center of mass of the bat before the collision and the coefficient of restitution, CoR. The least important parameter is the angular velocity of the pitched ball.}

- Explain a discovery that was not planned in the model's design. {(1) We were surprised when the equation for the kinetic energy lost in the collision fell right

Table 2 Typical values and first-order analytic semirelative sensitivities with respect to the batted ball velocity for the BaConLaws model

	Nominal values		Range of realistic values		$\tilde{S}_{\alpha}^{F} = \frac{\partial F}{\partial \alpha}\Big\|_{\mathrm{NOP}} \alpha_0$
Inputs and parameters	SI units	Baseball units	SI units	Baseball units	semirelative sensitivity values
Inputs					
$v_{\text{ball-before}}$	−37 m/s	−83 mph	−27 to −40 m/s	−60 to −100 mph	8
$\omega_{\text{ball-before}}$	209 rad/s	2000 rpm	209 ± 21 rad/s	2000 ± 200 rpm	0
$v_{\text{bat-cm-before}}$	23 m/s	52 mph	23 ± 5 m/s	52 ± 10 mph	28
$\omega_{\text{bat-before}}$	32 rad/s	309 rpm	32 ± 11 rad/s	300 ± 100 rpm	5
$vt_{\text{bat-cop-before}}$	28 m/s	62 mph			
Parameters					
CoR	0.465	0.465	0.465 ± 0.05	0.465 ± 0.05	25
$d_{\text{cm-cop}}$	0.134 m	5.3 in	0.134 ± 0.05 m	5.3 ± 2 in	−2
m_{ball}	0.145 kg	5.125 oz	0.145 ± 0.004 kg	5.125 ± 0.125 oz	−14
m_{bat}	0.905 kg	32 oz	0.709 to 0.964 kg	25 to 34 oz	10
$I_{\text{bat-cm}}$	0.048 kg-m^2	2624 oz-in^2	0.036 to 0.06 kg-m^2	1968 to 3280 oz -in^2	3

out of the BaConLaws set of equations. (2). Before writing that book, we did not expect to prove that cupping the barrel end of the bat does little good. (3) Although it seems intuitive, we were surprised when the mathematics showed that a baseball could be thrown farther than a tennis ball.}

- Perform a risk analysis. {*Risk to our publisher.* The biggest risk is that people might be reluctant to buy a book with equations in it. Also, Springer would be disappointed if sales were low. Therefore, by writing with the reader in mind, we tried to ensure that sales would not be below expectations. We expect no copyright problems because most of the material is original, and we have permissions for the two figures that are not. *Risk to our reader.* Someone could modify their bat and hurt him or herself by working with tools, or they could be thrown out of a game for using an altered bat. *Risk to the authors*. If our equations were wrong, or if important assumptions were omitted, then we would confuse our readers and tarnish our reputations. *Risk to quality.* The book is produced in India. Typographical and editing mistakes that occur are hard to correct because of poor communication channels. *Risk to baseball managers, general managers, and umpires*. It will put a burden on these people to understand the results of mathematical modeling. *Risk to Major League Baseball (MLB).* It could embarrass MLB into disclosing their algorithms. Some of these risks may seem unlikely. However, one of the most important parts of a risk analysis is exploring unlikely risks.}
- Analyze the performance of the model. {This was described above in the verification paragraph.}
- Reevaluate and improve the model. {In the future, we will explain why the curveball curves. We will also investigate the cognitive processing and decision-making of the batter [4, 5, 8, 10, 42]. We will describe the thrust and parry of the pitcher and the batter.}
- Suggest new experiments and measurements for the real system that might challenge existing models. MLB is providing copious amounts of new data. Next, scientists need MLB's actual algorithms and measurements for the spin on the batted ball, particularly for the home run trajectories that are so popular on television. Another proposed area of measurement and display involves the erratic meandering of fielders trying to catch pop-ups. This behavior and the paper by McBeath et al. [42] show that the ball's trajectory often has bizarre loops and cusps. MLB should show these trajectories on the television screen to help laypeople understand the fielders' wanderings. In the third edition of this book, once we build a gold standard input data set for swings of the bat, we will directly compare the BaConLaws model and other bat-ball collision models.}

In this section, we presented a checklist that should be used to ensure that the most important modeling tasks have been performed. The checklist was exemplified with a model for baseball-bat collisions.

Requirements Discovery Process

We must avoid hearing the common customer complaint, "You gave me what I asked for, not what I wanted!" (Norm Augustine, personal communication)

Figure 2 presents the requirements discovery process. Uncertainty exists in several aspects of this process. The activities that have the most uncertainty are marked with a ▼. To begin with, not all stakeholders can be identified with certainty at the start of the project [33]. The identification of customer needs is a step that

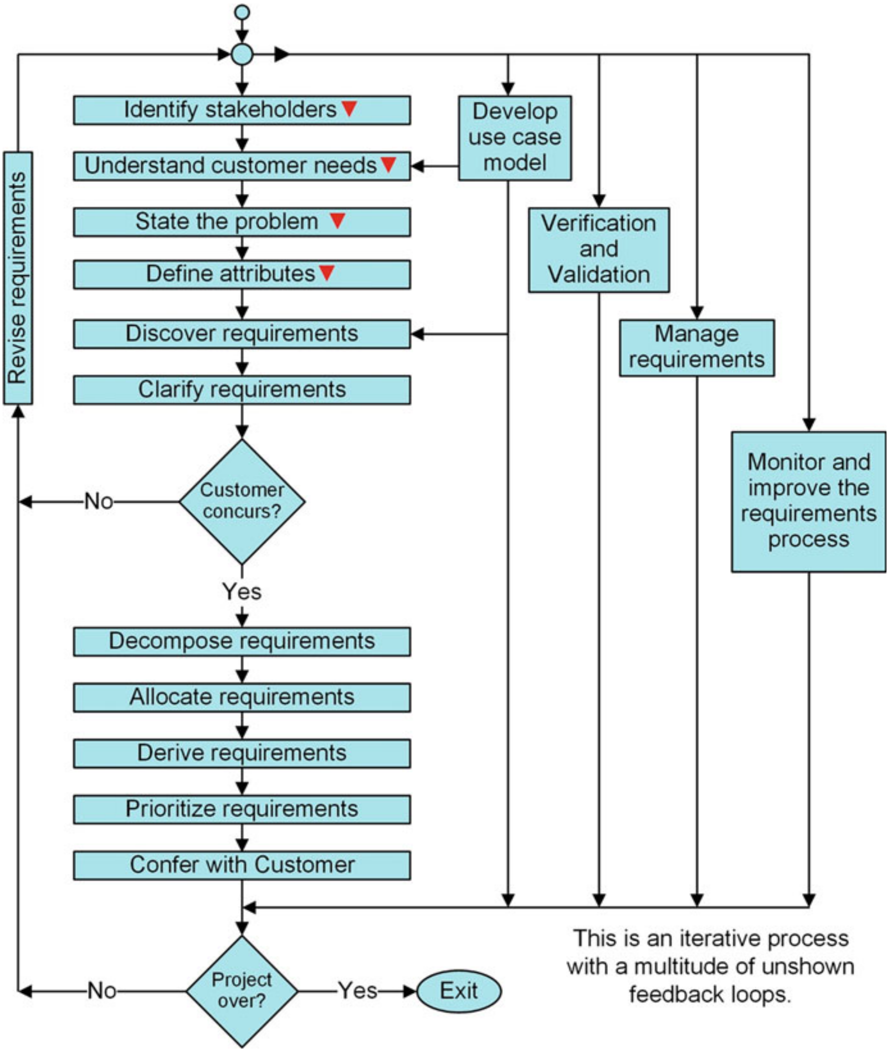


Fig. 2 Requirements discovery process. (Based on Bahill and Madni [8])

requires iterations because uncertainty exists in the initial identification of needs. The same goes for the problem statement. The initial statement of the problem can be expected to be reused and refined as customer needs are more clearly articulated. Therefore, requirements discovery is an iterative process in which each iteration reduces uncertainty in the identification of requirements. Upon customer consensus, the requirements are decomposed, allocated, further refined (derived), prioritized, validated, and finalized. Iteration is the primary means of reducing uncertainty in the requirements discovery and derivation task. Query reformulation is the primary means for reducing uncertainty in the problem statement task. Uncertainty in program schedule and technology maturation is addressed through incorporating schedule buffers and safety margins.

The requirements discovery process has a multitude of unshown feedback loops. For example, the Manage Requirements Activity has inputs to all of the mainline activities, such as discover requirements, clarify requirements. decompose requirements, allocate requirements, derive requirements, and prioritize requirements.

Where Do Requirements Come From?

Requirements come from project stakeholders [6]. Stakeholders include, among others, end users, operators, surrogate customers, managers, sponsors, staff members, testers, maintainers, bill payers, regulatory agencies, potential victims, and systems that will interact with your system [17, 27]. Many requirements can be derived from previous systems. And if you are lucky, requirements can come from the use cases, as shown in the following sections.

A Use Case Template

While the use case diagram is simple, the use case package is complex. It is inadequately explained in most books and papers. Therefore, we start our system development with a formal use case template. Less formal descriptions are called stories.

A *use case* is an abstraction of the required functions of a system. A use case usually produces an observable result of value to the user. Each use case describes a sequence of interactions between one or more actors and the system [47]. Our design process is use case-based.

Name: A use case should be named with a verb phrase in the active present tense form. It should not relate to any particular solution.

Iteration: This is configuration management. Sometimes, we just number them.

Derived from: Explain the source for the use case. For example, it might be the mission statement, the concept of operations (ConOps), a business use case, or a customer requirement.

Brief description: Describe the general sequence that produces an observable result of value to the user.

Level: The amount of detail required in the use case. Do not mix classes of different levels in the same use case.

Priority: The importance of this use case relative to other use cases

Scope: This defines the boundary of what the use case applies to.

Added value: Describe the benefit (usually) for the primary actor. This is an important slot.

Goal: The goal is the behavior that the primary actor expects to get from the use case. You should have a goal or an added value, but probably not both.

Primary actor: Actors are named with nouns or noun phrases. Actors reflect the roles of things outside the system that interact with the system. Primary actors initiate the functions described by use cases.

Supporting actor: Supporting (or secondary) actors are used by the system. They are not a part of the system and thus cannot be designed or altered. They often represent external systems or commercial-off-the-shelf (COTS) components. If your system changes them, then those effects are unintended consequences.

Frequency: How often is the use case likely to be used? When this slot is helpful, it is *very* helpful. When it is not, do not use it.

Precondition: The precondition should contain, among other things, the state of the system and values for pertinent attributes before the main success scenario starts.

Trigger: The trigger should contain the event that causes a transition from the preconditioned state to the first step in the main success scenario.

Main success scenario:

1. This numbered set of steps illustrates the usual, successful interactions of actors with the system. Usually, the first step states the action of the primary actor that starts the use case.

2a. The last step tells you where to go next (e.g., exit use case).

Alternate flows:

2b. Alternate flows describe failure conditions and unsuccessful interactions (exit use case).

The main success scenario and the alternate flows can contain diagrams, such as sequence and activity diagrams.

Postcondition: Describes the state of the system after exit from the use case no matter which flows were executed. This is hard to write.

Specific Requirements

The steps in the main success scenario should suggest the functions that the system is supposed to perform. From these, we should be able to write system requirements.

Functional requirements: Describe the functional requirements with shall statements.

Nonfunctional requirements: Describe the nonfunctional (often performance) requirements with shall statements.

Requirements are sometimes quantified with scoring functions ([54] pp. 385–397; [8], [21], pp. 246–258).

Author/owner: This is an important field. It tells you whom to talk to if you want to change the use case.

Last changed: Use some form of configuration management such as the date of the last change or the revision history.

No standard specifies which slots should be in a use case description. Your minimal set should be based on your company requirements template. The number of slots and the detail in each slot increase as the design progresses from the requirements model to the analysis model to the design model to the implementation model. Other useful slots contain rules, assumptions, and extension points. Do not use slots that do not help you. If you find that the trigger, precondition, or post-condition does not help you to create state machine diagrams, then do not use them. A use case description is also called a use case report, a use case narrative, and stories. A use case package contains a use case description, sequence diagrams, activity diagrams, supplementary requirements, and other UML stuff [47]. Other chapters in this handbook use other templates.

A Use Case Example from a Chocolate Chip Cookie-Making System

Imagine that while reading this book, you experience an irresistible urge for chocolate chip cookies. Frantically, you rummage your kitchen. Lo and behold, you find a tube of chocolate chip cookie dough in your refrigerator! Assume that you have a typical kitchen – a stove, an oven, a timer, pots and pans, utensils, and, of course, cookie sheets. And you have the all-important tube of Pillsbury's Chocolate Chip Cookie dough, with these instructions printed on the label:

- Preheat oven to 350 °F.
- Spoon heaping teaspoons of well-chilled dough about 2 in. apart onto a cool ungreased cookie sheet.
- Bake at 350 °F for 10 min.

You are in business!

Write a use case that will describe how your system should work.

Name: Bake my Cookies

Note: For clarity, we set use case names in a different font, probably Verdana.

Iteration: 3.1

Derived from: Problem statement

Brief description: The cookie-making system bakes cookies to perfection. It is named Cookie.

Level: High

Priority: This use case is of the highest priority.

Scope: A typical home kitchen with pots, pans, utensils, etc.

Added value: Students' brains always work better with a tummy full of cookies.

Goal: To produce stupendous freshly baked cookies
Primary actor: Student
Supporting actors: A tube of Pillsbury's Chocolate Chip Cookie dough
Frequency: Once a month
Precondition: All ingredients and cookware are available.
Trigger: Student gets "hungry" for cookies.
Main success scenario:

1. Student decides to bake cookies.
2. Student turns on the oven and sets the desired oven temperature to 350 °F.
3. Cookie increases the temperature in the oven.
4. Student gets a tube of Pillsbury's Chocolate Chip Cookie dough out of the refrigerator and spoons heaping teaspoons of well-chilled dough about 2 in. apart onto an ungreased cookie sheet. Your Mom would probably worry about the expiration date of the product, which is a possible risk.
5. Cookie signals that the oven has preheated to 350 °F.
6. Student puts the cookie sheet full of cookies into the oven and sets the timer for 10 min.
7. Cookie signals that the baking time is over. If this signal is erroneously too late, the cookies could burn.
8. Student takes the cookies out of the oven. The cookie sheet will be hot. Student must wear an oven mitt. Student puts the cookies on a cooling rack and turns the oven off. Failing to turn the oven off creates a risk.
9. Student eats the cookies and notes their quality (exit use case).

Unanchored alternate flow: At any time, Student can abort the process and turn off the oven (exit use case).

Postcondition: The kitchen is a mess, but the oven is off.

