Spring 2020

Validation Risk Across Hierarchical Models

John B. Young
Old Dominion University

VALIDATION RISK ACROSS HIERARCHICAL MODELS

by

John B. Young
B.S. May 2002, Virginia Polytechnic Institute and State University
M.Eng. May 2007, Old Dominion University

A Dissertation Submitted to the Faculty of
Old Dominion University in Partial Fulfillment of the
Requirements for the Degree of

DOCTOR OF PHILOSOPHY

MODELING AND SIMULATION

OLD DOMINION UNIVERSITY
May 2020

Approved by:

Barry Ezell, Ph.D.

Eric Weisel, PhD

John Sokolowski, PhD.

Roland Mielke, Ph.D.

Andreas Tolk, Ph.D.

ABSTRACT

VALIDATION RISK ACROSS HIERARCHICAL MODELS

John B. Young
Old Dominion University, 2020
Director: Dr. Barry Ezell

Modeling and simulation are applied in a great many methods across a variety of topics. Model developers and users alike have a professional duty to understand the complexities of the tools and methods they are using. Oftentimes, models that have been independently constructed and executed are used to inform one another for an analytic purpose, and the compatibility of the models is not always addressed. In the literature, great attention has been paid to model validation. When using models constructively with one another, analysts must understand the bounds of model validity and ensure that the combination of models does not generate poor information. The literature reveals significant research on model interoperability and model composability. Special analytic cases of composability in multi-resolution modeling have also been examined in the available research. What is not available, however, is the ability to assess models' abilities to inform one another without violating the validation of either model. Therefore, the purpose of this research is to develop a risk of method to model composability. To develop this method, a macroscopic model simulating large-scale transportation problems will be implemented. An available technique for Model Use Risk Methodology (MURM) will be applied to the macroscopic model to measure its appropriateness for use within its validated space. The model will be decomposed into atomic units of Objects and Processes. Next, a microscopic traffic model will be similarly decomposed into atomic units and be used to inform the macroscopic model. Applying model similarity techniques across the atoms of both models will yield an assessment of their compatibility of one another. The macroscopic model will be

reassessed using the MURM. Changes in its risk-of-use score will be compared against the model elements' similarity to derive a relationship between model similarity and its impact upon model use appropriateness.

With nothing but gratitude and love to my wife Mary. You believed in me when I doubted myself. This obviously could not have been done without you.

Dedicated to my sons Matthew and Theodore, who I hope will witness the fruits of hard work and dedication. May you both find your passions and follow them to their utmost limits.

ACKNOWLEDGEMENTS

The academic journey towards a PhD is long and difficult under the best of conditions. It is even more difficult when you are a working professional with a family. I must thank many, many coaches, mentors, and bosses on my journey who have been understanding, encouraging and helpful.

I owe many professional thanks to Ms. Gretchen Toliver at the Joint Warfare Analysis Center, to Mr. Mike Doyle and COL (ret) Richard Shelton at the Joint Chiefs of Staff, and to Mr. Andy Thompson and Ms. Sara Burlein at the MITRE Corporation. The work I've done in my career at these organizations mirrored my research interests. Without the support of these organizations, this research would not have been possible.

I am grateful to many friends and acquaintances along the way in my professional and personal life: Mr. James "Bex" Bexfield, Dr. Kevin Debisschop, Dr. Dale Herdegen, Dr. Rob Wittman, Dr. Wayne Jerzak, Dr. Julian Ouellet, Dr. George Williamowski, Mr. Matt Selwyn, Mr. Chris Reinhardt, and Mr. Chad Hinsch. Many of these good people talked through my ideas – even the crazy ones, commiserated and advised me on the process, and encouraged me when the workload simply seemed too much. Thank you.

Of course, the key to finishing the actual academic work is the committee itself. Above all, thanks to Dr. Barry Ezell for encouraging me, challenging me, keeping me on target, and bringing his deep knowledge of risk analysis to bear in my efforts. I was privileged to both work with and to be taught by Dr. Andreas Tolk, whose mentorship will not be forgotten and whose deep knowledge influenced my work. Dr. Roland Mielke challenged me to focus my thinking on several important issues. Dr. Eric Weisel was one of my first PhD-level instructors and offered broad insights to my research and influenced many of my research skills. Dr. John Sokolowski leant a critical eye to many of my assumptions. I am forever grateful.

NOMENCLATURE

AICMD	Army Integrated Core Data Model
DIS	distributed interactive simulation
FOM	Federated Object Model
HLA	High-Level Architecture
MOE	measure of effectiveness
MOM	measure of merit
MOP	measure of performance
MURM	Model Use Risk Methodology
OMSC	Object Management Standards Category
OPR	Object-Process-Relationship
PDU	Protocol Data Unit
RtePM	Real-Time Evacuation Planning Model
RTI	runtime interface
SUMO	Simulation of Urban Mobility
V&V	Verification and Validation
VMASC	Virginia Modeling and Simulation Center

TABLE OF CONTENTS

Page

LIST OF TABLES

LIST OF FIGURES

Figure Page

CHAPTER 1

INTRODUCTION

1.1 Purpose of the Research

This research will show that the risk of using a model for a decision is influenced by the integration of a second model into the decision space by decomposing models, applying a similarity metric, and applying a risk framework.

1.2 Problem Description

Models are applied to all manner of topics, including engineering of systems, experimenting on different strategies and policies, analysis, and scientific inquiries. They are virtual laboratories to experiment and make informed decisions in any of those domains. A well-accepted definition of a model that will be used for the duration of this dissertation is that a model is a deliberate abstraction of a real-world system or phenomena that is under investigation for some proposed purpose [1]. This deliberate abstraction means that models cannot answer everything, nor can they be applied to an arbitrary purpose or set of purposes; each model has been abstracted for its own purpose. The purpose of a model points directly to its overall validity and its utility to the decision being made [2]. Briefly, validity tells us that a model is successfully representing the system, systems, or phenomenon that we wish to represent, and conclusions about those systems can reasonably be drawn from the model's outputs.

Model interoperability deals with the ability of two or more models to run concurrently to share and use one another's data. Protocols have been developed to manage the interactions between such models. However, when models are used independently to inform the same decision space, they bring information that has been created under different assumptions, and the models will again have their own independent purposes. Model composability is the use of a model's inputs and insights to inform another model or decision. It can be potentially misleading to use the models together for the same decision space even if they nominally present the same or similar data and phenomena, due to their distinct purposes and their underlying assumptions during construction that make them valid for their respective purposes. Fundamentally, the ability to combine models is a question of the compatibility of the models' purposes.

It is not uncommon to use an existing model as the basis to validate a new model [3]. The assumption here is that the outputs of the model were sufficient for its purpose previously, and therefore if a new model can replicate those outputs, then it can be accepted as valid as well too. The caution is that each model was designed and built separately, and at some level must be different than one another as they replicate and depict phenomena differently. It is imperative to understand each models' structure before comparing them to one another.

Furthermore, models can be, and often are, used to as inputs to one another. High fidelity models are often of much narrower scope and specific purpose than models of lesser fidelity and address particular phenomena with greater specificity. Using a high-fidelity, narrowly-scoped model as an input to a lower-fidelity, broadly-scoped carries with it risks of effectively introducing new data into the decision space that is at best superfluous, and at worst misleading. Such lower scoped models have differing information demands as inputs and have different causative effects captured within them structurally. These high-fidelity models may in fact not be compatible with the lower fidelity, broadly scoped models that they feed. Within the U.S. Department of Defense, this is a common occurrence, even within its well-accepted hierarchy of analytic models that informs investments and programming, resource allocations, and strategy development.

Informal “tuning” of models in analytic domains is much more akin to model composability than to model interoperability [4, 5]. Where model interoperability is concerned with the effective usage of data from one model to another, model composability is concerned with the restructuring of models and their components. Both interoperability and composability are concerned with model reuse. Absent from model composability is a test for suitability of composition. Oftentimes, model composability is assumed or only briefly considered because of the complexity involved to ensure it. The suitability to compose the models is assumed away given that there is a model aggregation effect at hand wherein the detail offered by a high-resolution model is simply lost or not used, and it is not considered that the underlying structures and their causalities within the models may be substantively different and can lead to erroneous inputs to the broad model or erroneous decisions at a macroscopic level.

1.3 Significance of the Research

This research will inform model developers and users when incorporation of a second model into a decision space where one model is already being used is warranted, and under what conditions, and when it adversely affects the quality of decision making.

1.4 Organization

Chapter 1 presented a high-level view of models and discussed the problem. Chapter 2 of this dissertation will overview model federations, some of the underlying concepts of models and model federations, and risk. Chapter 3 will apply a similarity metric and a risk assessment to a simple, canonical example. Chapter 4 will extend similarity metrics into a risk assessment for a case of two existing transportation models of differing scopes and fidelity.