The steps in the main success scenario suggest the functions that the system is supposed to perform. From these steps, we can write the following system requirements. This process was developed by Daniels and Bahill [20].

Specific Requirements
Functional requirements:

ReqF1: Cookie shall provide a mechanism for Student to enter the desired baking time. The abbreviation ReqF* means a functional requirement.
ReqF2: Cookie shall display the desired baking time entered by Student.
ReqF3: Cookie shall heat the oven from room temperature to 350 °F in less than 5 min.
ReqF4: Cookie shall calculate and display the remaining baking time.
ReqF5: Cookie shall emit an audible signal when the oven is preheated and when the baking time is over.
ReqF6: Cookie shall visually indicate when the oven is preheated and when the baking time is over.

ReqF7: Cookie shall execute Built-in Self-Tests (BiST) (derived from company policy).
ReqF8: Cookie shall have a hard upper limit on oven temperature at 550°, even during self-cleaning (derived from the risk analysis). Rationale: This will help prevent fire.
ReqF9: Cookie shall turn off the oven when it is no longer being used (derived from the risk analysis). Implementation could be (1) turning the oven off 20 min after the end of the timer interval, (2) turning on an alarm 20 min after the end of the timer interval, and (3) turning the oven off 20 min after the end of the timer interval if there is no food inside of the oven.

Nonfunctional requirements:

ReqNF1: The remaining baking time displayed by Cookie shall be visible to a Student with 20/20 vision standing 5 ft from the oven in a room with an illuminance level between 0 and 1000 lux. The abbreviation ReqNF* means a nonfunctional (usually performance) requirement.
ReqNF2: Cookie shall raise the temperature of food in the oven so that temperatures at two distinct locations in the food differ by less than 10%.
ReqNF3: Cookie shall update the remaining baking time display every minute.
ReqNF4: The audible signal emitted by Cookie shall have an intensity level of 80 ± 2 decibels (dB) at a distance of 30 cm and a frequency of 440 Hz. Note: "Goalposts" like this, where all values inside the limits are accepted and all values outside the limits are rejected, are no longer fashionable for requirements since Taguchi (see [48]) scoring functions like that shown in Fig. 6 are preferable ([8], pp. 386–389).
ReqNF5: Cookie shall comply with section 1030 of Title 21, Food and Drugs, Chapter I – Food and Drug Administration, Department of Health and Human Services, Subchapter J: Radiological Health.
ReqNF6: The desired baking time shall be adjustable between 1 min and 10 h.

Author/owner: Hungry Student
Last changed: January 5, 2021

Requirements must be necessary, verifiable, unambiguous, etc. Bahill and Madni ([8], pp. 379–386) list, with explanations, 28 such characteristics of good requirements.

A Test Plan for This System

An important part of the incipient system design is describing how the system will be tested. Fortunately, with use cases, this is a simple task. To test means to apply inputs, measure and record outputs, compare outputs to requirements, and finally indicate passing status.

This test plan is based on the main success scenario of the Bake My Cookies use case.

1. Tester turns on the oven and sets the desired temperature to 350 °F.
2. Tester waits until Cookie signals that the oven has preheated to 350 °F.
3. Tester stands 5 ft from the oven and observes the visual display. He measures the sound intensity and the frequency of the auditory signal from a distance of 30 cm. He measures the actual temperature inside the oven. He records the results.
4. Tester sets the timer for 10 min.
5. Tester waits until Cookie signals that 10 min is over.
6. Tester stands 5 ft from the oven and observes the visual display. He or she measures the sound intensity and the frequency of the auditory signal from a distance of 30 cm. He or she measures the actual temperature inside the oven. He or she notes the desired and actual elapsed time (10 min) and records all of the results.
7. Tester turns the oven off.
8. Tester notes that the oven temperature is decreasing (end of test).

This series of steps can easily be converted into an activity diagram or a sequence diagram. Some chapters in this handbook may skip the text and go directly to an activity diagram or a sequence diagram.

This section has presented a high-level or front-end process for discovering system requirements. It has not shown how to document requirements at a low level: For this, relational databases or SysML diagrams can be used. Friedenthal et al. [25] and Madni and Sievers [41] show examples of using Use Case Diagrams (uc), Requirements Diagrams (req), Sequence Diagrams (sd), Activity Diagrams (act), Block Definition Diagrams (bdd), Package Diagrams (pkg), and requirements tables and matrices to show the implementation of requirements documentation at a low level.

Tradeoff Study Process

When a decision is important, a formal tradeoff study may be in order [22, 31]. Decisions that may require formal tradeoff studies include bid/no-bid, make-reuse-buy, formal inspection versus checklist inspection, tool and vendor selection, incipient architectural design, hiring and promotions, and helping your customer to choose a solution from among various alternatives.

A tradeoff study is not something that is done once at the beginning of a project. Throughout a project, you are continually making tradeoffs such as creating team communication methods, selecting components, choosing implementation techniques, designing test plans, and maintaining the schedule. Many of these tradeoff decisions should be formally documented.

Companies should have criteria for when to do formal decision analysis, such as:

- When the decision is related to a moderate- or high-risk issue
- When the decision affects work products under configuration management
- When the result of the decision could cause significant schedule delays or cost overruns
- On material procurement of the 20% of the parts that constitute 80% of the total material costs
- When the decision is selecting one or a few alternatives from a list
- When a decision is related to major changes in work products that have been baselined
- When a decision affects the ability to achieve project objectives
- When the cost of the formal evaluation is reasonable when compared to the decision's impact
- On design-implementation decisions when technical performance failure may cause a catastrophic failure
- On decisions with the potential to significantly reduce design risk, engineering changes, cycle time, or production costs

Killer trades are used to eliminate a large number of possible alternatives in one fell swoop. When evaluating alternatives is expensive, then early in the tradeoff study, you should identify important requirements that can eliminate many alternatives. If these requirements are performance related, then they are called key performance parameters (KPP). These requirements produce killer criteria. Subsequent killer trades can often eliminate, if you are lucky, maybe 90% of the possible alternatives.

In the Cookie Acquisition System, a killer criterion is that the cookies must be chocolate chip. Gingerbread, oatmeal, bonbons, rum balls, animal crackers, biscotti, ladyfingers, macarons, etc. will not do. This eliminates maybe 99% of possible cookies.

Alternative solutions should be suggested by stories, use cases, and the concept of operations (ConOps). Everybody should suggest alternatives and criteria during brainstorming sessions and in private contemplation. It is important to get many alternative solutions and criteria and then eliminate most. Bizarre alternatives should suggest new requirements.

Figure 3 presents the tradeoff study process. The activities that have the most uncertainty are marked with a ▼. The first step has several sources of uncertainty. It is inevitably the case that the initial problem statement is imprecise and the tradeoff space (alternative solutions) initially defined is incomplete. These sources of uncertainty need to be addressed before proceeding with tradeoff studies. Probing the statement of the problem, reformulating queries, and identifying new variables that need to be included in the tradeoff space are the means for reducing uncertainty. Thereafter, the steps are relatively straightforward.

In this chapter, we have assumed that the reader is familiar with the requirements discovery process, the tradeoff study process, and the risk analysis process. However, the tradeoff study process has a few subtleties that some readers may not be cognizant of. Other chapters in this handbook on MBSE do not recognize these

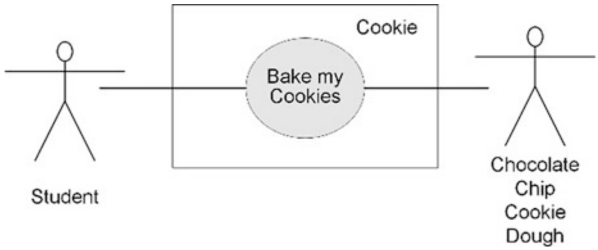


Fig. 3 A use case diagram for the Cookie Acquisition System. The chocolate chip cookie dough is shown as a secondary actor. More formally, the refrigerator might be the secondary actor, and it stores the chocolate chip cookie dough until it is requested

subtleties in doing tradeoff studies. So we offer a simplified example of a tradeoff study in this next section.

Tradeoff Study Example from a Chocolate Chip Cookie Acquisition System

Imagine that while reading this chapter, you experience an irresistible urge for chocolate chip cookies and a glass of milk. You can simply state this as, "I want chocolate chip cookies." You begin to explore how to get hold of chocolate chip cookies to satisfy this urge. You quickly discover that there are no chocolate chip cookies in your home. You do have yogurt, but that does not help. You begin to explore your options. You could head over to a bakery and buy chocolate chip cookies. But wait! That would cut into valuable study time. You simply cannot afford to do that! How about having a pizza delivered instead? No! You want *chocolate chip cookies*. Frantically, you start rummaging the kitchen. Lo and behold, you find a tube of chocolate chip cookie dough in your refrigerator! You are going to *make* chocolate chip cookies! However, is that the best alternative? Perhaps you need to do a tradeoff study.

The following is a tiny excerpt of a tradeoff study for the Chocolate Chip Cookie Acquisition System. The complete example is given in Bahill and Madni [8], pp. 15–30, 469–474, and 616–618). The question that this tradeoff study will answer is, "What is the best way for our student to get chocolate chip cookies?" First, is formal evaluation necessary? Our customer, the student, says that this is important. Therefore, we *will* do a tradeoff study. We will use the following three evaluation criteria which are derived from use case descriptions. Evaluation criteria are often called measures of effectiveness.

Name of criterion: Audible signal indicating cookies are ready

Description: An audible signal shall indicate when the cookies are ready. This signal should have a nominal intensity level of 80 ± 2 decibels.

Weight of importance: 9

Basic measure: Intensity level of an audible signal

Measurement method: During design and construction, the proposed device will be mounted on a test bench and will be activated following test instructions. The sound intensity level in decibels (dB) will be measured at a distance of 30 cm. At the final test and during operation, an actual oven will activate the audible signal, and the sound intensity level in decibels will be measured at a distance of 30 cm.

Units: Decibels (dB)

Trace to functional requirement, ReqF5, in the Bake My Cookies use case ([8], pp. 19–21)

Owner: Engineering

Date of last change: 12/25/2020

Name of criterion: Lost study time

Description: While the Student is making cookies, driving to the bakery to buy cookies, or bargaining with his mother to get her to make cookies for him, he will not be studying. This lost study time is the criterion. **Comment:** For optimal learning, students do need breaks.

Weight of importance: 7

Basic measure: Amount of study time lost in getting the cookies

Measurement method: During design, this lost study time will be calculated by analysis. At the final test and during operation, this lost time will be measured.

Units: Minutes

Scoring function: This criterion requires a scoring function that changes "lost study time" into a "more is better" situation.

Trace to the concept of operations

Owner: Student

Date of last change: 7/8/2020

Name of criterion: Nutrition

Description: Four cookies (2 oz) should contain less than 520 calories, 24 g of fat, and 72 g of carbohydrates.

Weight of importance: 5

Basic measure: Calories, grams of fat, and grams of carbohydrates

Measurement method: Use data from the Internet, for example, http://www.pillsbury.com/products/cookies/refrigerated-cookies/chocolate-chip.

Units: Calories, grams of fat, and grams of carbohydrates

Trace to the concept of operations

Owner: Student

Date of last change: 7/8/2020

Next, we consider these three alternatives:

- Ask your mother to bake cookies for you.
- Use a tube of Pillsbury refrigerated chocolate chip cookie dough.
- Go to the bakery to buy chocolate chip cookies.

Evaluation data (weights and scores) come from expert opinion and measurements during trips to a grocery store and a bakery.

The method for combining the data will be the sum of weighted scores combining function or simply the sum combining function. It is the simplest method for combining data.

The *sum combining function* is

$$f = \sum\nolimits_{i=1}^{n} w_i \times x_i. \tag{7}$$

In this equation, n is the number of evaluation criteria to be combined, x_i is the output of a scoring function (with values from 0 to 1) for the ith evaluation criterion, and w_i is the normalized weight of importance for the ith evaluation criterion. Weights of importance are expected to be normalized and vary from zero to one. The resulting scores are multiplied by their corresponding weights. The output of the function is the sum of the weight-times-score for each evaluation criterion. This sum combining function is commonly used, for example, when computing the grade point average of a student.

In a simple specific case, where there are only two evaluation criteria,

$$\begin{aligned} &\text{named } y \text{ and } z \text{ with } n = 2 \text{ then} \\ &f = w_y y + w_z z. \end{aligned} \tag{8}$$

This function is used when the evaluation criteria show perfect compensation, that is, when both criteria contribute to the result and when more of y and less of z is just as good as less of y and more of z. Stated formally, the sum combining function is appropriate when the decision-makers' preferences satisfy additive independence, which is the case for most industry examples that we have seen.

The question to be answered by the tradeoff study is, "What is the best way for our student to get chocolate chip cookies?"

These evaluation criteria and alternatives were put into a spreadsheet as shown in Table 3. We think the organization is clear enough that we do not have to explain each cell. The exceptions are rolling up the subcriteria into the criteria. So we will explain one of those.

The following numbers are in the blue-shaded cells of Table 3. Calories for Mom's cookies were given a relatively high score of 0.7 (more is better) because Mom won't make unhealthy cookies. The subcriteria calories was given a weight of 9, which became a normalized weight of 0.47. This weight was then multiplied by the score of 0.7 to produce a product of 0.33. Fat and carbohydrates were treated similarly to give numbers of 0.11 and 0.19. Those three numbers were added together to give the criteria nutrition a score of 0.63. This score was multiplied by its normalized weight of 0.24 to give a final score of 0.15.

This tradeoff study shows that Mom's cookies are the preferred alternative. The do-nothing alternative is ranked high, which is worrisome. This probably happened because we do not have any performance criteria, such as Anticipated Tastiness.

Table 3 Tradeoff study matrix for the Chocolate Chip Cookie Acquisition System

Hierarchal tradeoff study matrix for the Cookie Acquisition System

Evaluation criteria	Weight of importance	Normalized evaluation criteria weights	Subcriteria weight of importance	Normalized subcriteria weights	Alt 1 Do nothing			Alt 2 Ask Mom			Alt 3 Pillsbury's Chocolate Chip Cookie Dough			Alt 4 Buy cookies at bakery		
					sc	w x sc	w x sc	sc	w x sc	w x sc	sc	w x sc	w x sc	sc	w x sc	w x sc
Audible signal for cookies are ready	9	0.43			0.00		0.00	0.65		0.28	0.50		0.21	0.40		0.17
Lost study time	7	0.33			1.00		0.33	0.80		0.27	0.50		0.17	0.30		0.10
Nutrition	5	0.24			1.00		0.24	0.63		0.15	0.48		0.11	0.35		0.08
Calories			9	0.47	1.00	0.47		0.70	0.33		0.50	0.24		0.40	0.19	
Fat, grams			4	0.21	1.00	0.21		0.50	0.11		0.40	0.08		0.30	0.06	
Carbohydrates, grams			6	0.32	1.00	0.32		0.60	0.19		0.50	0.16		0.30	0.09	
Alternative ratings							**0.57**			**0.70**			**0.50**			**0.35**
Column sum		1.00		1.00												

These are the abbreviations used in this table: sc stands for the score, wt stands for weight, × is the multiplication sign, norm means normalized, and Alt means alternative

This study was criticized because (1) of its lack of performance and cost evaluation criteria, (2) the customer's desire for milk was ignored, (3) nothing was stated about being healthy! That would have posed an interesting contradiction in the analysis!

This tradeoff study was reviewed by the authors of this chapter and was stored in the process assets library (PAL). Some chapters in this handbook call this the model repository.

Risk Analysis Process

Figure 5 presents the risk analysis process. Activities with the highest uncertainty are marked with a ▼. Once again, preparing for risk requires the identification of risk events. Risks are explored by the risk analyst working with domain experts using storytelling, use cases, mental exploration, and envisioning methods to identify risk events. (This, of course, is not a complete list of the types of analyses used in risk assessment. There are many whole books filled with risk assessment techniques and disaster analyses. Most of the MBSE papers in this handbook devote pages to such analyses. An interesting problem with typical statistical analyses used in risk analyses is that the results may point to incorrect correlations that throw off risk determination. One use for a good model is that it could be useful in poking at causality which makes risk predictions more reliable. Furthermore, likelihood is almost always wrong – it may be high or low but generally wrong (Mike Sievers, personal communication).) Once a comprehensive set of risk events is identified, the risk analysis process proceeds following the steps laid out in Fig. 5.

Risk Analysis Example from a Chocolate Chip Cookie Making System

Risk management starts with the incipient system design. In this section, we (1) construct a risk register identifying and evaluating the major risks, (2) identify early tests or other measures that could be used to mitigate at least some of the major risks, (3) describe a few potential unintended consequences of this system, and (4) describe a few BiST for this system.

Table 4 shows some of the risks to the system and its primary actor. Likelihood expresses our feelings about how likely this failure event is, on a scale of 0 to 1. Severity expresses our feelings about how severe this failure event is, on a scale of 0 to 1. Risk is the product of likelihood and severity.

The biggest risk is incorrect ingredient quantities, which is why people prefer to use a tube of Pillsbury's Chocolate Chip Cookie dough instead of starting from scratch. Next, we identify the most likely outcome and the most severe outcome. Gaining weight is the most likely event that needs to be mitigated, if convenient. A fire in the oven is the most severe event; therefore, we need to keep an eye on the oven.

Table 4 A risk register

Risk no	Potential failure event	Consequences	Likelihood per system usage	Severity	Risk
1	Incorrect ingredient quantities when baking from scratch	Off-taste; bad consistency	0.09	0.5	**0.045**
2	Product is out of date	Possible illness	0.002	0.9	0.018
3	Oven temperature too high	Burned cookies	0.009	0.09	0.0008
4	Baking time too long	Burned cookies	0.009	0.09	0.0008
5	Student could gain weight	Student would be unhappy	0.1	0.1	0.01
6	Failure to turn oven off	The oven could start a fire	0.01	0.95	0.0095
7	Student does not wear an oven mitt and burns his or her hand	Student is unhappy	0.04	0.8	0.032

Risk is defined as likelihood times severity

Now, we should identify early tests or other measures that could be used to mitigate the major risks.

Possible early mitigation measures include:

1. Check “use by” dates on ingredients.
2. Use an independent (not a part of the oven) thermometer.
3. Use an independent (not a part of the oven) timer.
4. Cutting back on the quantity or quality of the butter or sugar is not a reasonable mitigation strategy.
5. Build in a governor that will restrict the maximum oven temperature to 550 °F.

It is important to also identify potential unintended consequences of the system. Two possible negative unintended consequences are (1) the aroma could attract undesirable neighbors to crash into your home and (2) heat generated by the oven could heat the house (particularly undesirable in the summer). This could ultimately contribute to global warming!

All systems should have Built-in Self-Test (BiST). Amazingly, a simple oven system has at least five BiSTs: (1) The clock indicates that the oven is connected to electricity, (2) the oven-on light indicates that the function select knob is operable, (3) the preheated indicator shows that the oven heats up, (4) an internal temperature probe indicates that the oven is functioning, and (5) the Student being vigilant by looking for smoke and smelling for cookies burning protects against catastrophic failures.

Figure 5 presented the risk analysis process. Risk events are identified by the risk analyst working with domain experts using storytelling and use cases. Once a comprehensive set of risk events is identified, they are prioritized and stored in the risk register.

Comparing the Requirements, Tradeoff, and Risk Processes

As noted earlier, requirements discovery, tradeoff studies, and risk analyses employ the same underlying structure in their processes. They appear different because they employ different vocabularies, inputs, and outputs. In this section, we explore the commonalities among these three processes. These are the actual phrases that were used in Figs. 2, 4, and 5. The processes that are shown in these figures were developed years before the Overarching Process was conceived.

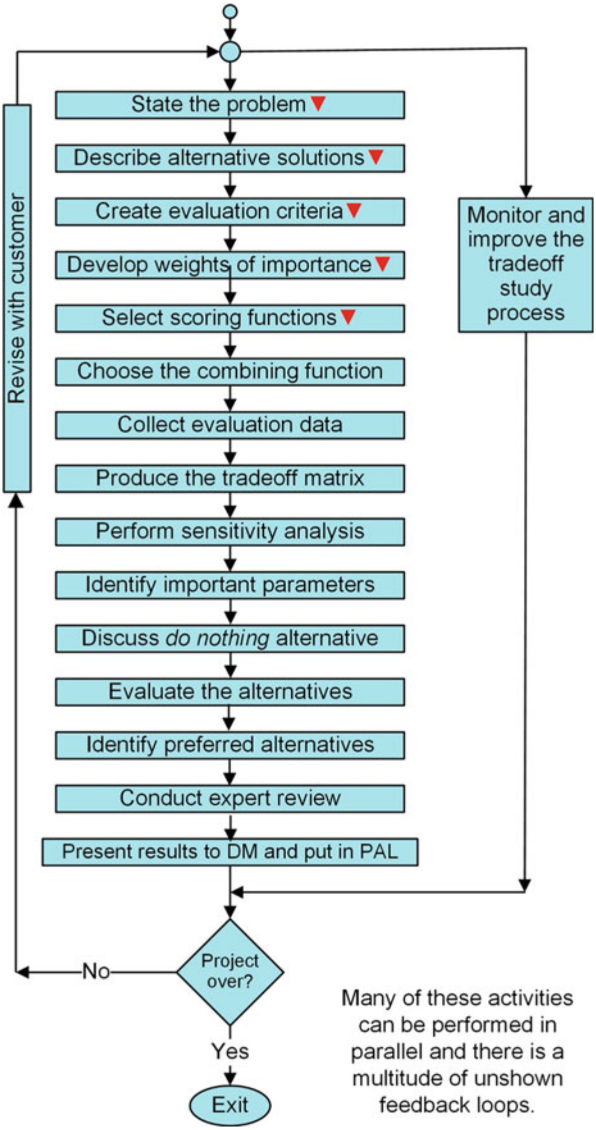
Comparing the *Activities* of the Requirements, Tradeoff, and Risk Processes

Table 5 compares the requirements discovery, tradeoff studies, and risk analysis processes in terms of the specific vocabularies used in each activity. It is important to note that we did not expect a one-to-one mapping between requirements discovery, tradeoff studies, and risk analysis because they are distinctly different processes employed in engineering design and systems engineering. However, it is striking how closely they correspond to each other from the perspective of process structure. The comparison of these three well-established processes has an added benefit — it facilitates process improvement. For example, comparing the rows in Table 5 might suggest that the requirements process should include an activity for expert review and that all processes should start with an understanding of customer's needs. In other words, activities in each row can be evaluated in terms of their relevance (or not) for the other processes. This activity is intended to introduce consistency and uniformity into the three processes, making it easier to address uncertainty sources and uncertainty handling mechanisms in a consistent, uniform way.

Many chapters in this handbook do not use the verify relationship. In MBSE, verification methods "satisfy" a requirement or possibly "trace" to a requirement but importantly are maintained separately. Consider that we modify a verification method in a real project. Doing that will mean a number of steps including stakeholder reviews and actions by a configuration control board (CCB). Since verification methods typically come later in the development life cycle and not all at once, we are constantly control boarding the requirements document. Rather, we want the verification methods to be kept somewhere else and managed separately so that we can make changes as needed without changing requirements. Another point is that the verification method does not add value for designers, especially since there are usually separate teams dealing with requirements and verification and validation (personal communication Mike Sievers, 2021).

Table 6 shows the arguably most important row of Table 5.

Fig. 4 Tradeoff study process. (© 2017, Bahill and Madni. Used with permission)



Comparing the *Products* of the Requirements, Tradeoff, and Risk Processes

Next, we compare the key products of the requirements discovery, tradeoff studies, and risk analysis processes. These *products* appear to be quite different because they employ different vocabularies and structures.

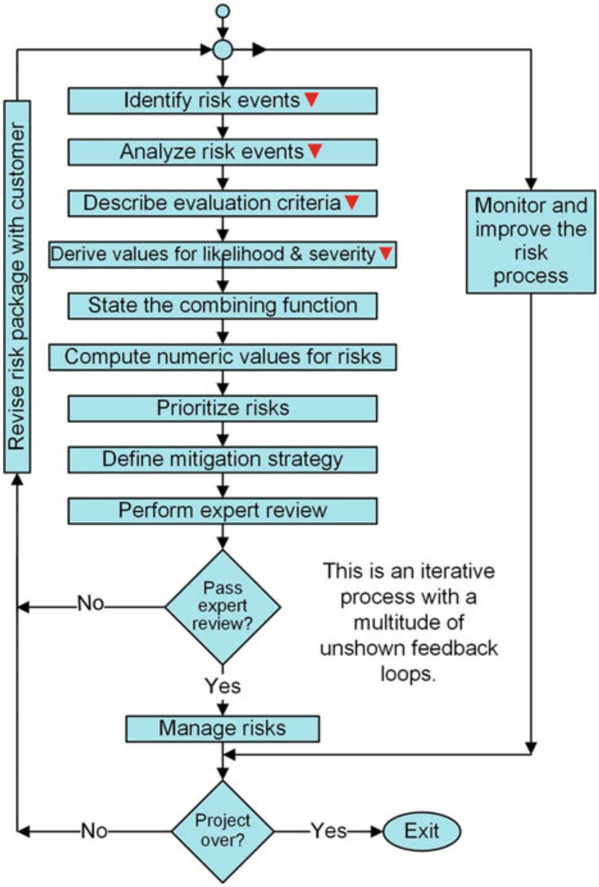


Fig. 5 Risk analysis process. (© 2017, Bahill and Madni. Used with permission)

In the following paragraphs, we discuss the foregoing within the context of a Chocolate Chip Cookie Acquisition System [8]. The Chocolate Chip Cookie Acquisition System is a system that allows a student on an afternoon study break to acquire chocolate chip cookies using one of several approaches. This illustrative example is used to convey the key terms associated with our three processes.

A Requirement from the Chocolate Chip Cookie Acquisition System

Table 7 presents a requirement from the Chocolate Chip Cookie Acquisition System.

Table 5 Comparison of the vocabularies used in the activities of these three processes

Terms used in the requirements discovery process, Fig. 2	Terms used in the tradeoff studies process, Fig. 4	Terms used in the risk analysis process, Fig. 5
Identify stakeholders, understand customer needs, state the problem, and develop use case models	State the problem	Identify risk events
Discover requirements	Describe alternative solutions	Analyze risk events
Define attributes	Create evaluation criteria	Derive values for likelihood and severity
Prioritize requirements	Develop weights of importance	Adjust the range of the criteria
Choose utility functions	Select scoring functions	Choose utility functions
Imply the Boolean AND function	Choose the combining function, usually the weighted sum	State the combining function, usually the product
	Collect evaluation data	
	Produce tradeoff matrix	
Identify cost drivers	Perform sensitivity analysis and identify important parameters	Perform sensitivity analysis
	Discuss the *do-nothing* alternative	Identify acceptable risks
	Evaluate the alternatives	Compute numerical values for risks
Clarify, decompose, allocate, and derive requirements		
Revise requirements with the customer	Revise with customer	Revise risk package with customer
		Manage risks
Prioritize the requirements set (find the most important requirements)	Identify preferred alternatives	Prioritize risks (find the greatest risks)
Test, verify, and validate		
Review with the customer and perform formal inspections	Conduct expert review	Perform expert review
	Track the marketplace for new alternatives	Manage risks
Manage requirements (put results in a requirements database)	Present results to decision-maker (DM) and put in PAL	Put results in the risk register
	Choose the combining function	Track outliers (both high frequency but low severity and low frequency but high severity)
Monitor and improve the requirements process	Monitor and improve the tradeoff study process	Monitor and improve the risk process

(continued)

Table 5 (continued)

Terms used in the requirements discovery process, Fig. 2	Terms used in the tradeoff studies process, Fig. 4	Terms used in the risk analysis process, Fig. 5
Inputs		
Requirements come from use cases and stakeholders	Alternatives, evaluation criteria, weights, and scores that come from use cases and the ConOps	Risks come from use cases and are identified by the risk analyst
Outputs		
Requirements specification	Preferred alternatives	Risk register

Table 6 This row from Table 5 might be the most important row

The term used in requirements discovery	The term used in tradeoff studies	The term used in risk analysis
Prioritize the requirements set (find the most important requirements)	Identify preferred alternatives	Prioritize risks (find the greatest risks)

An Evaluation Criterion from the Chocolate Chip Cookie Acquisition System

Name of criterion: Audible signal indicating cookies are ready

Description: An audible signal shall indicate when the cookies are ready. This signal should have a nominal intensity level of 80 $\pm$ 2 decibels.

Weight of importance 9

Basic measure: Intensity level of an audible signal

Measurement method: During design and construction, the proposed device is mounted on a test bench and is activated following test instructions. The sound intensity level in decibels (dB) is measured at a distance of 30 cm. At the final test and during operation, an actual oven activates the audible signal, and the sound intensity level in decibels is measured at a distance of 30 cm.

Units: Decibels (dB)

Scoring function input: The measured sound intensity will probably lie between 70 and 90 dB.

Scoring function: SSF5 (76, 78, 80, 82, 84, 0.5, −2, RLS (70–90)) ([54], pp. 385–397; [8], p. 470). Here, we used the mandatory requirement thresholds of 78 and 82 dB as the baseline values because we expect the values to improve through the design process.

Scoring function output: 0 to 1 (Fig. 6)

Trace to functional requirement, ReqF5, in the Bake My Cookies use case ([8], pp. 19–21).