CHAPTER 2

THE PROBLEM OF RISK AND MODEL FEDERATIONS

To address the question of risk within model integration, first a review of what a model federation is will be necessary, followed by an overview of what it means to qualify a model as valid, next a discussion of conceptual modeling, and then a presentation of how risk is defined, evaluated, and managed. Risk is discussed as a function of likelihood and consequence, encapsulated into scenarios. This chapter of this dissertation will address each of topics in turn. Fig. 1 below depicts the major topics that will be addressed in this chapter and how they map to a methodology that will be developed into Chapters 3 and 4.

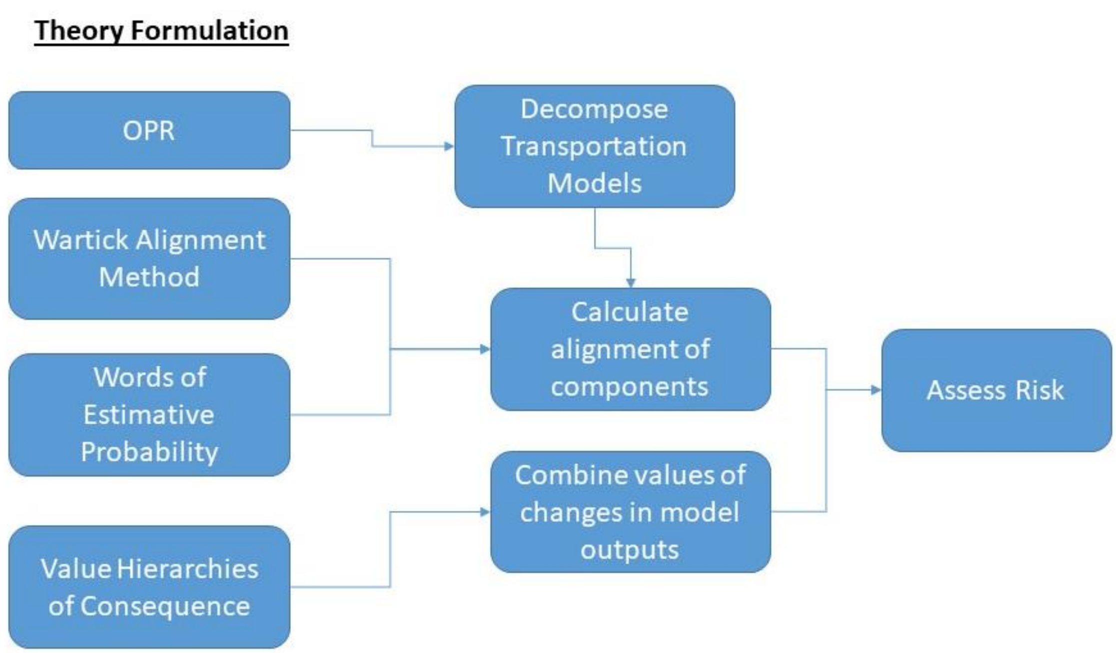


Fig. 1. Methodology overview.

2.1 Model Federations

Model federations are composed of multiple models that can stand separately on their own merit for their own distinct purposes, but have been brought together for some larger purpose. Those independent purposes of models may mean that they have differing representations of the system(s) being models, they each carry with them varying assumptions, and may in fact be incompatible for many of their respective uses, even while representing the same domain space. Each of the component models are referred to as federates. Oftentimes, federations are created for the purposes of training and the federates represent systems within a larger of system of systems context where many individuals will be expected to perform with their individual systems for a larger objective. The United States military and Department of Defense is one of the largest users of models and model federations for this purpose, and training federations are used to train commanders, operators, and decision makers in as-realistic-as-possible situations to prepare for potential real-world contingencies. Running a simulation to train operators on concurrent systems comes with a large set of challenges. Among these problems are developing the appropriate infrastructure, establishing protocols by which distributed simulations can share information, and ensuring that the component models are consistent with one another. This latter concern of ensuring their consistency points to the need of ensuring overall validity among the models and ensuring that they share a common representation of truth in the context of the federation. In these cases, model federations display an attribute known as interoperability which is defined as follows:

Def. 1 Interoperability is the ability of two or more systems or components to exchange information and to use the information that has been exchanged. [6]

Implied in this overview of interoperability is the need for run-time concurrency. That is, the model federation is running all the federates at the same time, otherwise only one of the systems would be using information from the other system(s). In fact, some of the formal protocols and methods used for model interoperability make mention of the run-time interfaces, explicitly making temporal concurrency a requirement. These protocols will be surveyed later in this chapter, in section 2.1.1. In the case of training models, this is a clear need where multiple stakeholders may have discrete tasks and functions and must depend on one another to accomplish their objective(s). As an example, a military force may have any number of numerous functions being performed by single officers or small teams. They may be responsible for logistics, intelligence, air operations, artillery fires, defense, and maneuver. Training these officers to work together with their individual responsibilities must be done concurrently, and done with as much fidelity to real battlefield conditions as possible. Likewise, in testing simulations, there may be multiple systems that have influence on the system under test, and concurrent models may be used to model the behaviors of the systems that stimulate the system under test.

In cases where models are *not* run concurrently, such as in analytic models for investments, allocation of resources, or strategies, there is less-strict guidance on how to ensure models' composability. The definition for model composability is as follows:

Def. 2 Composability is the capability to select and assemble simulation components in various combinations into simulation systems to satisfy specific user requirements. [7]

There are several reasons why models could be—and frequently are—used together in an analytic setting. For instance, one model could be used to "tune" another model [5]. That is, certain behaviors or values of one model could be used to ensure a second model behaves in accordance with the first. Secondly, and very closely related to the first reason, an existing, accepted model could be used to validate a new model. There are validation methods that will be discussed later in section 2.3 that compare the outputs of one model against another to ensure consistent results. The rationale is that if one model is valid, we can use its outputs to compare to the outputs of a second model to ensure its validity. This assumption is a difficult one because the two models clearly are different from one another in their construction and assumptions, and can reasonably be expected to generate distinct results, even if they are only slightly different as will be seen in section 2.3, this is a difference between replicative validity and the stronger structural validity. This research focuses on analytic models that have been independently developed and are used to inform one another. Finally, one analytic model may serve as input to second, broader model. That is, it offers information about a specific piece or pieces of the federation. In this case, the first model likely offers significantly more detail about a subcomponent or subcomponents that are represented in the second model. It may provide extra information or may worse yet detract from the rest of the larger model because other components of the second model are not captured with the same level of detail. As an example, some combat models may offer information about the shear stress on an airplane's wings, but a strategic simulation only needs to know how reliable the plane is in combat [8]. The higher detailed model can represent one, some, or all of the components necessary for the larger model and likely has conceptual components of its own that are distinct from the larger model. These recombinations of models as components of the decision space is in fact model composability.

2.1.1 Federation Standards

Interoperability of multiple models has been an area of practice and concern in the modeling and simulation profession for many years and some of this problem space is well addressed [9, 10]. This section is meant to survey some of the protocols in use, discuss their assumptions, and identify where the protocols are most applicable.

The first notable protocol standard is the distributed interactive simulation (DIS). This system was developed in the early 1990s. In it, federated models are put into the model and receive data in a predefined format, called a Protocol Data Unit (PDU). There are a variety of PDUs managed by IEEE for specific purposes, such as warfare, logistics, radios, and many more systems and purposes. Federates in this case do not have any requirement as to where or how data is generated, which is to say that the models involved are "agnostic" about any other model or data source in the federation. The PDU is meant to prescribe data standards throughout the model federation. But, as the major purpose of interoperability is to ensure the effective usage of data in and among models. In DIS, there is no central system that is responsible for managing data flows; federated models are responsible to monitor the data fields from other federates to update their own statuses. This can occasionally lead to inefficiencies, but the key is that the perception and therefore the meaning of changes in other federates is interpreted at a local level.

The second notable protocol standard is the High-Level Architecture (HLA) and has been around since the late 1990s and was meant to replace the DIS. While it did not completely do that, there has been decreased funding into DIS protocols since its inception. Like DIS, HLA is not a specific software instantiation, but a useful construct to address some interoperability challenges. HLA is an architecture for distributed simulations, which, like the DIS has a run-time concurrency requirement. Reference [10] describes the need for HLA as "based on the premise that no simulation can satisfy all uses and users." They further tell us that HLA is intended to allow the application of one or more simulations to different purposes. This latter statement of differing purposes is critical to this dissertation. HLA is a mechanism in which we seek to broaden a given model's purposes. The HLA Tutorial expands and says that HLA be required to simulate models from different organizations interacting in a bigger picture or context [11]. But this extension of the model's purpose must be caveated based on the model's design. The HLA is a mechanism to begin that process.

The HLA has three major components to it: first the federates themselves, of which there can be many, second is the runtime interface (RTI), and third is the Federated Object Model (FOM). The RTI is meant to be a governance mechanism that allows the federates to communicate with one another, agnostic of where the data comes from. Unlike DIS, the RTI is responsible for routing the appropriate information from one source to another at the appropriate time. The FOM is likened to the "language" of the federation in that it describes the objects, services, and data that will be shared across the RTI when the federation is being executed. As written in reference [12], the mapping from any model to another model (such as an FOM) is a model in and of itself. This recursive mapping problem progresses ad infinitum and presents a paradoxical problem of models mapping to models. However, for practical purposes, the FOM offers a prescription of a use-case-specific data model that is useful to that particular federation. It also enforces representations of certain data, to include necessary considerations of simulated time and update rates.

One of the challenges that exists in the scenarios previously outlined in the analytic models' composability discussion is that there is no RTI when the models are run non-concurrently. The RTI serves as a governing piece of software in the federation that's responsible for routing the correct information from one federate and providing it to another as the federation progresses and enforcing consistent data representations. This runtime capability is of course unnecessary when the models are not run at the same time. But the functionality of enforcing a standard, which in HLA is provided by the FOM, generates the possibility that model federates will present information in their own particular manners which may not be consistent with one another. HLA standards through the FOM enforces a consistent representation of truth at runtime across the federates, and model composability does not have this enforcement mechanism.

2.1.2 Detail vs. Fidelity

While discussing model composability, attention must be paid to terms like resolution, granularity, fidelity, and scope. Terms like *resolution* and *fidelity* can be thought of us the same general concept, which is a general description of how much detail is incorporated into a model. Reference [4] offers the following definitions which will be used in this dissertation:

Def. 3 Scope refers to how much of the real work is represented. [4]

Def. 4 Resolution refers to the number of variables and their precision or granularity. [4]

Detail can be ascribed to the entities themselves, to attributes describing the same entities, or behaviors and processes of the same or similar entities [5]. Scope and resolution often vary inversely to one another. That is, more resolution often is applied when there is a narrower scope and more broadly-scoped models frequently have less resolution. When dealing with broadly scoped models, it is extremely difficult to account for all permutations of precise details. A precise answer does not mean that an answer is accurate, particularly when making large decisions. An oft-heard adage is that "the answer is precisely wrong."

Reference [13] provides a thorough example of two competing models where one better captures detail that the other does not in order to better appreciate the impact of unknown detail modeled as uncertainty. The loss of detail is in fact noticeable when comparing the model results, where the higher fidelity model tends to be less sensitive to probabilistically modeled unknowns. This can be quite important within the broad models such as a complex warfare simulation where there are likely to be a great many unknowns. As a corollary, higher fidelity models are only valid within a much smaller context of assumptions and use-cases where the otherwise probabilistic behaviors are assumed into a narrow window of parameters. This is in part due the fact that their experimental frames differ in the amount of detail they provide to the models, which will be talked about in section 2.4.

Germane to composability is the robustness of their behaviors. If a highly-detailed model is ingested into a higher-tiered but lower-detailed model in order to create a federation model, the behaviors that were produced by the higher-detail model have a less-sensitive response to the rest of the lower-detailed, but more broadly scoped model. This might cause local optima surrounding the behavior(s) captured by the highly-detailed model. Such inflexibility, while admittedly not an optimization would run counter to the danger or unnecessity of optimization presented in reference [14]. The fragility of assumptions in high-fidelity models might be violated or not applicable in broadly-scoped applications, which could make them incompatible with the broadly-scoped model.

2.1.3 Model Similarity

When given that two or more models have been developed independently, with their own assumptions, concepts, degrees of fidelity and experimental frames, we must know how well they align with one another before any an interoperable federation can be composed. Understanding their individual compositions and within what contexts they are valid is necessary to understand the contexts in which they can remain valid together. Wartik et al.'s work develops a degree of alignment metric by which the objects of two models can be compared [15]. A brief synopsis of this method will be given here, as it will be extended later into a risk assessment of model composability. They cite as part of their motivation was the expense of ad hoc modeling solutions without a common object data model. This reiterates the point made earlier when discussing HLA and its FOM being absent from non-concurrent models.

Reference [15] developed a degree of alignment methodology to quantitatively describe how similar objects in two models are in their expression similar information. They proposed a four-tiered alignment table, which is partially reproduced here for illustration. Briefly, each level builds upon the other, with conceptual alignment being the highest possible alignment, meaning that each model represents the same concepts. The entity level is a disaggregation of the conceptual level describing individual entities, sets of entities, or objects within each model. The state level are descriptions of each entity's states and behaviors. The value state is the data domain wherein data types are compared with one another. Table 1 below summarizes the levels used, comparing the Object Management Standards Category (OMSC) and the Army Integrated Core Data Model (AICDM).

In their method, the assessment of alignment at any given level of alignment is assessed a percentage score based on the following criteria in Table 2:

TABLE 1

FOUR LEVELS OF ALIGNMENT BETWEEN MODELS

Level	Participating Model Entities	
	OMSC	AICDM
Conceptual	Standard Object	View
Entity	Class	Entity
State	Method	Attribute
Value	Data Type	Attribute Domain

TABLE 2

ALIGNMENT LEVELS [15]

Value	Standard Phrase	Definition
0%	No Alignment	This value is assigned in either of the following circumstances: • There is no overlap between the models. One model contains an instance of an element that has no analog in the other. • Lack of information in one model prevents alignment analysis.
25%	Low Degree of Alignment	There is some overlap, but it seems coincidental. Overlap might have been achieved by using some attributes in ways that its designers did not originally intend.
50%	Medium Degree of Alignment	There is a moderate amount of overlap, but still a significant disconnect between the models.
75%	High Degree of Alignment	Perfect alignment can probably be achieved by small changes to one model or the other.
100%	Perfect Alignment	There is an exact, unambiguous mapping between the models.

The scoring method is undoubtedly qualitative in nature, but the act of assessing these alignment scores for every conceptual piece of information in a model allows for assessing an averaged alignment score at each level of alignment. The judgement to score each alignment is also somewhat subjective in nature. However, some rigor can be applied to the alignment assessments. Ambiguity of assessments is a well-documented issue for predictions and has been addressed by the United States Intelligence Community. Particular verbiage was proposed by Kent in 1964 and generally accepted as an approximation of certainty, particularly when subject matter expertise is involved to make qualitative assessments [16]. The levels of uncertainty, their meaning, and the numeric value associated with each level espoused in reference [16] by Kent are reproduced below in Table 3.

TABLE 3

WORDS OF ESTIMATIVE PROBABILITY [16]

General region of uncertainty	100%	Certainty	
	93%	Give or take about 6%	Almost certain
	75%	Give or take about 12%	Probable
	50%	Give or take about 10%	Chances are about even
	30%	Give or take about 10%	Probably Not
	7%	Give or take about 5%	Almost certainly not
	0%	Impossible	

Thus, when assessing the levels of alignment, or to interpolate between Wartik et al.'s levels of alignment, the Kent scales can aide in conveying the certainty of how well two concepts align to one another. For instance, if two concepts are of a "Medium Degree of Alignment," or 50%, one could also assess this statement as "probable" or "almost certain" as per the Kent scale, conveying a certain level of certainty / uncertainty about the assessment made.

The levels depicted in the alignment table are not prescriptive, and values can be assessed between levels, so long as justification accompanies the assessment. The Kent scale of estimations of certainty can aid in making interpolations [16]. Degrees of uncertainty of the assessment can help to project interpolated values. It is also important to note that the alignment decision is directional and relative. That is, in the example used in reference [15] by Wartik et al., one model—OMSC—was compared in the context of another: AICDM. It should not be assumed that the comparisons would be the same if the AICDM was measured in the context of OMSC.

At each level of alignment, beginning at the state level and working upwards through entity and conceptual levels, each concept within that level is evaluated from one model to the next and assigned a score from this table. The values are then averaged to attain an alignment score at that level.

For illustrative but arbitrary example, suppose Model A was a naval model and had five types of entities, where Model B was an air warfare model and had six entity types. Suppose those entities were identified as in Table 4.