Owner: Engineering

Date of last change: 12/25/2020

Table 7 A requirement from the Chocolate Chip Cookie Acquisition System

Attribute	Explanation
Identification tag (ID)	ReqNF4
Name	Audible signal for cookies are ready
Text	An audible signal shall indicate when the cookies are ready. This signal shall have an intensity level of 80 ± 2 decibels at a distance of 30 cm and a frequency of 440 Hz
Priority	9
Verification method	During design and construction, this requirement will be verified by test. The proposed device will be mounted on a test bench and will be activated per test instructions. The sound power level in decibels (dB) will be measured at a distance of 30 cm. At the final test and during operation, this requirement will be verified by demonstration. An actual oven will activate the audible signal, and the sound power level in decibels (dB) will be measured at a distance of 30 cm
Verification difficulty	It will be easy to verify this requirement
Refined by technical performance measure (TPM) [46]?	No
DeriveReqt:	This requirement refines ReqF5: Cookie shall emit an audible signal when the timer has elapsed
Owner	Pat the engineer
Date of last change	January 26, 2021

A Risk from the Chocolate Chip Cookie Acquisition System

Failure event: Audible signal for “cookies are ready” is too loud.

Potential effects: Someone’s hearing could be damaged.

Relative likelihood: Noise-induced hearing loss affects 10% of Americans. So we assess this likelihood at 0.1.

Severity of consequences: 1.0

Estimated risk: 0.1

Priority: This should be described with a risk management chart.

Mitigation method: During design, the proposed device is mounted on a test bench and activated following test instructions. The sound intensity level in decibels (dB) is measured at a distance of 30 cm. The device is put under configuration control to ensure that it is not replaced or altered. This design is conservative in that the device should produce only 80 dB. Exposure to 120 dB or less for only a few seconds is unlikely to cause permanent hearing loss.

Status: Active

Trace to: ConOps

Assigned to: Pat the engineer

Date: Tracking started on April 1, 2020.

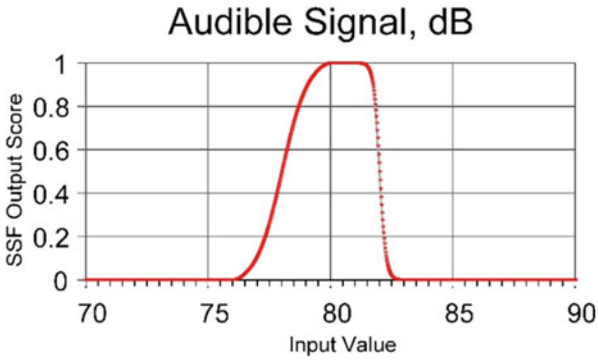


Fig. 6 The audible signal scoring function of Chocolate Chip Cookie Acquisition System

Comparing a Requirement, an Evaluation Criterion, and a Risk

Table 8 compares the three descriptions of the auditory output given above. Once again, it is not surprising that these products are similar because they come from similar processes. However, we might compare these three products and determine whether changes to the templates for each of them are warranted.

For example, the tradeoff study process requires scoring functions ([54], pp. 385–397; [8], pp. 246–258). Should the requirements process and the risk process also *require* scoring functions? Some requirements have scoring functions, but we do not want to require scoring functions for *all* requirements. We have not found risk analyses that used scoring functions.

The risk analysis process has an attribute named *status*. This attribute should change frequently as the design develops. On the other hand, requirements are not likely to come and go like risks. But it would be easy to add a column in the requirements database to cover this possibility. In a tradeoff study, it is not likely that evaluation criteria would come and go. But the whole tradeoff study could have status, and possible values would be under construction, gathering data, alternatives being evaluated, and the decision has been made.

Requirements have an attribute listing the difficulty of satisfying and verifying the requirement. Should the tradeoff study process and the risk process also have such an attribute? First, we do not think that the final tradeoff study should. However, it might be useful to state confidence in the results. This difficulty typically stems from uncertainty in the measurements and evaluation criteria and qualitatively from how different the options are. For example, is therapy better than drugs to treat depression? These are very different alternatives, and thus, it is hard to do a fair comparison. Your confidence in the result should be low. Second, one school of risk analysis has three columns in their risk tables: relative likelihood, severity of consequences, and *difficulty of detection*. We did not use difficulty of detecting the failure event because we found that it added complexity without comparable added value.

Both the requirements process and the tradeoff study process have an attribute to trace where the item came from, for example, from a particular use case or review. It

Table 8 Comparison of a requirement, an evaluation criterion, and a risk for the Chocolate Chip Cookie Acquisition System

Requirement	Evaluation criterion for a tradeoff study	Risk
Name: Audible signal indicating cookies are ready	Name of criterion: Audible signal indicating cookies are ready	Failure event: Audible signal indicating "cookies are ready" is too loud
Text: This audible signal shall have an intensity level of 80 ± 2 dB	Description: This audible signal should have an intensity level of 80 ± 2 dB	Potential effects: Someone's hearing could be damaged
Priority: 9	Weight of importance: 9	Relative likelihood: 0.1 Severity of consequences: 1.0 Estimated risk: 0.1 Priority: High
Verification method: The sound power level in decibels (dB) will be measured at a distance of 30 cm	Measurement method: Sound intensity level in decibels (dB) is measured at a distance of 30 cm Units: dB	Mitigation method: Sound intensity level in decibels (dB) is measured at a distance of 30 cm
Difficulty: Easy to satisfy and verify		
	Scoring function SSF 5 (76, 78, 80, 82, 84, 0.5, −2, RLS (70–90)	
		Status: Active
Refined by TPM? No DeriveReqt: This requirement refines Functional requirement ReqF5 in the Bake My Cookies use case	Trace to functional requirement ReqF5 in the Bake My Cookies use case ([8], pp. 19–21)	Trace to ConOps
Owner: Engineer	Owner: Pat the engineer	Assigned to: Pat the engineer
Date of last change: 4/1/20	Date of last change: 12/25/20	Date of last change: 4/1/20

might be hard to implement, but this would also be nice for the risk process. For example, if the risk factor was derived from an FMEA or industrial experience, it could be marked as such.

This section compared the requirements discovery, tradeoff studies, and risk analyses processes. It did this by comparing the activities and products of these processes and by comparing example requirements, evaluation criteria, and risks.

The SIMILAR Process

The SIMILAR process of Fig. 7 is based on Bahill and Gissing [7]. It comprises seven key activities: **S**tate the problem, **I**nvestigate alternatives, **M**odel the system, **I**ntegrate, **L**aunch the system, **A**ssess performance, and **R**eevaluate. These seven activities are conveniently summarized using the acronym SIMILAR. We use this process to

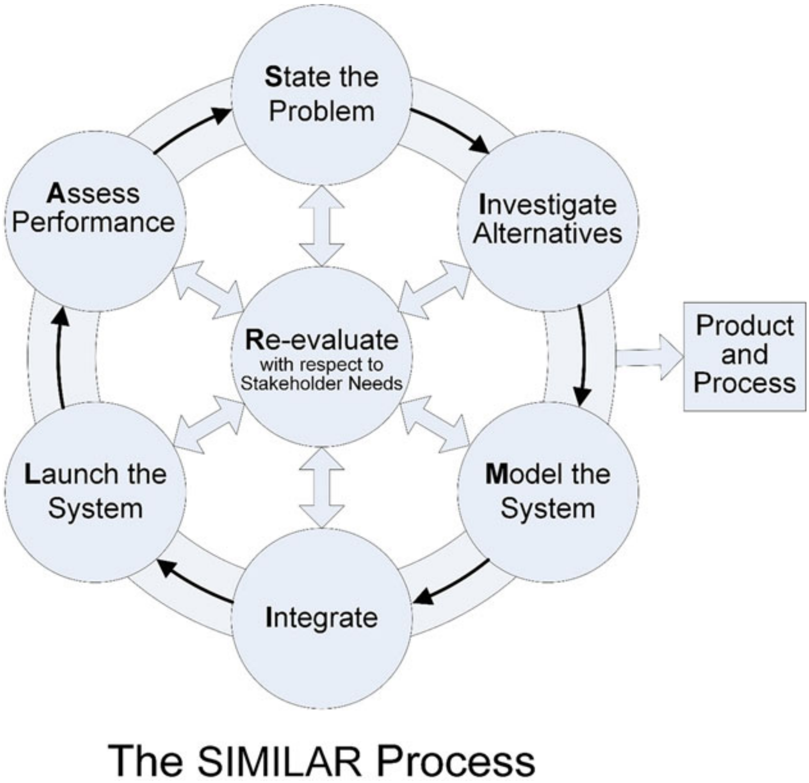


Fig. 7 The SIMILAR process. (Based on Bahill and Gissing [7])

provide the overall context for problem-solving during system design and every other human activity. At the outset, we want to clarify that the activities in the SIMILAR process are performed iteratively and in parallel with many unshown feedback loops. Each activity in the SIMILAR process is described next.

State the Problem

> "The beginning is the most important part of the work" Plato, *The Republic*, 4th century BC.
> "Begin at the beginning," the King said gravely, "and go on 'til you come to the end; then stop." From Lewis Carroll, *Alice's Adventures in Wonderland*.

The problem statement contains many tasks that are performed iteratively, many of which can be performed in parallel. We examined Figs. 2, 4, and 5 and studied Tables 5 and 8 and found the following listed tasks that fit into the problem statement activity.

- Understanding customer needs is the first and foremost task.
- Identify stakeholders such as end users, operators, maintainers, suppliers, acquirers, owners, customers, bill payers, regulatory agencies, affected individuals or organizations, victims, sponsors, manufacturers, etc. Our stories that explain how the system will work are a source for identifying stakeholders. Stories have villains. Some villains that the system designer might encounter are competitors, the IRS, the EPA, Mexican drug cartels, and their associated gangs like MS-13.
- Where do the inputs come from? Requirements come mainly from the use cases and the stakeholders. They are presented by the customer and the systems engineer. Evaluation criteria and proposed alternatives for tradeoff studies come from use cases, meetings, and reviews and are presented by the design engineer. Risk events are identified in use cases, brainstorming, meetings, and reviews and are described by the risk analyst (Fig. 8).

- Describe how the system works using stories and use case models. The use case models provide requirements and test cases.
- State the problem in terms of *what* needs to be done, not *how* it must be done. The problem statement may be in prose form or the form of a model.
- Develop the incipient architecture.
- Define the scope of the project. This shows the boundary between what is inside the system and the external world.
- Initiate risk analysis. Yes, the risk analysis of the system should begin at the same time as the requirements discovery and tradeoff study processes.

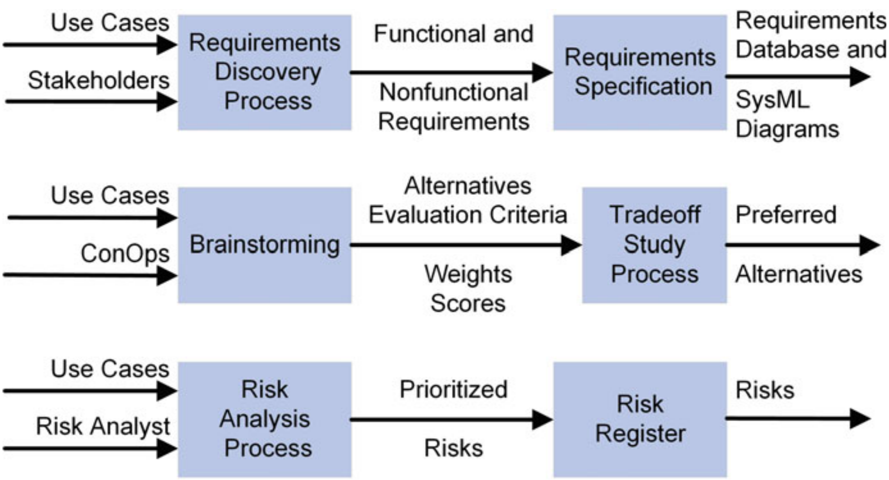


Fig. 8 Inputs and outputs

An interesting aside: By law, with minor room for excursions, the role of a US CEO has historically been defined as maximizing long-term (sic!) shareholder return. Recently, there has been a movement to redefine this requirement as "benefitting owners, employees, customers, communities, society, etc." This is going to pose a huge tradeoff problem (albeit perhaps an appropriate one) for CEOs and boards! It will at least assure lifetime employment for lawyers (Norm Augustine, personal communication).

Investigate Alternatives

We examined Figs. 2, 4, and 5 and studied Tables 5 and 8 and found the following listed tasks that fit into the Investigate Alternatives activity.

- One should investigate alternative requirements, designs, and risk events using evaluation criteria such as performance, cost, schedule, and risk.
- For quantitative analyses, identify attributes of requirements, evaluation criteria for tradeoff studies, and the likelihood of occurrence and severity of consequences for risk events. Assign them weights of importance to show priorities.
- Scoring (utility) functions are mandatory for tradeoff studies but are optional for requirements and risks.
- Select methods for combining the data. State the combining function that will be used. Usually, this will be the Boolean AND function for requirements, the sum of weighted products for tradeoff studies, and a chart or a matrix for risks.
- Finally, one must collect evaluation data and use it to assign values to attributes for requirements, weights and scores for tradeoff studies, and likelihoods and severities for risk analyses.

Model the System

We examined Figs. 2, 4, and 5 and studied Tables 5 and 8 and found the following listed tasks that fit into the Model the System activity.

- Models are typically created for most requirements, alternative designs, and risk events. These models are *consistently elaborated* ([54], pp. 178–180) (that is, expanded) throughout the system life cycle. A variety of models can be used.
- Requirements can be modeled with use case models, textual shall statements, tables, spreadsheets, and specialized databases. Friedenthal et al. [25] and Madni and Sievers [41] model requirements with Use Case Diagrams (uc), Requirements Diagrams (req), Sequence Diagrams (sd), Activity Diagrams (act), Block Definition Diagrams (bdd), and Package Diagrams (pkg). Subsequently, the requirements must be clarified, decomposed, allocated, and derived.
- Tradeoff studies are usually modeled with tradeoff matrices implemented with spreadsheets. The alternative designs within them are modeled with UML

diagrams, SysML diagrams, analytic equations, computer simulations, and mental models.

- Risks are modeled with tables containing values for the likelihood of occurrence and severity of consequences and figures displaying these data.
- Everything must be prioritized. The requirements set should be prioritized to find the most important requirements. For tradeoff studies, the preferred alternatives are identified with a tradeoff matrix. The ranges for likelihood and severity are adjusted for risk events to find the greatest risks.
- The results of a sensitivity analysis can be used to validate a model, flag unrealistic model behavior, point out important assumptions, help formulate model structure, simplify a model, suggest new experiments, guide future data collection efforts, suggest accuracy for calculating parameters, adjust numerical values of parameters, choose an operating point, allocate resources, detect critical evaluation criteria, suggest tolerance for manufacturing parts, and most importantly identify cost drivers.