Those five entities might be scored from Model A to Model B as 75%, 100%, 75%, 100%, and 0% as shown in Table 5. Each score would be given a justification as to why that score was determined as shown above. The overall score at this level of alignment would be:

$$\frac{0.75 + 1.0 + 0.75 + 1.0 + 0}{5} = 0.7$$

TABLE 4

ARBITRARY ALIGNMENT EXAMPLE (1)

Entity	Model A: Naval War	Model B: Air War	Alignment (A to B)	Justification
1	Destroyer	CRUDES (Cruiser / Destroyer)	75%	"CRUDES" is a superset of Destroyer and Cruiser
2	Aircraft Carrier	Carrier	100%	Same, or nearly same entity
3	Fighter	Strike Fighter	75%	Naval model is not as specific as strike fighter
4	Tanker	Tanker	100%	Same, or nearly same entity
5	Submarine		0%	No equivalent
6		Stealth Fighter	0%	No equivalent
7		Stealth Bomber	0%	No equivalent

TABLE 5

ARBITRARY ALIGNMENT EXAMPLE (2)

Entity	Model A: Naval War	Model B: Air War	Alignment (B to A)	Justification
1	Destroyer	CRUDES (Cruiser / Destroyer)	75%	"Destroyer" is more specific than "CRUDES" and implies additional missile defense missions
2	Aircraft Carrier	Carrier	100%	Same, or nearly same entity
3	Fighter	Strike Fighter	80%	Naval Fighter may have additional missions beyond Strike Fighter
4	Tanker	Tanker	75%	Same, or nearly same entity
5	Submarine		0%	No equivalent
6		Stealth Fighter	0%	No equivalent
7		Stealth Bomber	0%	No equivalent

The 0% here shows the presence of one entity in one model, but not included in the other. At a coarse level, these two models would be 70% common with one another.

In the inverse case, mapping Model B to Model A, different alignment assessment might be made, but there are at least different quantities of entity types.

Calculating the alignment of entities 1, 2, 3, 4, 6, and 7, the alignment from Model B to A would be scored as:

$$\frac{0.75 + 1.0 + 0.75 + 1.0 + \ 0 + 0}{6} = 0.58$$

From simply counting the entities, Model B is more broadly scoped—simply because it has more objects included. We also see that the alignment from Model A to Model B is not the same value, suggesting that this alignment value is not transitive, but relative depending on the nature of the alignment. Either model could be used to inform the other, but cannot provide all the information to the other model, even in some of the components that are shared between them. This arbitrary example does not delve deeply into descriptive attributes of the objects that may lead to differences in ascribing an alignment score. This trivial example demonstrates that there are differences in the sets of entities contained in each model, but what is important for the modeling analyst is that even the entities shared in both models are not necessarily the same representation of a truth, and using one model to inform the other is not a simple comparison.

2.2 The Rationale for Multi-scale Modeling

Reference [8] presents four governing reasons why it is use hierarchical models, even knowing that integrating two or more models is likely to present composability challenges. Summarized, they are:

1) Purpose – as stated previously, models have a specific purpose, and there is an appropriate time for high detail and an appropriate time for broadly-scoped models. When a decision maker has a need, he or she should be presented with the information that is germane to their problem with the parameters that they can influence. There may be additional information in models that do not meet this criterion.
2) Analytic Applicability – The results of a model are only as good as its inputs and assumptions which may constrain its broader application. Too much detail in a broad model can confound the sensitivity of the model to individual input(s). So-called "rolling up" or summarizing of the model distills the information to primary decision's needs.
3) Efficient Search – Using a broadly scoped model can highlight the cases of interest than can signal to an analyst or modeler to develop or use more detailed models.
4) Cost – The cost of building models, validating them, collecting data, and analyzing results can be burdensome, particularly when timely decisions are required.

In *Warfare Modeling*, Davis differentiates between what he coins as variable resolution modeling and cross resolution model connections [5]. Variable resolution modeling is a software design simulation with the abilities to expand and contract on the resolution of the model. Variable resolution is a purposeful design decision before a model is even implemented in order to afford flexibility in answering a wider specific question.

Conversely, cross resolution model connection is the act of linking two or more models together that were never designed to be linked together. Davis acknowledges there are methods from a software perspective that can allow one to connect these models, but states it is not necessarily meaningful to do so. In essence, he is talking about the composability of these models. The models may be technically integrated, albeit through some intermediary, though the semantics of doing so may be incorrect [17]. Reference [18] lists several reasons why people may bring together two or more models that were not initially designed to work together, to include attempts to save costs by leveraging legacy models and simulations or due to the growing complexity of the problem space. However, processes that both Davis and North are describing fundamentally in references [5, 18] only speak to the technical interoperability of models. That is to say that the models can potentially exchange data with one another, but the meaning of such an interaction is not guaranteed, and in fact may be nonexistent.

2.3 Validation of Models and Simulations

Validation is a well-understood requirement of successful modeling and simulation projects. There are myriad of definitions of validity in the literature [19, 20]. Many definitions have varying degrees of the phrase "accurate representation" or "from the perspective of the intended users." The key term is the relationship of a model to its intended users, and by proxy, it's intended use. Modeling best practices include an intended use statement or set of statements that provide a brief overview of what the model is meant to accomplish, to represent, and to experiment. Oftentimes, such statements are absent, or are assumed, which can be an impediment to proper validation of a model. However, it is important to note that models cannot arbitrarily answer any question, event within their own domains. This can be a roadblock to model re-use, and in the case of this research, to model composability.

Validation fundamentally answers the question, “Does this model represent the system(s) we wish to study?” or, “Did we build the right model?” [19] What the process of validation must answer is whether the phenomenon and any appropriate causations that might influence it are represented in the model in a manner consistent with our understanding of the real-world system. Sargent usefully offers a paradigm of when validation activities need to occur within a simulation-based study. Validation is the development of a conceptual model from a set of system theories about the real world at the outset of a modeling and simulations study and the examination of model behaviors and outputs compared to our understanding of the real world. The conceptual model is the effort where understanding about the system(s) context also becomes important – to model the system or phenomenon, decisions must be made on what to include into the model and what is deliberately excluded.

While there are many ways to validate models, they relate to the important processes of abstraction from the real system to system theories and again from modeling a conceptual model from system theories. This overarching validation process checks that the move from a real world into a conceptual model (by way of system theories) includes the necessary theories, components, and phenomena that are necessary for the simulation.

Models are deliberate abstractions of the real-world system, and there must be some underlying purpose or intent to the model to select the components of the real world that are necessary for a conceptual model to be developed and a simulation system. So, even models that purportedly examine the same phenomenon or systems may have slightly nuanced differences in their instantiations, sometimes inadvertently through developer or user biases, perceptions, and experiences. Models that are known to be different will certainly have differences in what they include, exclude, how they depict the underlying system theories, and how they handle uncertainties and unknowns.

Reference [21] likens model validation to an act of comparison. That is, validation is a comparison of a model or its outputs to some accepted standard or sets of standards. In Fig. 2 from reference [21], the simulation study process is highlighted, and again shows where validation occurs within the larger process. A simuland is the real-world system that is to be modeled and requirements for a modeling study are derived from that simuland. As in Sargent's paradigm [19], a conceptual model is derived, but in this case directly from the simuland. Petty states that simulands need not have a real-life corollary, which allows for the system theories depicted in Sargent's method. A conceptual model is implemented into an executable model, which develops results. Validation in this method is a twofold comparison process. The first comparison is between the simuland and the conceptual model and the second from the model results to the simuland. These comparison processes should be of sufficient rigor that a model user will have confidence in the model's performance. The comparison from conceptual model to simuland is the process by which we ensure that the conceptual model captures all the relevant components, behaviors, and assumptions that are necessary for the model's purpose. The comparison of model results to the simuland provides a step of rigor at the end of the process that ensures that results of the model are consistent with our understanding of the simuland.

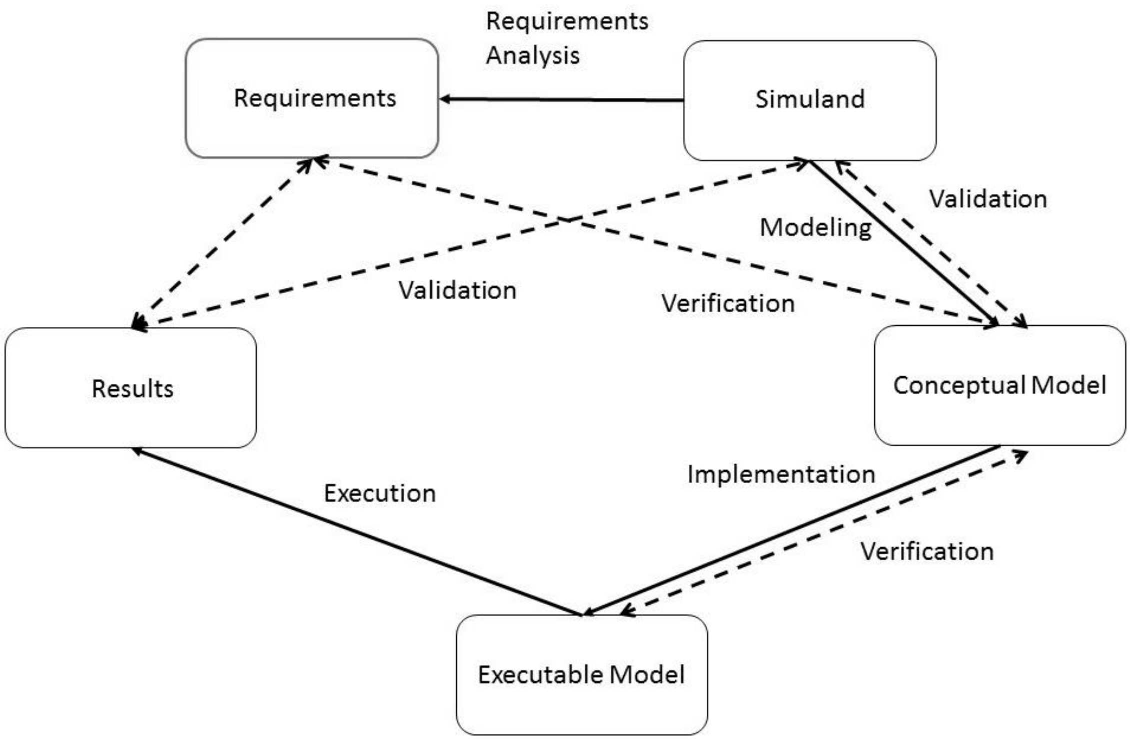


Fig. 2. Validation as comparisons. [21]