Integrate

We examined Figs. 2, 4, and 5 and studied Tables 5 and 8 and found the following list of tasks that fit into the Integrate activity.

- Integration means bringing elements together so that they work as a whole to accomplish their intended purpose and deliver value. (A new systems engineering buzzword is emergent behavior. It suggests that, in terms of behavior, the result might be greater than the sum of its parts.) Specifically, systems, enterprises, and people need to be integrated to achieve desired outcomes. To this end, interfaces need to be designed between subsystems. Subsystems are typically defined along natural boundaries in a manner that minimizes the amount of information exchanged between the subsystems. Feedback loops between individual subsystems are easier to manage than feedback loops involving densely interconnected subsystems.
- Evaluation criteria should trace to requirements. Risks should trace to requirements or particular meetings or reviews. Requirements should refine higher-level requirements and should link to risks. Requirements and risks might be refined by technical performance measures (TPMs) [46]. TPMs are evaluated continually during the design process as a way of detecting and mitigating risk.

Launch the System

We examined Figs. 2, 4, and 5 and studied Tables 5 and 8 and found the following listed tasks that fit into the Launch the System activity.

- Launching the System means either deploying and running the actual system in the operational environment or exercising the model in a simulated environment to produce necessary outputs for evaluation. In a manufacturing environment, this might mean buying commercial off-the-shelf hardware and software, writing code, and/or bending metal. The purpose of system launch is to provide an environment that allows the system or its model to do what it is being designed to do.
- The outputs of these processes are a requirements specification, preferred alternatives, and the risk register. One should continually monitor the requirements (in the requirements database), alternative designs (in the process assets library, PAL), and risks (in the risk register) looking for possible changes and bring these to the attention of the decision-makers. One should continually monitor the marketplace looking for new requirements, products, designs, and risks and bring these to the attention of the decision-makers.

Assess Performance

We examined Figs. 2, 4, and 5 and studied Tables 5 and 8 and found the following listed tasks that fit into the Assess performance activity.

- Test, validation, and verification are important tasks for all processes.
- There should be regularly scheduled and performance-initiated expert reviews. The results of these reviews are presented to the decision-maker (DM) and are put in the process assets library (PAL).
- Evaluation criteria, measures, metrics, and TPMs are all used to quantify system performance. Evaluation criteria are used in requirements discovery, tradeoff studies, and risk analyses. Measures and metrics are used to help manage a company's processes. TPMs are used to mitigate risk during design and manufacturing.

Reevaluate

The distinction between an engineer and a mathematician is arguably the use of feedback in design. For two and a half centuries, engineers have used feedback to control systems and improve performance. It is one of the most fundamental engineering concepts. Reevaluation is a continual feedback process with multiple parallel loops. Reevaluation means observing outputs and using this information to modify the inputs, the system, the product, and/or the process.

The SIMILAR process (Fig. 7) shows the distributed nature of the reevaluate function in the feedback loops. However, it is important to realize that not all loops will always come into play all of the time. The loops that are used depend on the problem to be solved and the problem context. Reevaluation includes formal inspections, expert reviews, and reviews with the customer.

A very important and often neglected task in any process is monitoring and improving the process itself. This self-improvement process is shown explicitly in Figs. 2, 4, and 5 with the following tasks: Monitor and improve the requirements process, monitor and improve the tradeoff study process, and monitor and improve the risk process. These processes use a different timescale than the mainline processes. For example, the monitor and improve the X process tasks run with timescales of months to years, whereas the mainline processes, like revise with the customer and prioritize X, run with timescales of days to weeks.

This section presented the SIMILAR process that was developed by Bahill and Gissing [7]. It has served as a general model for doing everything.

The Overarching Process

We used Figs. 2, 4, 5, and 7 and Tables 5 and 8 and created the Overarching Process of Fig. 9 that can be used for system design, requirements discovery, decision analysis and resolution, tradeoff studies, and risk analysis. The processes in these figures were created many years before the Overarching Process was conceived. This Overarching Process is a superset of the three processes: requirements, tradeoffs, and risks. Viewed another way, each of the three processes is a tailoring of the Overarching Process [32]. A new feature in the Overarching Process is identifying and handling uncertainty. In Fig. 9, we have marked with a ▼ those activities that have the most uncertainty.

It is important to note that this is not a waterfall process. One activity does not have to wait for the previous activity to end before it can start. Also, it is an iterative process with a multitude of unshown feedback loops.

In the Investigate Alternatives block, it looks like discover requirements and describe alternative solutions are performed in parallel. Well, some of them are. However, this whole figure is very iterative. The systems engineer gets a few requirements and then gets a few alternatives and criteria. Then he gets a few more requirements and a few more alternatives and criteria, etc.

Some tasks listed in Figs. 2, 3, and 4 might seem to be missing in the Overarching Process of Fig. 9. However, these tasks have been subsumed in the activities shown. For example, the task of "examining the shape of the data" is included in the "choose combining function activity." The task of studying the do-nothing alternative is in the Create Tradeoff Matrix and Evaluate Alternatives activity. The identify important parameters task is in the perform a sensitivity analysis activity. The Develop Incipient Architecture task is included in the Describe Alternative Activities activity. Finally, the task to track outliers (both high frequency but low severity and low frequency but high severity) is in the Monitor and Manage risks activity.

Sometimes key stakeholders impose system-level constraints, requirements, goals, etc. The Overarching Process can include collecting and evaluating these before getting started with stating the problem. That is, we can have work before

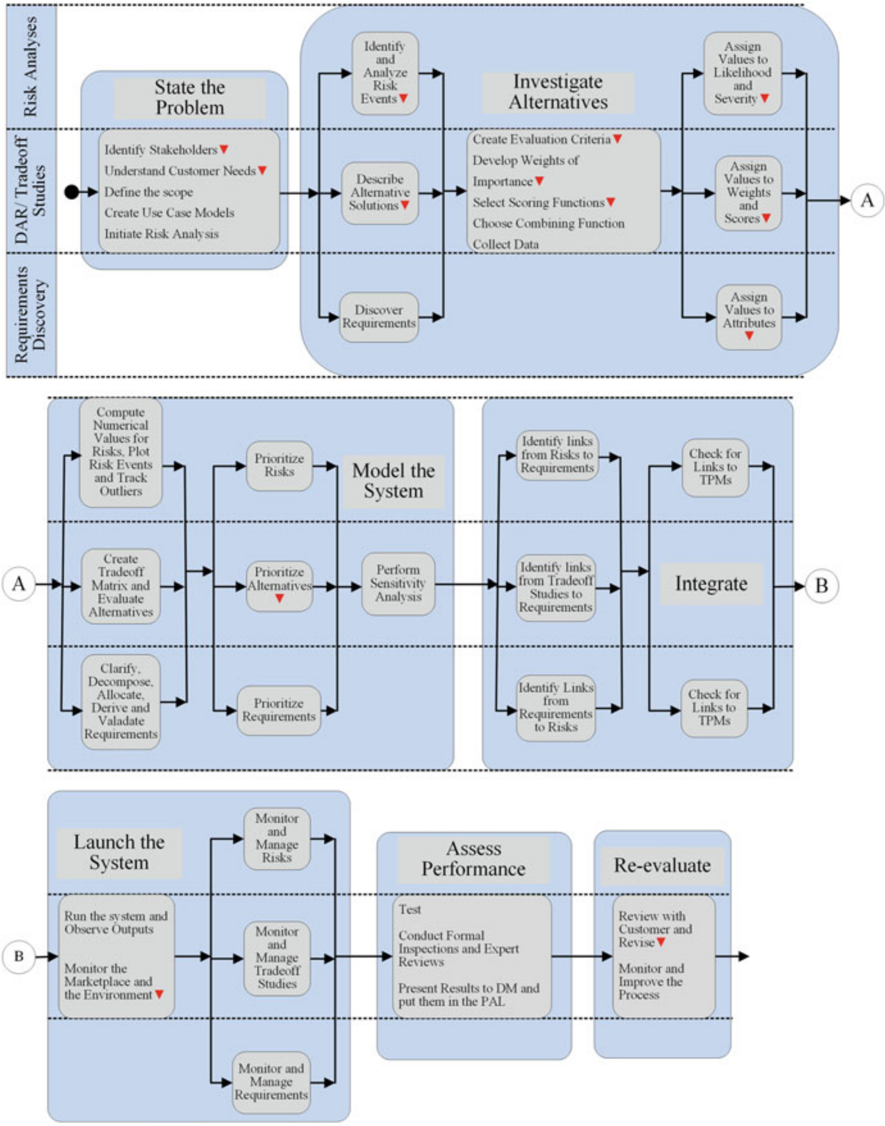


Fig. 9 The Overarching Process. DAR is decision analysis and resolution. DM is the decision-maker, PAL is the process assets library, and TPM is a technical performance measure

Fig. 9 the goal of which is to establish feasibility. When no solution is evident, then there must be iterations that add or eliminate the top-level desires so that the option space is increased. When there is no set of options agreeable to the primary stakeholders, then we do not continue to the next phase (personal communication Mike Sievers, 2021).

Effects of Human Decision-Making on the Overarching Process

The study of human decision-making reveals that the presence of cognitive biases can never be ruled out [43, 52]. This is also the contention of the economic school of heuristics and biases, which produced Prospect Theory [29], a theory that describes how people respond to choices under risk and uncertainty. Innate human biases, and external circumstances, such as the framing or the context of a question, can compromise decisions. It is important to note that subjects maintain a strong sense that they are acting rationally even when they are exhibiting these biases [28].

Other chapters in this handbook do not show MBSE processes, diagrams, or viewpoints that help ameliorate human biases in decision-making. So this section is unique in this handbook.

In Fig. 9, we have marked with a ▼ the actions that are the biggest contributors to uncertainty. They all deal with human decision-making rather than uncertainty in the weather, climate, solar variability, geology, political actions, or experimental data. In the upcoming paragraphs, we will explain how the activities of Fig. 9 are affected by uncertainty. Most reasons involve confirmation bias, severity amplifiers, and framing. Therefore, we will first discuss these three decision modifiers.

Confirmation Bias

Arguably, the most important cause of fallibility in human decision-making is confirmation bias. Humans hear what they want to hear and reject what they do not want to hear. Humans filter out information that contradicts their preconceived notions and remember things that reinforce their beliefs. Confirmation bias causes decision-makers to actively seek out and assign more weight to evidence that confirms their hypotheses and ignore or under weigh the evidence that could disconfirm their hypotheses. For example, mothers emphasize the good deeds of their children and de-emphasize their bad deeds. This is why we often hear the mother of a terrorist crying out, “My boy is innocent. He could never have killed all those people.” People who think that they have perfect memory and perfect recall tend to ignore instances when they forgot something and tend to secure in long-term memory instances when they correctly recalled events and facts. Senior citizens often believe that they are good drivers despite tests that show that they have poor vision, fading cognitive processes, and slow reflexes. Thirty years ago, most cigarette smokers were in denial about the hazards of smoking. Some people say, “There must be a storm coming because my arthritic joints are hurting.”

Social media is making this worse. Not only do you filter what you see and hear, but also Facebook filters what you are exposed to. They present to you things from the friends you care about. These friends are probably ideologically like you, which accentuates the filtering process.

Nickerson [44] reported many common instances of confirmation bias. In one, the subjects were given a triplet such as (2, 4, 6) and were asked to guess the rule that was used to generate the triplet and then try to prove or disprove that rule by giving examples. After each guess, they were told if they were right or wrong. For example, if the subject's mental model for the rule was "successive even numbers," they might guess (10, 12, 14) or (20, 22, 24), triplets that would confirm their mental model, but they would seldom guess (1, 3, 5) or (2, 4, 8), triplets that might disprove their mental model. He also presented another example of confirmation bias – witches.

The execution of 40,000 suspected witches in seventeenth century England is a particularly horrific case of confirmation bias functioning in an extreme way at the societal level. From the perspective of the people of the time, belief in witchcraft was perfectly natural, and sorcery was widely viewed as the reason for all ills and troubles that could not otherwise be explained. In one test of a woman being a witch, the mob tied the suspect to a chair and threw her into a river. If she floated, it was proof that she was a witch, and she was executed. If she sank, well, too bad.

Until the nineteenth century, physicians often did more harm than good because of confirmation bias. Virtually anything that could be dreamed up for the treatment of disease was tried and, once tried, lasted decades or even centuries before being given up. It was, in retrospect, the most frivolous and irresponsible kind of human experimentation. They used bloodletting, purging, infusions of plant extracts and solutions of metals, and every conceivable diet including total fasting. Most of these were based on no scientific evidence. How could such ineffective measures continue for decades or centuries without their ineffectiveness being discovered? Probably, because sometimes patients got better when they were treated, sometimes they did not, and sometimes they got better when they were not treated at all. Peoples' beliefs about the efficacy of specific treatments seem to have been influenced more strongly by those instances in which treatment was followed by recovery than by those instances in which there was no recovery. A tendency to focus on positive cases could explain why the discovery that diseases have a natural history and people often recover from them with or without treatment was not made until much later.

Most people react to news articles with confirmation bias. If a left-wing liberal reads a news story about a scientific study that showed how effective it was to give money to poor people, he might think, "That's an insightful article. I'll remember it." However, if one of those same people reads about a new study showing that giving people money when they are unemployed just makes their lives worse, then he might start looking for flaws in the study. If a person has a long-felt belief that the income gap between the rich and the poor in America is too large and is growing too fast, then a new study that challenges this belief might be met with hostility and resistance. However, if that person readily accepts a study that confirms his belief, then that is confirmation bias.

Before a person participates in an activity that involves evaluating requirements, alternatives, evaluation criteria, weights, scores, or risks, they should be reminded about confirmation bias. During the evaluation process, people should be on the lookout for instances of confirmation bias exhibited by other people and politely suggest that it might be influencing their evaluations.

Most people do not think like scientists: They think like lawyers. They form an opinion and then emphasize only evidence that backs up that opinion.

In filling out the tradeoff study matrix for the Cookie Acquisition System of Table 3, a physical fitness pundit might want to add more (possibly dependent) subcriteria such as protein, fiber, antioxidants, and unsaturated fats to the nutrition evaluation criterion. The system engineer must explain to the team why some people might want more subcriteria and the effects of adding dependent subcriteria.

Severity Amplifiers

Interpersonal variability in evaluating the seriousness of a situation depends on the circumstances surrounding the event. An evaluation may depend on factors such as how the criterion affects that person, whether that person voluntarily exposed himself to the risk, how well that person understands the alternative technologies, and the severity of the results. The following are severity amplifiers: lack of control, lack of choice, lack of trust, lack of warning, lack of understanding, being man-made, newness, dreadfulness, fear, personalization, ego, recallability, availability, representativeness, vividness, uncertainty, and immediacy.

The following paragraphs explain some severity amplifiers. *Lack of control*: A man may be less afraid of driving his car up a steep mountain road at 55 mph than having an autonomous vehicle drive him to school at 35 mph. *Lack of choice*: We are more afraid of risks that are imposed on us than those we take by choice. *Lack of trust*: We are less afraid while listening to the head of the Centers for Disease Control explaining anthrax than while listening to a politician explain it. *Lack of warning*: People dread earthquakes more than hurricanes because hurricanes give days of warning. People in California follow strict earthquake regulations in new construction. People in New Orleans seem to ignore the possibility of hurricanes. *Lack of understanding*: We are more afraid of ionizing radiation from a nuclear reactor than of infrared radiation from the sun. In the 1980s, engineers invented nuclear magnetic resonance imaging (NMRI). When the medical community adopted it, they renamed it magnetic resonance imaging (MRI). They dropped the adjective *nuclear* to make it sound friendlier. *Man-made*: We are more afraid of nuclear power accidents than solar radiation.

Newness: We are more afraid when a new disease (e.g., swine flu, SARS, MERS, Ebola, Zika, and COVID-19) first shows up in our area than after it has been around a few years. *Dreadfulness*: We are more afraid of dying in an airplane crash than of dying from heart disease. *Fear:* If a friend tells you that a six-foot rattlesnake struck at him, how long do you think the snake was? We suspect 3 ft. But of course, the length of the snake is irrelevant to the harm it could cause. It is only related to the fear it might induce. *Personalization*: A risk threatening us is worse than that same risk threatening you. *Ego:* A risk threatening our reputations is more serious than one threatening the environment.

Recallability: If something can be readily *recalled*, it must be more important than alternatives that are not as readily recalled. We are more afraid of cancer if a friend has recently died of cancer. We are more afraid of traffic accidents if we have just observed one. Recallability is often called *availability.* Something readily *available* to the mind must be more important than alternatives that are not as readily available. *Representativeness:* The degree to which an event is similar in essential characteristics to its parent population increases its importance. In the dice game of craps, rolling a seven would be typical of random rolls, and therefore, it would be representative of the parent population and would therefore be important. *Vividness of description*: An Edgar Allen Poe story read by Vincent Price will be scarier, than one that either of us reads to you. *Ambiguity or uncertainty:* Most people would rather hear their ophthalmologist say, "You have a detached retina. We will operate tonight" than "You might have a detaching vitreous, or it could be a detaching retina, or maybe its cancer. We will do some tests and let you know the results in a week." *Immediacy*: A famous astrophysicist was explaining a model for the life cycle of the universe. He said, "In a billion years, our sun will run out of fuel, and the earth will become a frozen rock." A man who was slightly dozing awoke suddenly, jumped up, and excitedly exclaimed, "What did you just say?" The astrophysicist repeated, "In a billion years, our sun will run out of fuel, and the earth will become a frozen rock." With a sigh of relief, the disturbed man said, "Oh, thank God. I thought you said in a *million* years."

In filling out the tradeoff study matrix for the Cookie Acquisition System of Table 3, *personalization* was an important severity amplifier. It created variability in the responses of the team making the evaluations. For example, a diabetic or someone on a strict diet will certainly want to give a much higher weight to the nutrition evaluation criterion and will give different weights to the nutrition subcriteria. Similarly, a dietitian or a nutritionist will also give different weights to the nutrition subcriteria. When these large variabilities occur, the systems engineer should explain possible causes to the team.

Framing

In the human decision-making community, *utility* is a subjective measure of happiness, satisfaction, or reward a person gains (or loses) from receiving a good or service. Utility is considered not in an absolute sense (from zero), but subjectively from a reference point, established by the decision-maker's (DM) perspective and wealth before the decision, which is his frame of reference [28]. (Kahneman and Tversky's [29] utility functions show a human's subjective utility as a function of its objective value ([8], pp. 167–176). Economists use utility functions to show consumer preference of one product over another and assign a numerical value to that preference. Systems engineers use utility functions in tradeoff studies to relate different evaluation criteria that use different units of measure. Despite the incompatibility of the measures, these diverse evaluation criteria (measures of effectiveness) may nevertheless contribute to the same overall goal. Utility functions therefore convert the different input evaluation criteria (physical characteristics) into output quantities (called utility values) which are mutually and completely compatible. Wymore's scoring functions ([8], pp. 246–257) are eloquent mathematical elaborations of such utility functions.) Framing (the context of a question) could affect his decision. The section on severity amplifiers stated that interpersonal variability in evaluating the seriousness of a situation depends on framing. That is, the circumstances surrounding the event will affect how a DM responds to it. An evaluation may depend on factors such as how the criterion affects that DM, whether that DM voluntarily exposed himself to the risk, how well that DM understands the alternative technologies, and the severity of the results. In the previous section, we gave over a dozen severity amplifiers that would affect the framing of a problem.

In contrast to defining framing in passing, as we have done so far, we will now explain *framing* directly, based on Beach and Connolly [12]. The DM has a vision, a mission, values, morals, ethics, beliefs, evaluation criteria, and standards for how things should be and how people ought to behave. Collectively, these are called *principles.* They are what the DM, the group, or the organization stands for. They limit the goals that are worthy of pursuing and acceptable ways of pursuing these goals. These principles are difficult to articulate, but they powerfully influence the DM's behavior. They are the foundation of the DM's decisions and goals; actions that contradict them will be unacceptable. The utility of the outcomes of decisions derives from the degree to which these decisions conform to and enhance the DM's preconceived principles.

Goals are what the DM wants to accomplish. The goals are dictated by the principles, the problem, the problem statement, opportunities, desires, competitive issues, or gaps encountered in the environment. Goals might seed more principles. Goals should be SMART: specific, measurable, achievable, realistic, and time-bound.

The DM has *plans* for implementing the goals. Each goal has an accompanying plan. Each plan has two aspects: (1) Tactics are the concrete behavioral aspects that deal with local environmental conditions, and (2) forecasts are the anticipation of the future that provides a scenario for forecasting what might result if the tactics are successful. The plans for the various goals must be coordinated so that they do not interfere with each other and so that the DM can maintain an orderly pursuit of the goals. The plans are also fed back to the principles; therefore, they might foment more principles.

Framing means embedding observed events into a context that gives them meaning. Events do not occur in isolation; the DM usually has an idea about what led up to them. This knowledge supplies the context, the ongoing story that gives coherence to experiences, without which things would appear random and unrelated. A frame consists of the principles, goals, and plans that are deemed relevant to the decision at hand and that fixes the set of principles that influence that decision.

The DM uses contextual information to probe his or her memory. If the probe locates a contextual memory that has similar features to the current context, then the current context is said to be recognized. *Recognition* defines which principles, goals, and plans are relevant to the current context and provides information about the goals and plans that were previously pursued in this context. If a similar goal is being pursued this time, then the plan that was used before may be used again.

In summary, framing means describing all aspects of the problem, the problem statement, and the DM's mind that will affect decisions.

In filling out the tradeoff study matrix for the Cookie Acquisition System of Table 3, *framing* was important. The student's frame of mind includes his or her present grade in the class and the scheduled occurrence or not of an exam the next day. These will affect his or her weights and scores for the lost study time evaluation criterion. The tradeoff study team must be aware of their teammates' frames of mind.

Because of these human mental mistakes, and many more [14, 52], weights and scores in tradeoff studies and other values that depend on human judgments are subjective and have large variations. This is the reason for performing sensitivity analyses: to identify simple judgments that have a large effect on the outcome.

The Overarching Process

Figure 9 shows a diagram for the Overarching Process. We have marked with a ▼ those activities that are the biggest contributors to uncertainty. We will now examine these activities. Specifically, we identify human psychological factors that can adversely influence human decision-making when dealing with uncertainty. Here, we only give short phrases listing these factors. Three are described in detail above. The others are explained in detail in Smith et al. [52] and Bohlman and Bahill [14].

- State the problem. This activity tends to be affected by severity amplifiers and framing. Additionally, it is affected by incorrect phrasing, attribute substitution, political correctness, and feeling invincible.
- Identify stakeholders. This activity is affected by framing.
- Understand customer needs. This activity is affected by confirmation bias, severity amplifiers, and framing.
- Define the scope, which is given in a high-level use case.
- Create use case models.
- Initiate risk analysis.
- Investigate alternative solutions. This activity is affected by confirmation bias, severity amplifiers, and framing.
- Identify and analyze risk events. This activity is affected by confirmation bias and severity amplifiers.
- Create evaluation criteria. This activity is affected by severity amplifiers. Additionally, it is affected by dependent evaluation criteria, relying on personal experience, the Forer Effect, and attribute substitution.
- Develop weights of importance. This activity tends to be affected by severity amplifiers. Additionally, it can be affected by whether the weights are the result of choice or calculation.
- Select scoring functions. Mistakes here include mixing gains and losses, not using scoring functions, and anchoring. The biggest mistake is stating output scores with false precision.
- Choose combining functions. Lack of knowledge is the key problem in this activity. There are several appropriate combining functions. One of the oldest and most studied means for combining data under uncertainty is the certainty

factor calculus employed by the Mycin expert system at Stanford University in the 1980s [16]. It is now called the sum combining function.

- Assign values to (1) attributes, (2) weights and scores, and (3) likelihood and severity. All three of these activities can be adversely affected by confirmation bias, severity amplifiers, relying on personal experience, magnitude and reliability, and judging probabilities poorly. In addition to these human decision-making errors, we also have metrology errors. No measurement is exact: You can always do better (that is, until we get to subatomic particles). Therefore, you decide how much uncertainty you will allow in your measurements and budget appropriately.
  - Our understanding of the laws of physics is not accurate. This precludes precise models for nature. Likewise, our models for dark energy and dark matter are inaccurate. But these physics problems do not manifest in most of our design problems.
  - If models are being used to compute values for risks, evaluation criteria, etc., then statistical measures like mean, standard deviation, and correlation coefficients can be used to identify the range of expected values. For some problems, for example, calculating the time for an asteroid to impact the earth, the best we can do is give uncertainty ranges.
- Prioritize alternatives. This activity is affected by confirmation bias, severity amplifiers, and framing. This activity can be degraded by serial consideration of alternatives, isolated or juxtaposed alternatives, conflicting evaluation criteria, adding alternatives, maintaining the status quo, and uneven level of detail. The order in which the alternatives are listed has a big effect on the values that humans give for the evaluation data. Therefore, a tradeoff study matrix should be filled out row by row with the status quo being the alternative in the first column. This makes the evaluation data for the status quo the anchors needed for estimating the evaluation data for the other alternatives. This is a good choice because the anchoring alternative is known and is consistent, and you have control over it. Prioritization also depends on the algorithm being used to combine the data.
- Perform sensitivity analyses. Done right, there should be no problems. Otherwise, lack of training and the Hawthorne effect can potentially confound the study.
- Monitor the marketplace and the environment. This activity is typically affected by severity amplifiers. Additionally, tunnel vision can throw off the analysis. Therefore, to avoid tunnel vision, the environment must be a part of the framing.
- Conduct formal inspections [24] and expert reviews. These inspections and reviews are done entirely by humans. Therefore, every human limitation such as cognitive biases, misconceptions, and preconceptions must be addressed.
- Review with stakeholders and revise. The most common mistake in design projects is failing to engage stakeholders and consult with experts in universities and local industries [14]. It is imperative to engage all stakeholders, especially in upfront engineering, to avoid the likelihood of extraneous design iterations and rework.
- The out arrow at the lower right feeds back to all of the boxes in Fig. 9.

Most of these areas of uncertainty involved human decision-making, and our models for this are imprecise. Overall, the biggest cause of uncertainty is simply that we cannot predict the future. A total reversal of the Earth's magnetic field is imminent, but we cannot predict when it will occur.

Uncertainty in Stating the Problem for the Overarching Process

Now that we have *identified* sources of uncertainty in the Overarching Process, we will present examples of some techniques for *handling* uncertainty in the tradeoff study process. The first and most important step in performing a tradeoff study is stating the problem. Uncertainty can cause mistakes in the problem statement. This section is based on Diogenes [1]. These are some of the tasks that were described in the state the problem paragraph of the SIMILAR process section of this chapter:

- Using stories and use cases, explain what the system is supposed to do.
- Understand customer needs.
- Identify stakeholders.
- Discover the inputs and their sources.

At the beginning of any system design, we do not know exactly what the finished product will do. Functions, requirements, and desirements may have been stated, but incomplete understanding, mistakes, unknown technology, and improvement opportunities usually change the preconceived functioning of any system. To understand and explain what the system is supposed to do and how it works, we use a multitude of stories and use case models. An example of a use case model for handling uncertainty is coming up shortly.

It turns out that all of these activities involve human decision-making. Therefore, most of the mistakes caused by uncertainty will be found in the system models and documentation.

Understanding customer needs, identifying stakeholders, and discovering the system inputs are all affected by uncertainty, confirmation bias, severity amplifiers, framing, and many mental mistakes [52].

The primary reason that these mental mistakes are so important is that people do not realize that they exist. And the people that know of their existence believe that these mistakes do not affect *their* decision-making. However, when the results of these mistakes are pointed out, most people are willing to rewrite to eliminate their undesirable effects. So the best way to get rid of such mistakes is to bring them out in the open.

A System for Handling Uncertainty in Models and Documentation

We will now present our process for ameliorating such mental mistakes. To handle uncertainty, all of the work products must be available for public view, must be subjected to formal reviews, and must be approved in expert reviews, and all of these activities must be in a feedback control loop with frequent small iterations.

The first step in our process is to prepare a document that explains confirmation bias, severity amplifiers, and framing, as well as the mental mistakes of poor problem stating, incorrect phrasing, attribute substitution, political correctness, and feeling invincible [52]. Portions of this chapter could serve this purpose. All people involved in the process must read this document in advance. We have been using this approach for over a decade [1], but it may seem new to the systems engineering community. The following use case is our process for identifying and ameliorating mistakes in a system design.

The system described by this use case can be used to *handle* uncertainty in the models and documentation of the Overarching Process. It is an abstract that included use case.

Name: Perform Formal Inspection to Find Mistakes Caused by Uncertainty

Iteration: 4.6

Derived from the concept exploration document, Diogenes [1]

Brief description: A formal inspection is a structured group review process used to find defects and mistakes in requirements, programming code, test plans, models, and designs [24]. The flow of this use case is called from the handling uncertainty use case, which is not described in this chapter. When this sub-flow ends, the use case instance continues from where this included use case was called.

Level: Medium

Priority: Medium

Scope: The Inspection Team, the work products to be inspected, and the process assets library (PAL).