Knowing when and where validation activities need to occur within a modeling and simulation study is useful, but a richer understanding of how one validates and what rigor should be applied is still needed. There are many methods by which one can examine a simulation's validity. References [3, 19, 22] offer a wide range of techniques, and all encourage taking model validation as a whole—that is, individual variables and states within a model cannot be validated independently of one another. A high-level summary of many of these techniques is presented in Table 6, the structure of which is adapted from Balci [3] and supplemented with validation methods identified by Law [22] and Sargent [19].

TABLE 6

FAMILIES OF VALIDATION METHODS

Technique	Summary	Sources
Informal	Commonly used. Methods include double-checking one's work, walkthroughs of model execution, and the oft-used face validation to subjectively judge a model's behavior and outputs. While there is subjectivity, this does not imply a lack of rigor; important insights may be learned from expert opinion	Balci [3], Law [22]
Static	Accuracy assessment of the model while not in execution-mode. Exercises include observing the flow of data within the model and ensuring the model is structured correctly	Balci [3], Law [22], Sargent [19]
Dynamic	This family of methods requires model execution and will often involve executing subsections of the model and creating additional code or input data to observe the simulation's tolerances, both as a whole, and submodels within the overall model. Within this family are methods that include observing the model's ability to properly develop meaningful outputs through activities such as trace validation - observing interim results. This can include statistical checks, regression, and observing a model's predictive behavior. Visualization also falls within this category	Balci [3], Law [22]
Symbolic Techniques	Like dynamic testing before it, but attempt to logically decompose cause and effect relationships	Balci [3]
Constraints	Ensuring that the model's internally behaviors as well as the outputs remain within the governing tolerances. This is a mechanism to ensure no underlying assumptions or truth statements are violated	Balci [3], Law [22], Sargent [19]
Formal	Mathematical formulas are used to prove the model's behavior. While an ideal goal of validation, it is frequently not an attainable method. A mathematical argument is considered valid if it follows rules of inference	Balci [3]

Many of these techniques can be quickly distilled into two broader categories of validation, namely subjective and objective [19]. Objective methods are those mathematical tests and proofs that should be independent of any persons responsible for the validation process. When there is no real world or existing system to which to compare, model behaviors and outputs can (and frequently are) compared against the results of other models and simulations. While such an approach is espoused as useful, it is not without caution that one can do so. With multiple models being developed by multiple authors, there can often be semantic, undocumented assumptions and meaning embedded within a model [23]. When comparing against another model for validation purposes, one can easily introduce errors due to the incompatibility of not only the models' languages, but also due to nuanced differences of their experimental frameworks. Using subjective methods, such as the commonly used Subject Matter Expert/SME face validation, can exacerbate that problem.

Reference [3] presents two principal types of validation error that have been derived from statistical testing. Type I error is a model user rejecting a valid model as invalid due to the results of objective tests. This error is sometimes called the model *developer's risk*, as the development would fundamentally be for naught if the *model* were to be rejected. Type II error is called *the model user's risk* and is a failure to reject an invalid model and accepting it as a valid. Type I error is often times correctable by further refinement or development of the model and the largest consequence of such an error is increased cost in the model development. However, Type II error can be catastrophic as it can lead a model user to make an incorrect decision. These two types of errors can be thought of in more subjective or informal methods of validation as well. An additional form of validation error is sometimes referred to as Type III error, where one has answered the wrong question or formulated the problem incorrectly, an idea first espoused by Mitroff and Featheringham in reference [24]. In this sense, a model has been designed and built well enough that it could answer a purpose that may be related, but distinct from the modeling problem on hand. This can be relevant for the model composability problem as the introduction of a new model and its data may change the results of federation's outputs such that they no longer meet the intent of the question being asked.

The answer to whether the model is correct, or accurate enough may be somewhat subjective [25], or at least informed by a model user or decision-maker's personal experience and biases. If one wishes to ensure that the representation of entities and their behaviors is accurate as the general concept of validation requires, then the evaluator of that model will ultimately have to be assured of its validity. The model evaluator can be considered to be the same entity—person or organization—that makes a decision based upon the outputs of a simulation or sets of simulations. While subjectivity in this process may be based on the equity of the models' outcomes with an organization, it may also be personality- and individual experience-dependent. Reference [26] stated "simulation validation in practice is really the process of persuading the evaluators to believe that the simulation is valid with respect to the objective." An evaluator of a simulation brings to bear his or her own experiences, and expects that certain representations are either directly implemented in the model or are at least accounted for within the process. Often in the case of senior decision makers, they have a certain degree of subject matter expertise of their own to bring to bear. Such an evaluator also has spoken or unspoken expectations to the analytic rigor applied to assure a model's accuracy. They may have unspoken rules from their own backgrounds that frame their own judgements as to the validity of a model. Such semantics lie in the evaluators' own mental model of the way the model should behave. It becomes imperative to communicate with decision makers in order to best communicate one's own model results to the users of the model in order to bridge communication gaps at the most basic levels of understanding [27].

Heath and Hill further claim that no model can ever be *proven* to be a valid representation of a problem, and only can only be accepted as valid when such a model cannot be analytically falsified against empirical or system data. This view is not dissimilar from Popper's philosophical view on science in general, wherein theories are only accepted until such time that they can be disproven [28]. George Box famously stated, "Essentially, all models are wrong, but some are useful" [29], a phrase that succinctly sums up this notion that a model is inherently wrong, but is fundamentally a mechanism to learn something or to make a decision if it is done well and meets its intent—that is, valid.

In reference [30], Zeigler et al. describes three levels of validation. The first tier is the weakest form of validity, being *replicative* validity, which suggests that model outputs are within the tolerances of the real-world system's behavior. The second tier is *predictive* validity wherein a model would be able to predict outputs of a real-world system that have not been observed yet. This obviously suggests that a posteriori evidence from the real world will be available to test such validity. The third tier is *structural* validity wherein the internal states of the model mimic the internal states of the real-world system. Zeigler does not explicitly enumerate any methods that would be most appropriate for any of these levels of validity, but it may be reasonable to believe that as one approaches structural validation, more detailed methods such as statistical tests within Balci's family of dynamic methods or perhaps even mathematical formalisms would be required in order to make the assertion a model is structurally valid. These validation checks would belong in the family of objective metrics as set out by Sargent. Less stringent tiers of validation such as replicative validation may still use such statistical tests, but it may be sufficient for informal methods, such as structured walkthroughs or subject matter expert opinion to claim replicative validity.

A decomposition of the modeling process is presented by Jones in reference [31]. Fig. 3 was adapted from Jones' description. As one moves inward from the real world at the outermost circle to simulation instances at the innermost circle, there become increasingly large opportunities to develop multiple interpretations from the previous layer. That is, as one moves from a real-world system to a referent model, there are potentially many referent models that could exist of the real system. Potentially further complicating matters is that there can be more than one real world system, particularly when the real-world system does not exist, potentially because it is actually the proposal under study of the simulation, as is potentially the case in warfare models, or has not been designed yet. The experimental frame is the fundamental "modeling question" wherein the model has to be valid or not, and that experimental frame points to a given model's purpose and intent. Perhaps unsurprisingly, there are an infinite number of experimental frames that could be applied to a real-world system. Thus, the experimental frame creates the possibility for numerous base models, numerous conceptual models, and still more simulation instances. There are potentially infinite ways to model the real-world system. For models to be valid, they must consistently represent truth from the outer-most layer Real-World all the way through to the instantiation of a computerized model.

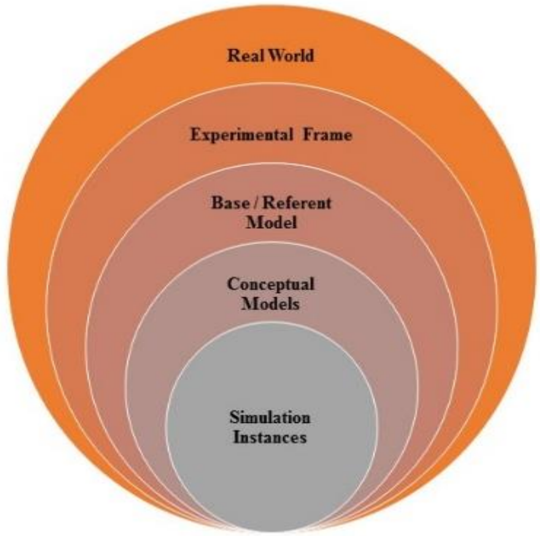


Fig. 3. Problem space decomposition. Adapted from [31].

Thus, there can be any number of simulation instances that are valid and answer their intended purposes. Even when those purposes are extremely similar to one another, the models may in fact be substantively different. Validation is a structured exercise of tracing a model and its representations back from the model instantiation to the real world and ensuring consistency throughout [3, 19].

2.4 Conceptual Modeling and Experimental Frames

As presented in the previous section, the modeling process starts with the construction of a conceptual model based on a referent model within a given context, or experimental frame. Validation of the conceptual model is also the first major step in a validation and verification assessment. Conceptual modeling is the exercise of determining what is to be captured in the model, what assumptions will be made, what data will be incorporated, and what the model's structure shall be, all based on the intention for the model's usage. In reference [32], Tolk and Turnitsa distill the definition of conceptual modeling to the following:

Def. 5 "Modeling is the purposeful process of abstracting and theorizing about a system, and capturing the resulting concepts and relations in a *conceptual model*." [32]

Conceptual modeling is the cornerstone of a good modeling study, and begins before there is any computerized representation of the system(s) under study. In reference [1], Turnitsa develops the Object-Process-Relationship (OPR) method as a description of conceptual components or atomic elements in models. He defines components as "identifiable parts of the model which represents some knowledge that makes up the whole model." The most common components of a conceptual model are distilled into one of the three classes.

To review the OPR method, objects are persistent entities within the system that maintain their identity within the model and remain stable until acted upon by a process. In this sense they are nearly Newtonian. Objects will represent an artifact of the system [1]. Furthermore, objects in this paradigm will carry certain distinctive attributes that can be qualitative or quantitative in order to differentiate it from other objects, which is part of the similarity process discussed in 2.1.3. Processes represent the dynamic part of a model and the requisite causality of the phenomena being modeled. This conceptual element describes changes and transformations within the model [1]. "A process is a marker between two states of a model," that differentiates states before and after in the dynamic process of the overall model where objects change their state. The model's state changes as a function of all the component entities states. The third conceptual component of the OPR paradigm is the relationship which is a component that links other components. For example, a relationship could link a certain object with a certain process, or potentially several objects to several processes.

Each of these three elements carry elements of knowledge of the system being modeled. The word "assumptions" carries significant meaning for different modelers and model users [33]. It is not meant to be synonymous with a conceptual model, but a conceptual model cannot exist without a list of assumptions. Reference [34] linked the list of assumptions to the conceptual model and demonstrated that even trivial models can have nuanced differences in assumptions that can have profound impact on the development and execution of a model. The understanding of what assumptions and assertions are for the system(s) under study, the model, and the model's context are critical to validating a model. With inconsistent understanding of these concepts, there is risk in not developing a sufficiently accurate model of the problem at hand.

Adding to this is the idea that even within the same problem domain, reference [27] highlights potential semantic nuances to the domain. Certain words or phrases come loaded with meaning to practitioners in one field that may not come with exactly the same meaning to practitioners in other domains. Developers and users of modeling all come with their experiences and biases [35].

With a gap in understanding what assumptions *are* and what various terms may or may not mean within the context of a model, there are significant obstacles to overcome. Development of a consistent model that represents truth of the problem and its domain such that stakeholders all understand it is not a trivial task, and to validate such a model contains all of those individuals' understandings, experiences and biases, which is why so often, *validation* is sometimes seen as a subjective endeavor. There is a deliberate effort to meet stakeholders' expectations.

An often-overlooked piece of validation is the context in which a model is the framework in which it is valid. The variance of a context can radically change what is expected as output. As Denil et al. succinctly phrased in reference [36]: “A model that is valid for one use case can produce invalid results for another.” The validity of any given model can only be measured against the context in which it was designed [37]. Zeigler et al. describe the model instantiation of the *source system* as its experimental frame [30]; their definition of experimental frame is:

Def. 6 An Experimental Frame is the operational formulation of the objectives that motivate a modeling and simulation project. [30]

The full interoperability of two or more models depends upon the compatibility of its conceptual components and of their respective experimental frames or contexts. Both the components and the experimental frames speak to the *purpose* of a model, the key component of its validation. When two or more models as federates are used to inform a single decision, they have brought their own particular contexts and concepts with them, and to use them together generates a change of the new model’s concepts and context. The question is: what risk does this create in validating the federation model when considering each federate model’s development?

2.4.1 Validity Summarized

Model validation, however it may be conducted, is a function of several key components. Those components are the *modeling question*, i.e. the set of phenomena that we wish to model, the *experimental frame*, and the model’s *purpose*. Logically, model validity can be represented as:

$$V_m = f\{Q, EF, Purpose => (O, P, R)\} \quad (1)$$

Where

V_m = Validity (V) of the model (m)

Q = Modeling Question

EF = Experimental Frame

$Purpose$ = Purpose for which the model was designed, which is composed of:

O = Modeled Objects

P = Modeled Processes

R = Modeled Relationships

There are many considerations for validation, and a well-defined question and purpose aid in both selecting a model and ensuring that the model is valid for the question.

2.5 Risk

Thus far, this dissertation has reviewed at a somewhat high-level concepts related to model theory and model validation. To apply a risk assessment to the usage of two or more models in a single decision space, an overview of what risk *is* and how it is assessed is required.

Generally, risk is some combination of uncertainty and of damage [38]. Often, risk is seen as the *product* of uncertainty and damage, but this need not be the case. The multiplication of uncertainty and damage assumes that the decision maker is risk neutral and does not have a particular preference in mind [39]. In reality, the calculation of uncertainty or the calculation of damage might be non-linear and there are particular outcomes that may be significantly worse than others. Kaplan and Garrick stress the need for some sort of a loss or damage as a key component of risk, beyond simple uncertainty [38]. They also espouse risk as a triplet, wherein each potential outcome is enumerated as a scenario, a probability, and a consequence. These scenarios aid in the development and enumeration of outcomes that are undesirable so that they can be addressed and mitigated. Therefore, risk can then be expressed as:

$$R = \langle S, P, C \rangle \quad (2)$$

Where:

R = Risk

S = Scenario

P = Probability

C = Consequence

Then, when risk is assessed, a table is generated wherein each scenario is listed, its likelihood or uncertainty, and the damage that could be expected if this scenario were to come to pass.

In the scenarios that will be developed in this dissertation to assess risk, the tuple from reference [38] and shown in equation (2) will be extended to include two different probabilities. The probability will be represented by two distance metrics, the first metric is the distance between the component models at the object level and the second probability will be the distance between the first model and the second model at the process level. These probability values are derived from the OPR method discussed in section 2.4, and addresses the similarity between models based upon their structure of objects and of processes.

The distance metric is simply 1—the Alignment Value, as was discussed in section 2.1.3. To use the example set forth there, where an alignment score of 0.55 was found, the distance metric would be 1 – 0.55 = 0.45. Where a 1 would be a perfect alignment between the two models. So, the risk triplet is extended to a tuple:

$$R = \langle S, D(Obj), D(Proc), D(Rel), C \rangle \quad (3)$$

Where:

R = Risk

S = Scenario

$D(Obj)$ = Distance between models in the Object Domain

$D(Proc)$ = Distance between models in the Process Domain

$D(Rel)$ = $D(Proc)$ = Distance between models in the Relationship Domain

C = Consequence

Reference [2] discusses a structured, systematic method to risk analysis and discuss the validity of that risk analysis. Many of the elements of risk analysis discussed there have a recurring theme of completeness. That is, completely describing assumptions, scope, scenarios, methods, and data that are used in a risk analysis. So, in order to properly conduct a risk assessment on the conjunction of two models, we need to completely enumerate the possible scenarios with these two models, the assumptions, and how damage and uncertainty can be expressed.