Added value: The company will be able to look for unresolved uncertainties, mental mistakes, risks, opportunities for Built-in Self-Test (BiST), and unintended consequences of the system being designed all at the same time. This should increase efficiency. Furthermore, discovering *positive* unintended consequences could provide additional revenue.

Goal: Find defects caused by uncertainty and mental mistakes. Find unidentified risks, opportunities for BiST, and unintended consequences of the system being designed.

Primary actors: The Inspection Team is comprised of the moderator, systems engineer, author/designer, reader, recorder, and additional inspectors

The **moderator** leads the inspection, schedules meetings, distributes inspection materials, controls the meetings, reports inspection results, and follows up on rework issues. Moderators should be trained in how to conduct inspections. Risk or quality assurance managers often serve in this role.

The s**ystems engineer** coordinates the inspection with the overall design process. The systems engineer delivers the lists of unresolved uncertainties, mistakes, risks, opportunities for BiST, and unintended consequences to risk management, test engineering, marketing, management, and legal. He or she also puts these lists in the project PAL.

The **author/designer c**reates and/or maintains the work products being inspected. The author/designer answers questions asked about the work products during the inspection, looks for defects, and fixes defects. The author/designer, or other members of the design team, cannot serve as moderator, reader, or recorder.

During the meeting, the **reader** leads the Inspection Team through the work products being inspected, interprets sections of the artifact by paraphrase, and highlights important parts**.**

The **recorder** classifies and records unresolved uncertainties, mental mistakes, risks, opportunities for BiST, unintended consequences of the system being designed, and issues raised during the inspection. The moderator might perform this role in a small Inspection Team.

The **inspector** attempts to find errors in the work products. This role is filled by several people. All participants act as inspectors, in addition to any other responsibilities. The following may make good inspectors: the person who wrote the specification for the work products being inspected; the people responsible for implementing, testing, or maintaining the work product; a quality assurance representative; a representative of the user community; and someone who is not involved in the project but has infinite experience and impeccable wisdom.

Secondary actors: The process assets library (PAL)

Frequency: Once a month or before specified reviews

Precondition: An author/designer has requested an inspection of his work product.

Trigger: This use case will be included from the handling uncertainty use case.

Main Success Scenario:

1. **Planning activity:** The moderator selects the Inspection Team, obtains the problem statement and the work products to be inspected from the author/designer, and distributes them along with other relevant documents to the Inspection Team. As a rule of thumb, the work products to be inspected typically comprises 200 lines of code or 2000 lines of text.
2. **Overview meeting:** The moderator explains the inspection process to the Inspection Team. This will take from 10 min to 3 h depending on the backgrounds of the team members. The author/designer may describe the key features of the work products.
3. **Preparation:** Each member of the Inspection Team examines the work products before the actual inspection meeting. Each member should be looking for unresolved uncertainties, mental mistakes, risks, opportunities for BiST, and unintended consequences of the system being designed. (Often, risks are handled by a separate department isolated from design. But there is no reason why it has to be this way. And it may be more efficient to include it with these other activities.) Typically, this will take 2 h for each member. The amount of time each person spends will be recorded. This time would be substantially increased for an inspector running models and simulations to verify the system.
4. **Inspection meeting:** The moderator and reader lead the team through the work products. The issues are brought up one by one, and each one is discussed in a

round-robin fashion where each member comments on each issue. (Although time constraints may try to prevent this, the moderator should ensure that each participant says something. This promotes the sense of inclusion and most importantly is the best mechanism for discovering unknown unknowns.) During the discussion, all inspectors can report unresolved uncertainties, mental mistakes, risks, opportunities for BiST, and unintended consequences of the system being designed, all of which are documented by the recorder. The meeting should last no more than 2 h.

How can we prime our inspectors to look for unresolved uncertainties and unintended consequences?

If they are looking at an activity, action, process, procedure, or another verb phrase (an active verb followed by a measurable noun), then tell them to ask, "What problems could this activity create for other systems?" "How could doing this activity hurt other systems?" "If this activity failed, how could that hurt other systems?"

If they are looking at an object, component, model, or another noun phrase, then tell them to ask, "How could this object hurt other systems?" "How could this object fail?" For each failure event, ask, "How could this failure event hurt other systems?"

If they are looking at a risk, then tell them to ask, "How could this failure event hurt other systems?"

If they are looking at a use case scenario or other sequences of events, then tell them to ask, "What-if?" For example, when the document states, "The user does this and the system does that." Ask, "What if it doesn't?"

Inspectors should look for common mental mistakes that people make [52], particularly for attribute substitution, which is the most common mental mistake [51].

Inspectors should look to see if the designers used fundamental principles of good design [5], including design for resiliency [39, 45].

But we really want the mindset of looking for unresolved uncertainties and unintended consequences to become a part of company culture.

5. **Databases:** The team creates and maintains five databases that contain newly resolved and unresolved uncertainties, mistakes, risks, opportunities for BiST, and unintended consequences of the system being designed.
6. **Prioritized. lists:** The moderator and the systems engineer consolidate and edit the five databases to create five prioritized [15] lists.

The list of newly resolved and unresolved uncertainties is given to the systems engineer.

The prioritized list of mistakes is given to the author/designer for rework and resolution.

The prioritized list of risks that could adversely affect the system being designed is given to risk management.

The prioritized list of opportunities for Built-in Self-Test (BiST) is given to test engineering.

The prioritized list of positive unintended consequences that could beneficially affect other systems is given to marketing.
The prioritized list of negative unintended consequences that could adversely affect other systems is given to management and the legal department.

7. **PAL:** The systems engineer puts these prioritized lists in the project PAL.
8. **Rework:** The author/designer fixes the mistakes. Each of the other owners will know what to do with his or her list.
9. **Follow-up:** The moderator must verify that all fixes are effective and that no additional mistakes have been created. The moderator checks the exit criteria for completing an inspection.
10. **Update PAL:** The team updates the project PAL (exit use case).

Postcondition: The project PAL has been updated, and the systems engineer is ready to schedule a new inspection.

Specific requirements can be derived from this use case [20] as follows. These requirements came directly from the main success scenario.

Functional Requirements:

FR3-1 The moderator shall form the Inspection Team.
FR3-2 The moderator shall collect the inspection work products and other relevant materials and distribute them to the Inspection Team To Be Determined (TBD) days before the inspection.
FR3-3 The moderator shall chair the overview meeting.
FR3-4 Each member of the Inspection Team shall examine the work products before the actual inspection meeting looking for unresolved uncertainties, mental mistakes, risks, opportunities for BiST, and unintended consequences of the system being designed.
FR3-5 Each member of the Inspection Team shall record and report the number of hours he or she spent inspecting the materials. Typically, this will be 2 h.
FR3-6 The moderator shall chair the inspection meeting.
FR3-7 The recorder shall create and maintain the five databases that contain newly resolved and unresolved uncertainties, mistakes, risks, opportunities for BiST, and unintended consequences of the system being designed.
FR3-8 The moderator and the systems engineer shall consolidate and edit the databases to create prioritized lists.
FR3-9 The systems engineer shall deliver the lists to their respective owners.
Stipulation: Each owner will know what to do with his or her list.
FR3-10 The systems engineer shall put these prioritized lists in the project PAL.
FR3-11 The moderator shall verify that all fixes are effective and that no additional defects have been created. The moderator shall check the exit criteria for completing an inspection.

It is often said that we can impose requirements on our system, but we cannot impose requirements on operators, pilots, and other secondary actors. This is still true. However, here, we are imposing requirements on members of the Inspection Team. That is all right because they are a part of our system.

Nonfunctional Requirements:

NFR3-1 The moderator shall schedule the inspection meeting for 2 h. The moderator shall prepare about two-dozen pages of models and documentation for each inspection.

Author/owner: Terry Bahill
Last changed: January 29, 2021

This is the end of the section describing our process for ameliorating uncertainty in the models and documentation caused by human mental mistakes.

Chapter Summary

In the year 2020, the COVID-19 pandemic devastated the world. Scientists continually produced new scientific results and new models using uncertain knowledge of the virus and the pandemic. Inevitably, predictions made with these models were uncertain. Politicians, supposedly using these scientific “facts” and uncertain models, frequently strutted out new policies – policies that often contradicted previous policies. Because of this flip-flopping, people lost faith in their governments’ abilities to understand and interpret scientific results from models.

In the next few pages, we will show how governments could have used the principles in this chapter to help manage this pandemic. The third activity in the Overarching Process is to model the system. As shown in this chapter, making models is affected by confirmation bias, severity amplifiers, and framing. These phenomena affected the interpretation of scientific results and the assumptions made in models of this pandemic.

The early COVID pandemic models needed improvements. These early models were created and used by scientists, the media, government bureaucrats, and politicians. We will refer to this group collectively as *they*. Their models needed additional knowledge about the COVID virus, such as transmission by asymptomatic carriers, the impact of super-spreaders, a diminished role for children, the role of airborne aerosol-mediated transmission, and the usefulness of wearing masks. And most importantly, they should have predicted human behavioral responses to government-imposed policy interventions and the mental health problems created by social isolation and the closing of schools.

They form a pyramid. Around 10^6 to 10^7 peer-reviewed journal papers are published each year. Of course, not all of them are about COVID or even about science. Only 10% are scientific. These scientific papers are not read by the public. Then there are on the order of 10^5 science writers who convert these papers into layman articles. These articles are given to maybe 10^4 people in the media, who transform them into summaries and snippets fed to the public. The media also report the number of COVID cases and deaths both geographically and temporally. But these numbers are not science. They are just a collection of numbers. They were not derived using the scientific method. They are not even displayed scientifically (with mean, variance, and sources). To further explain this point, consider the Watson-Crick DNA double helix model published in 1953 that won the Nobel Prize in 1962. Now *that* was science. Compare that to the dozens of DNA ancestry tests on the market today, which do their analyses based on cheek swabs and spit. These companies collect those data to build their databases that include millions of people. They have lots of numbers. But we see little scientific usefulness in those numbers. Just collecting lots of numbers does not make it science. Now, back to the pyramid. Around 10^3 bureaucrats (government bureau chiefs and department heads) filter this information to around 100 politicians making policies for the COVID pandemic. Finally, there is one US president who makes the final decisions. All of these people make up what, in this chapter, we call *they*.

As the year went on, they gathered some of the missing knowledge mentioned above and improved their models. However, they never stated that their old models were based on incorrect assumptions and that these old models were being replaced by new models based on new data and new assumptions. At the beginning of this chapter, we wrote,

> A model is a simplified representation of some aspect of a real system. Models are ephemeral: They are created, they explain a phenomenon, they stimulate discussion, they foment alternatives, and then they are replaced by newer models.

Therefore, old models are *supposed to be* replaced by new models: It is the nature of modeling. It is not something to be ashamed of, something to be covered up. It is *good* to state that because of new knowledge and insights, an old model is being replaced by a new model. It is *very bad* to cover up incorrect knowledge and bad assumptions of an old model.

People infected with the coronavirus (COVID) were generally contagious for around 4 days before they showed symptoms, and some infected people never showed symptoms at all. Because people with influenza (the flu) are most contagious for only 1 day before symptoms appear, this contradicted the decision-makers' preconceived notions about how the virus spread. It took a long time for them to overcome this cognitive bias (tunnel vision).

Their risk analyses were puzzling. Elderly people were at the highest risk; however, elderly people did not get the highest priority for prevention, treatment, or vaccinations. Indeed, New York sent elderly infected COVID patients back into nursing homes! In covering up this fiasco, they never explained why, in their risk analyses, the estimated likelihoods and severities gave such a bizarre recommendation.

Initially, there were frantic efforts to accelerate the manufacturing of ventilators. After they were manufactured and delivered, many were not used. It turned out that high-flow oxygen was a better treatment alternative than ventilators. A tradeoff study might have revealed this mistake earlier.

Their tradeoff studies were faulty or nonexistent. In trying to determine how the virus was transmitted, instead of doing a tradeoff study, they jumped to a single-point solution. They assumed that the virus, like the flu, was spread by touching people and contaminated surfaces. They recommended that people disinfect highly touched surfaces often and wash their hands frequently. Initially, they only looked at evidence that confirmed their bias for this mode of transmission. They ignored the alternative that the virus was spread by aerosol-mediated transmission through the air. As a consequence, in March, they told the public that masks were not useful and people should *not* wear them. (Some have suggested that their secret ulterior motive was to save the good masks for medical personnel.) Several months later, they realized that the coronavirus was transmitted through the air. Then they mandated the wearing of masks for everyone, everywhere, all the time.

To prioritize their alternatives, throughout the year, bureaucrats and politicians made decisions without the benefits of documented tradeoff studies. Thus, they missed golden opportunities to gain public trust. One of the prime benefits of a tradeoff study is that making the evaluation criteria and the scores public produces transparency and greatly increases public trust in the decision-makers.

In prioritizing vaccine distribution, they failed to explain and justify their evaluation criteria. Sometimes, this was a deliberate attempt to disguise their true intent, such as with the contrived evaluation criterion “equity.” As a result, their decisions lacked transparency, and people did not trust them. Notably, some vaccination prioritization schemes might have put politicians first (because they were *essential* workers), but they never told that to the public. In fact, the evidence of this that the public saw was media photographs of high-ranking politicians getting the first vaccinations.

They created a divisive fiction that Americans were either science believers (those that agreed with scientific results filtered by bureaucrats and summarized by the media) or science deniers (those who were not convinced by those media summaries). The science deniers were shamed as bad people. Actually, neither of these groups existed [26]. However, Americans were deeply divided. There seemed to be a group that fearfully followed orders and did whatever the government told them to do and another group that did not question scientific results but rather questioned government policies that could hardly have been based on these scientific results, policies such as wearing masks; obeying lockdowns of restaurants, bars, and churches; accepting job losses; closing schools; and letting loved ones die alone. The people labeled science deniers were mostly people who accepted scientific results but disagreed with government policies to ameliorate the problem. The most important activity of the Overarching Process is stating the problem. The invention of science believers and science deniers caused political delight and media churning. But what problem was it supposed to solve?

The biggest problem remaining in 2021 may have been side effects of vaccinations. The scientists probably did risk analyses of the two-dozen admitted vaccine side effects, but the media did not publish their estimated likelihoods and severities. So the public did not understand the risks of the vaccines’ side effects.

The most outstanding success during the COVID pandemic was Investigating Alternatives. The search for a vaccine started with dozens of alternatives. Scientific researchers and companies kept a half-dozen alternatives alive throughout the entire year. Developing a half-dozen vaccines and inoculating millions of people in less than a year was the fastest vaccine development and deployment in history.

Note added in proof: This summary was written in the Spring of 2021. This note is being inserted in the Summer of 2022. The predictions of this section have held up well over the last year and no changes were made to this section.

But that is enough about the COVID pandemic. Let us now summarize uncertainty in systems engineering and design in general.