Fundamentally, the models that are brought into this ad hoc federation will have different, but somewhat similar experimental frames. They carry their own assumptions and biases, and each may conceptually define objects processes and relations differently —perhaps substantially, or perhaps nuanced, but there is a difference. The risk to using the models to inform one another or to inform a single decision is that the experimental frames, objects, processes, and relations of each model has a difference that may not be apparent, driving uncertainty. The damage or consequence of using multiple models in one decision space is a degradation of the decision space for which we were using a model in the first place. Introducing a new model to the decision space can actually lead to a worse decision being made due to the presence of unaccounted-for assumptions, structures, and causalities within the models. In effect, the introduction of a new model can introduce additional caveats that limit the validity of the outputs or generate new constraints on the model's outputs that may not be sufficient to answer the federation's overarching purpose. The addition of caveats and constraints could mean that we have developed a Type 3 error, where we have answered a question that is different, perhaps only nuancedly so, then what was intended. The table presented in Appendix A enumerates the possible risk scenarios that could result from the integration of two or more models. Each scenario lists a permutation where hypothetical Models A and B completely share, partially share, or do not share the critical components of Objects and Processes. A simple one to one mapping cannot always be assumed, which is why the similarity metric outlined in section 2.1.3 can be used to determine how well one model object maps to another model object or how well one model's process maps to the other model's processes.

The cases where risk needs to be examined are those cases where the models share some form of overlap between two types of components – their objects and processes. As an illustration, Fig. 4 shows a Venn diagram depicting two arbitrary models with some form of overlap. In this image, each model has some set of objects that are unique to its concept of a system, each model has some set of processes that are unique to its concept of a system, and the share some set of both objects and processes. What is not depicted in this image is the nature of relations as a set of components in the model. The components that are shared between the models are not necessarily a one to one mapping, either. That is to say, that the models' shared components are not necessarily exactly the same atomic concept in each model. The similarity metric presented in section 2.1.3 describes the terms of how these elements differ from one another and can be used to describe the uncertainty in the risk scenario table of Appendix A.

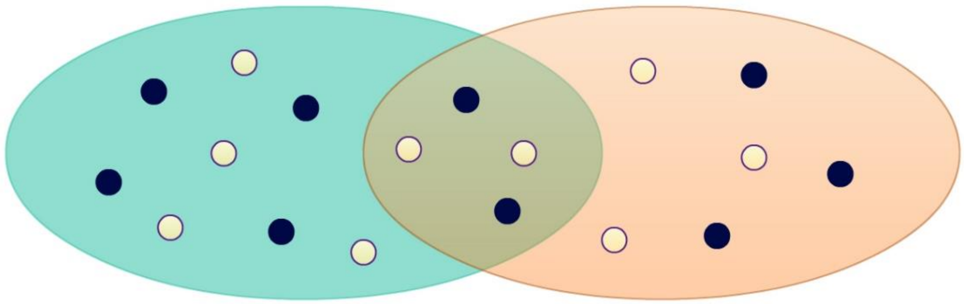


Fig. 4. Venn diagram of two models and their conceptual components.

2.5.1 Model Use Risk Methodology

In the Model Use Risk Methodology, or MURM, the Johns Hopkins University Applied Physics Laboratory applies risk analysis to model usage in *general*, though not explicitly to model composability or interoperability [40]. It is meant to understand how risky it is to use a single model within the context of a single decision. This method offers a useful definition of risk applied specifically to modeling and simulation: "The probability that inappropriate application of M&S Results for the intended use will produce unacceptable consequences to the decision maker." In this methodology, they decompose both the concepts of probability and of consequence. An overview will be presented in this subsection.

The mathematical definition of model use risk that they proffer is:

$$M\&S\ Use\ Risk = p[(C \wedge E) \wedge (C => E)] \quad (4)$$

Where

Causes C = Inappropriate Application of M&S Results and,

Effects E = unacceptable consequences to the decision maker.

In plain words, this function states that there is a probability that using a model inappropriately would cause adverse effects to the decision maker and that the model was actually applied inappropriately and caused the adverse or unacceptable consequences to the decision maker.

Their definition of *causes* is a logical union of a lack of clarity on the model's intended use, an adverse impact on decision if a model's capability is not achieved, and an incorrect recommendation to employ or not to employ a model. Logically, this is expressed as:

$$p(Causes) = p(C_1 \cup C_2 \cup C_3) \quad (5)$$

Where

$C1$ is the lack of clarity,

$C2$ is the importance of a modeling capability or functionality, and

$C3$ is the confidence in the model's results.

The authors develop a table for each of these components with differing descriptions at each level and use the maximum information entropy principle to probabilities associated with each level. Table 7, Table 8, and Table 9 for each of the factors are as follows:

TABLE 7

FACTOR C1: CLARITY [40]

Factor Level	Clarity of Intended Use	P(True)
A	Lucid	0.1667
B	Partial	0.5
C	Unclear	0.8333

TABLE 8

FACTOR C2: IMPORTANCE [40]

Factor Level	Consequence / Mitigation	P(True)
A	Negligible consequence / Mitigation not required	0.038
B	Negligible consequence / Mitigation complete	0.115
C	Negligible consequence / Mitigation partial OR Minor consequence / Mitigation compete	0.231
D	Negligible consequence / Mitigation impossible OR Minor consequence / Mitigation partial OR Serious consequence / Mitigation complete	0.423
E	Minor consequence / Mitigation impossible OR Serious consequence / Mitigation partial OR Grave consequence / Mitigation complete	0.654
F	Serious consequence / Mitigation impossible OR Grave consequence / Mitigation partial	0.846
G	Grave consequence / Mitigation impossible	0.962

TABLE 9

FACTOR C3: CONFIDENCE [40]

Factor Level	Recommended Confidence	P(True)
A	Confidence percentiles 80 to ≤ 100: Very High	0.05
B	Confidence percentiles 60 to ≤ 80: High	0.15
C	Confidence percentiles 40 to ≤ 60: Medium	0.25
D	Confidence percentiles 20 to ≤ 40: Low	0.35
E	Confidence percentiles 0 to ≤ 20: Very Low	0.45

Once ascribing levels to each of these factors, the union of causes can be calculated. As an example, of $p(Causes)$, consider a model with partial clarity (Clarity Factor, Level B), negligible consequence / mitigation possible (Consequence Factor, Level D), and high confidence (Confidence Factor, Level B). The calculation of $p(Causes)$ would be:

$$p(Causes) = p(0.5 \cup 0.423 \ \cup 0.15) = \ 0.755$$

The next major consideration is the effects that a model has upon the decision factor. As the authors point out, the weighting ascribed to the occurrence of an acceptable consequence is dependent upon the decision maker. As an illustration, Table 10 shows a simple, three-level table for potential effects. It assumes a linear, and therefore risk-neutral posture, which may not be true for all decision-makers, but can serve as a starting point to understand a decision maker's risk tolerance.

TABLE 10

STATE TABLE FOR EFFECTS [40]

Factor Level	Unacceptable Consequences to Decision Maker	Level Weighting	P(Effects)
A	Probability of unacceptable consequences is low	1	0.167
B	Probability of unacceptable consequences is medium	3	0.5
C	Probability of unacceptable consequences is high	5	0.833

The MURM is a useful tool for determining what risk is associated with using a model for a purpose, and includes a description of that purpose within its probabilistic assessment. Their definition does not address data sources; which, in this dissertation is another model. Therefore, this dissertation will expand upon this methodology and measure the impacts on these factors by applying a second model into the decision space to inform that decision. It is expected that the purposes of multiple models may not be compatible and can drive an increase in risk.

Their methodology also largely focuses on a model's outputs, and the impact of those results on a decision. It does not address model composition or structure, and, as was presented in section 2.3 of this dissertation, reference [30] discussed several levels of validity, and discussed structural validity as one of the more stringent forms of validity. The MURM is meant to aid in finding appropriate verification and validation (V&V) methods for a model's causes and effects, so it is plausible that the MURM could point to rigorous forms of validation such as the structure of a model. This dissertation will decompose models' structures into their conceptual components in order to examine the impact of one upon the other within the context of a risk assessment.

CHAPTER 3

MATHEMATICS OF RISK IN A SIMPLE EXAMPLE

3.1 Structure of Methodology

This research method will be a mixed-method approach, leveraging some quantitative approaches in support of qualitative study issues. The overarching method will be qualitative in nature as the research's main points are primarily exploratory or interpretive [41]. It is also highly probable that the data available will be sparse, further supporting a qualitative study. This section will discuss an example, simple problem and derive a risk assessment about the two models' interoperability.

3.2 A Canonical Example: Computing the Mean

As an example of multiple models performing similar functions, consider three different algorithms for computing the mean for a sample. All of them are considered Pythagorean means. The first such algorithm is the well-known Arithmetic Mean, the second is Geometric Mean, and the third is the Harmonic Mean. A short description of each:

The Arithmetic Mean or simply "average" computes a value by summing all values in a set together and dividing by the sample size. Mathematically, it is defined as:

$$A = \left(\frac{1}{n}\right)\sum_{i=1}^{n} a_i \tag{6}$$

Where

A is the computed average,

and n is the sample size.