Uncertainty arises from factors that are both external and internal to the system. Examples of factors that contribute to external uncertainty are changes in market conditions, the operational environment, new competitors or threats, emerging requirements, partial observability, changes in priorities, and delays in maturation times of promising new technologies. By far, the greatest uncertainty is coping with unknown futures. This problem requires designing for alternate futures – the hallmark of resilient design [39, 45]. Internal uncertainties stem from human behavior and unanticipated challenges that surface during program/project execution, system design and implementation, and creating performance requirements.

There are several traditional approaches to dealing with uncertainty depending on the context. Uncertainty may stem from incomplete/fuzzy requirements or technology maturation rate. If requirements and technologies are both stable in a project, then it is relatively straightforward to plan ahead and execute the plan because there is very little uncertainty to further ameliorate. On the other hand, when a project intends to capitalize on new or emerging technologies, uncertainty is best handled by placing "smart bits" or incorporating real options in both system architecture/design and program schedule [37]. Finally, when the technology aspect is relatively stable, but the requirements continue to evolve, then an incremental commitment approach needs to be pursued [13].

This chapter presented a requirements discovery process, a tradeoff study process, and a risk analysis process. It compared and contrasted these three processes and then combined them into one *Overarching Process*. The three original processes could then be viewed as tailorings of the resulting Overarching (superset) Process. This Overarching Process can also be a top-level or a precursor process for MBSE implementations. The Overarching Process itself is not an MBSE implementation. For example, it does not even use SysML diagrams. This Overarching Process is not a subset of MBSE: It is a superset.

In conclusion, the requirements discovery, tradeoff studies, and risk processes shown in the figures of this chapter were developed many years before the Overarching Process was conceived.

This chapter identified activities in the Overarching Process that were responsible for creating uncertainty. Then it addressed uncertainty handling for the three core processes (requirements discovery, tradeoff studies, and risk analysis) and the general Overarching Process. The approach presented reduces overall complexity in uncertainty management by first creating an enveloping Overarching Process, then systematizing the process of uncertainty handling for this Overarching Process, and then instantiating the approach for the three core processes. This approach should appeal to both engineering and management professionals engaged in modeling, analysis, and design of complex systems in the presence of uncertainty.

Cross-References

▶ MBSE Methodologies

References

1. Bahill AT (2012) Diogenes, a process for finding unintended consequences, *Systems Engineering*, 15(3):287–306. https://doi.org/10.1002/sys.20208
2. Bahill AT (2016) Model for absorption of perfluoropropane intraocular gas after retinal surgeries, International Journal of Medical and Health Sciences Research, PAK Publishing Group, 3(5):50–76, 2016. https://doi.org/10.18488/journal.9/2016.3.5/9.5.50.76. http://www.pakinsight.com/journal/9/abstract/4546
3. Bahill AT (2019) *The Science of Baseball: Batting, Bats, Bat-Ball Collisions and the Flight of the Ball,* Springer Nature, NY, NY, *s*econd edition, 2019, ISBN 978-3-030-03032-2
4. Bahill AT, Baldwin DG (2004) The rising fastball and other perceptual illusions of batters. In: Hung G, Pallis J (eds) Biomedical engineering principles in sports. Kluwer Academic, pp 257–287
5. Bahill AT, Baldwin DG (2008) Mechanics of baseball pitching and batting, Chapter 16. In: Ghista D (ed) Applied biomedical engineering mechanics. CRC Press and Taylor & Francis Asia Pacific, pp 445–488
6. Bahill AT, Dean F (2009) Discovering system requirements, Chapter 4 In Handbook of Systems Engineering and Management, second edition, (eds.) AP Sage, WB Rouse, John Wiley & Sons, pp. 205–266
7. Bahill AT, Gissing B (1998) Re-evaluating systems engineering concepts using systems thinking. IEEE Trans Syst, Man, Cybern C. 28(4):516–27
8. Bahill AT, Madni AM (2017) Tradeoff Decisions in System Design, Springer International Publishing, Switzerland
9. Bahill AT, Szidarovszky F (2009) Comparison of dynamic system modeling methods, Systems Engineering, **12**(3):183-200
10. Bahill AT, Baldwin DG, Venkateswaran J (2005) Predicting a baseball's path. Am Sci 93 (3):218–225
11. Bahill AT, Szidarovszky F, Botta R, Smith ED (2008) Valid models require defined levels, International Journal of General Systems, 37(5):533–571
12. Beach LR, Connolly T (2005) The Psychology of Decision Making: People in Organizations. second edition. Thousand Oaks, CA: Sage Publications
13. Boehm B, Lane J, Koolmanojwong S, Turner, R (2014) The Incremental Commitment Spiral Model, Addison-Wesley, Pearson Education, Inc.
14. Bohlman J, Bahill AT (2014) Examples of Mental Mistakes Made by Systems Engineers While Creating Tradeoff Studies, Studies in Engineering and Technology, 1(1): 22–43, https://doi.org/10.11114/set.v1i1.239
15. Botta R, Bahill AT (2007) A Prioritization Process. Engineering Management Journal, 9(4): 20–7.
16. Buchanan BG, Shortliffe EH (1984) Rule-based Expert Systems. Reading, MA: Addison-Wesley
17. Buede DM (2009) The engineering design of systems: Models and methods. Second edition. New York: John Wiley and Sons, Inc.
18. Cohen RB (2003) Teaching Note on A-Rod: Signing the best player in baseball, *Harvard Business School Case Study* 5-203-091, April 16, 2003.
19. Cohen RB, Wallace J (2003) A-Rod: Signing the best player in baseball, *Harvard Business School Case Study* 9-203-047, January 27, 2003.
20. Daniels J, Bahill AT (2004) The hybrid process that combines traditional requirements and use cases, *Systems Engineering* 7(4) pp. 303-319.

21. Daniels J, Werner PW, Bahill AT (2001) Quantitative methods for tradeoff analyses, *Systems Engineering,* **4**(3):190-212, 2001, with a correction published in **8**(1):93, 2005
22. Edwards W (1977) How to Use Multiattribute Utility Measurement for Social Decision making. IEEE Trans Syst, Man, Cybern. 7(5):326-40
23. Estefan JA (2021) Survey of Model-Based Systems Engineering (MBSE) Methodologies, to be published in the *Handbook of Model-Based Systems Engineering,* edited by Azad M. Madni and Norman Augustine
24. Fagan M (2020) Improved Fagan Inspections and Continuous Process Improvement http://www.mfagan.com/index.html
25. Friedenthal S, Moore A, Steiner R (2012) A practical guide to SysML: the systems modeling language, Elsevier, Morgan Kaufman, OMG
26. Hilgartner S, Hurlbut JB, Jasonoff S (2021) Was 'science' on the ballot? *Science*, 371, 6532, 893–894.
27. Hooks IF, Farry KA (2001) Customer-centered products: Creating successful products through smart requirements management, AMACOM, New York
28. Kahneman D (2011) Thinking, fast and slow. Farrar, Straus and Giroux, New York. ISBN: 978-0-374-27563-1
29. Kahneman D, Tversky A (1979) Prospect theory: an analysis of decision under risk. Econometrica, 46(2):171–85
30. Karnavas WJ, Sanchez PJ, Bahill AT (1993) Sensitivity analyses of continuous and discrete systems in the time and frequency domains. IEEE Trans Syst Man Cybern 23(2):488–501
31. Keeney RL, Raiffa H (1976) Decisions with multiple objectives: Preferences and value tradeoffs. New York: Wiley
32. Madni AM (2000) Thriving on Change through Process Support: The Evolution of the Process Edge Enterprise Suite and Team Edge. Information Knowledge Systems Management 2.1 7–32
33. Madni AM (2014) Generating novel options during systems architecting: psychological principles, systems thinking, and computer-based aiding. Syst Eng 17(1):1–9
34. Madni AM (2020) Models in Systems Engineering: From Engineering Artifacts to Source of Competitive Advantage, Conference on Systems Engineering Research (CSER 2020), Redondo Beach, October 8–10
35. Madni AM (2020) Minimum Viable Model to Demonstrate the Value Proposition of Ontologies for Model Based Systems Engineering, 2020 Conference on Systems Engineering Research, Redondo Beach, CA, October 8–10
36. Madni AM (2020) Exploiting augmented intelligence in systems engineering and engineered systems. INSIGHT Special Issue, Systems Engineering and AI, March 2020
37. Madni AM, Allen K (2011) Systems thinking-enabled real options reasoning for complex socio-technical systems programs. In: Conference on systems engineering research
38. Madni AM, Bahill AT (2021) Handling Uncertainty in Engineered Systems, Handbook of Systems Sciences, Gary S. Metcalf et al. (eds), (Chapter 22-1), Springer
39. Madni AM, Jackson S (2009) Towards a conceptual framework for resilience engineering. Syst J IEEE 3.2:181–191
40. Madni A, Sievers M (2014) Systems integration: key perspectives, experiences, and challenges. Syst Eng 17(1):37–51
41. Madni AM, Sievers M (2018) Model-based systems engineering: motivation, current status, and research opportunities. *Systems Engineering*, 21: 172–190
42. McBeath MK, Nathan AM, Bahill AT, Baldwin DG (2008) Paradoxical pop-ups: why are they difficult to catch? Am J Phys 76(8):723–729
43. Mohanani R, Salman I, Turhan B, Rodriguez P, Ralph P (2020) Cognitive biases in software engineering: a systematic mapping study. IEEE Trans Softw Eng 46(2):1318–1339. https://doi.org/10.1109/TSE.2018.2877759
44. Nickerson RS (1998) Confirmation Bias: A Ubiquitous Phenomenon in Many Guises. Review of General Psychology. Vol. 2, No. 2, 175-220

45. Neches R, Madni AM (2013) Towards Affordably Adaptable and Effective Systems, Systems Engineering, Vol. 16, No. 2, pp. 224–234
46. Oakes J, Botta R, Bahill AT (2006) Technical Performance Measures. This paper was presented at the National Defense Industrial Association, *NDIA Systems Engineering Conference,* October 2005. It was also presented at the INCOSE Symposium and was published in the *Proceedings of the INCOSE Symposium,* July 2006
47. Overgaard G, Palmkvist K (2005) *Use Cases: Patterns and Blueprints,* Addison-Wesley, Indianapolis, IN.
48. Pignatiello JJ, Ramberg JS (1985) Off-line quality-control, parameter design, and the Taguchi method-discussion. *J Qual Technol* 17(4):198–206
49. Rechtin E, Maier MW (1997) *The Art of System Architecting,* CRC Press, Boca Raton
50. Rosenblit JW (1985) A conceptual basis for model-based system design, Ph.D. dissertation in Computer Science at Wayne State University (published by University Microfilms International, Ann Arbor)
51. Smith E, Bahill AT (2010) Attribute substitution in systems engineering. Syst Eng 13(2):130–148
52. Smith ED, Son YJ, Piattelli-Palmarini M, Bahill AT (2007) Ameliorating mental mistakes in tradeoff studies. Syst Eng 10(3):222–240
53. Smith ED, Szidarovszky F, Karnavas WJ, Bahill AT (2008) Sensitivity analysis, a powerful system validation technique. Open Cybernetics & Systemics Journal, 2:39–56
54. Wymore AW (1993) Model-based systems engineering. CRC Press, Boca Raton

Terry Bahill is Professor Emeritus of Systems and Industrial Engineering at the University of Arizona in Tucson. He served 10 years in the United States Navy leaving as a lieutenant. He received his PhD in electrical engineering and computer science from the University of California, Berkeley, in 1975. He is the author of six engineering books and 250 papers; over 100 of these are in peer-reviewed scientific journals. Bahill has worked with dozens of technology companies presenting seminars on systems engineering, working on system development teams, and helping them to describe their systems engineering process. He holds a US patent for the Bat Chooser™, a system that computes the Ideal Bat Weight™ for individual baseball and softball batters. He was elected to the Omega Alpha Association, the systems engineering honor society. He received the Sandia National Laboratories Gold President's Quality Award. He is an elected Fellow of the Institute of Electrical and Electronics Engineers (IEEE), of Raytheon Missile Systems, of the International Council on Systems Engineering (INCOSE), and the American Association for the Advancement of Science (AAAS). He is the Founding Chair Emeritus of the INCOSE Fellows Committee. His picture is in the Baseball Hall of Fame's exhibition "Baseball as America." You can view this picture at http://www.sie.arizona.edu/sysengr/. His research interests are in the fields of system design, modeling physiological systems, eye-hand-head coordination, human decision-making, and systems engineering application and theory. He has tried to make the public appreciate engineering research by applying his scientific findings to the sport of baseball.

Azad M. Madni is a Professor of Astronautical Engineering and the Technical Director of the Systems Architecting and Engineering Program in the University of Southern California's Viterbi School of Engineering. He is also a Professor (by courtesy) in USC's Schools of Medicine and Education. He is the founder and Chairman of Intelligent Systems Technology, Inc., a high-tech R&D company specializing in game-based educational simulations and methods, processes, and tools for complex systems engineering. He received his BS, MS, and PhD degrees from the University of California, Los Angeles. His research has been sponsored by both government research organizations such as DARPA, OSD, ARL, RDECOM, ONR, AFOSR, DHS S&T, DTRA, NIST, DOE, and NASA and aerospace and automotive companies such as Boeing, Northrop Grumman, Raytheon, and General Motors. He is an elected Fellow of the American Association for Advancement of Science (AAAS), the American Institute for Aeronautics and Astronautics (AIAA), the Institute for Electrical and Electronics Engineers (IEEE), the Institution for Electronics and Telecommunications Engineers (IETE), the International Council on Systems Engineering (INCOSE), and the Society for Design and Process Science (SDPS). His recent awards include the *2016 Boeing Lifetime Accomplishment Award* and *2016 Boeing Visionary Systems Engineering Leadership Award* for contributions to industry and academia (awards received in Boeing's 100th anniversary), the 2016 INCOSE RMC Special Award for pioneering industry-relevant contributions to Transdisciplinary Systems Engineering, the *2016 Distinguished Engineering Educator Award* from the Engineers' Council, the *2016 Outstanding Educator Award* from the Orange County Engineering Council, the 2014 *Lifetime Achievement Award* from the International Council on Systems Engineering, the 2013 *Innovation in Curriculum Award* from the Institute of Industrial Engineers, the 2012 *Exceptional Achievement Award* from INCOSE, and the 2011 *Pioneer Award* from the International Council on Systems Engineering. He serves on the USC's Council of the Center of Cyber-Physical Systems and the Internet of Things (CCI) and the Steering Committee of USC Provost's STEM Consortium. His research interests include formal and probabilistic methods in systems engineering, model-based architecting and engineering, engineered resilient systems, cyber-physical systems, and exploiting disciplinary and technology convergence in systems engineering. He is listed in the Who's Who in Science and Engineering, Who's Who in Industry and Finance, and Who's Who in America.