The Arithmetic Mean calculates a value that trends towards the center of the sample set and yet provides equal weighting to all values in the sample set. Numbers lower than the mean are offset by numbers higher than the mean.

The Geometric Mean is computed by multiplying all values in a set together and taking the n^{th} root of the product. It's often used to compare differing items with differing properties. It shows the central tendency- or typical values- of a set. It mathematically defined as:

$$A = \sqrt[n]{\prod_{i=1}^{n} ai} \tag{7}$$

Where:

A represents the computed average,

and n is the sample size.

The Harmonic Mean is computed by adding the reciprocals of all the values in a set together, dividing by the sample size, and taking the reciprocal of the result. It is useful when comparing rates.

$$A = \left(\frac{\sum_{i=1}^{n} a_i^{-1}}{n}\right)^{-1} \tag{8}$$

Where:

A represents the computed average,

and n is the sample size.

In Appendix B, four samples are randomly generated from a uniform random distribution in Microsoft Excel in four different series. The Arithmetic Mean, the Geometric Mean, and the Harmonic Mean are calculated from these random sets of 40 samples each and presented here in Table 11. The samples in each of these series are completely arbitrary and used for illustration purposes.

TABLE 11

PYTHAGOREAN MEANS

	Arithmetic Mean	Geometric Average	Harmonic Mean
Series Alpha	21.550	16.244	9.510
Series Beta	19.650	14.637	8.230
Series Gamma	23.325	19.681	15.467
Series Delta	18.950	13.768	8.215

3.3 The Conceptual Components of Each Algorithm

The OPR method used by Turnitsa [1] and discussed in section 2.4 can be applied to the simple example here. To begin, let us highlight the objects within each model. Each of these three averaging models uses a sample set of 40 samples. Each individual sample carries with it a singular value. For example, referring to Appendix B, the seventh value of Series Gamma is 29. The value of the object is an attribute that describes the sample. To use the same example in Series Gamma, X7 is the object, and the value of that object is 29. Each model also has an object that is called *sample size*, and it is simply a count of the number of values contained in the set. In the example set forth here, the sample size is arbitrarily 40. These are the obvious types of objects in each model, but the less obvious objects are the sum of all the samples in the Arithmetic Mean that will be divided by sample size, the product of all the samples in the Geometric Mean that will be rooted by the sample size, and in the Harmonic Mean - the reciprocals of each sample and the sum of all the reciprocals. Recalling the definition of object discussed in Chapter 2, these objects have a value that is generated by the processes discussed later, and until those values are ascribed to these objects, the objects carry no meaning.

From the process perspective, the models begin to diverge. The arithmetic mean contains two processes. The first process is a summation of all the samples. The second process is a division of that sum by the sample size.

Next, the Geometric Mean contains two processes of its own. The first process is the multiplication of all the samples with one another. The second process is the rooting of that product by the sample size.

Finally, the Harmonic Mean contains four processes. The first process is taking the reciprocal of each sample. The second process is the summation of all those reciprocals. The third process is dividing that sum by the sample size. The fourth process is taking the reciprocal of that result.

Table 12 below summarizes the components of each these models

TABLE 12

SUMMARY OF ALGORITHMIC CONCEPTUAL COMPONENTS

	Arithmetic Mean	Geometric Average	Harmonic Mean
Object 1	Individual samples	Individual samples	Individual samples
Object 2	Sample size	Sample size	Sample size
Object 3	Sum of Samples	Product of Samples	Reciprocals of Samples
Object 4	NONE	NONE	Sum of Reciprocals
Process 1	Summation of Samples	Multiplication of Samples	Reciprocal of each Sample
Process 2	Division of Sum of Samples by Sample Size	Root of Product of Samples by the Sample Size	Summation of Reciprocals
Process 3	NONE	NONE	Dividing Sum of Reciprocals by Sample Size
Process 4	NONE	NONE	Reciprocal of Quotient

3.4 Calculating the Risk of Joining These Models

In a scenario where these models were being compared for compatibility, the similarity of objects from one model to the next is as follows.

For illustration, the arithmetic mean will be compared against the harmonic mean using the alignment methodology introduced earlier. This case is interesting because the two different algorithms have differing numbers of both objects and of processes. To move from the arithmetic mean to the harmonic mean, each object concept in the arithmetic mean will be inspected for a mapping or a corollary within the geometric mean, and Table 13 is generated.

TABLE 13

ALIGNMENT OF OBJECTS FROM ARITHMETIC MEAN TO HARMONIC MEAN

Arithmetic Mean Object	Alignment	Rationale
Individual Samples	100%	Each algorithm uses presumed unbiased random numbers in its distribution
Sample Size	100%	Each algorithm uses an object that is based on the individual samples
Sum of Samples	0%	The object that holds the value of the sum in the arithmetic mean does not have a corollary in the harmonic mean

The overall alignment of the objects from arithmetic mean to harmonic mean is:

$$(1.0 + 1.0 + 0.0)/3 = 0.67$$

The distance between these two models with respect to objects is:

$$1 - 0.67 = 0.33$$

The alignment of processes from the arithmetic mean to the harmonic mean follows a similar construct where each process concept is inspected for a mapping in the other algorithm (Table 14).

TABLE 14

ALIGNMENT OF PROCESSES FROM ARITHMETIC TO HARMONIC MEAN

Arithmetic Mean Process	Alignment	Rationale
Summation of the Samples	100%	The harmonic mean calculates a sum by adding a series of numbers together as well
Division of the Sum of Samples by the Sample Size	100%	The harmonic mean divides a sum by a sample size as well

The overall alignment of the processes from arithmetic mean to geometric mean is:

$$\frac{1.0 + 1.0}{2} = 1.00$$

The distance between these two models with respect to objects is:

$$1 - 1.00 = 0.00$$

For completeness' sake, and to demonstrate that this alignment metric is not transitive, let us consider the inverse case, the alignment of the harmonic mean to the arithmetic mean. First, an inspection of the harmonic mean's objects and their mapping to arithmetic mean objects (Table 15):

TABLE 15

ALIGNMENT OF OBJECTS FROM HARMONIC MEAN TO ARITHMETIC MEAN

Harmonic Mean Object	Alignment	Rationale
Individual samples	100%	Each algorithm uses presumed unbiased random numbers in its distribution
Sample size	100%	Each algorithm uses an object that is based on the individual samples
Reciprocals of Samples	0%	Each sample has a reciprocal that has no corollary in the arithmetic mean
Sum of Reciprocals	50%	The sum of the samples' reciprocals is a simple addition that can be likened to the summation of samples in the arithmetic mean, though it clearly depends upon different input data

The overall alignment of the objects from arithmetic mean to harmonic mean is:

$$\frac{1.0 + 1.0 + 0.0 + 0.5}{4} = 0.63$$

The distance between these two models with respect to objects is:

$$1 - 0.63 = 0.37$$

The alignment of processes from the harmonic mean to the arithmetic mean follows the same construct as before where each process concept is inspected for a mapping in the arithmetic mean algorithm (Table 16).

TABLE 16

ALIGNMENT OF PROCESSES FROM HARMONIC MEAN TO ARITHMETIC MEAN

Harmonic Mean Process	Alignment	Rationale
Reciprocal of each Sample	0%	The arithmetic mean has nothing for a reciprocal value
Summation of Reciprocals	100%	The summation process is like the summation process used in the arithmetic mean
Dividing Sum of Reciprocals by Sample Size	100%	The division of a sum by the sample size is identical to the process of dividing a sum by the sample size
Reciprocal of Quotient	0%	There is no similar process in the arithmetic mean

The overall alignment of the processes from arithmetic mean to geometric mean is:

$$\frac{0.0 + +1.0 + 1.0 + 0.0}{42} = 0.50$$

The distance between these two models with respect to objects is:

$$1 - 0.50 = 0.50$$

To summarize, the distance in objects from the arithmetic mean to the harmonic mean is 0.33 whereas the distance in objects from the harmonic mean to the arithmetic mean is 0.37. The distance in processes from the arithmetic mean to the harmonic mean is 0.00. The distance in processes from the harmonic mean to the arithmetic mean is 0.50. It is perhaps unsurprising to conclude that these two models would pose a significant risk if they were used inappropriately.

3.5 Demonstration with Robust Models

This section will describe the experiment using two analytical readily available models and the results that are expected to be found between them.

3.5.1 Models to be Used

In this experiment, two models—RtePM and SUMO—are used. Both models are within the transportation domain and were selected because of their differences in scope and fidelity.

Real-Time Evacuation Planning Model, or RtePM, was developed to aid in emergency evacuation planning. Virginia Modeling and Simulation Center (VMASC) developed this tool in support of first responders and the Department of Homeland Security. This model will serve as the macroscopic model in the experiment. In it, road networks and their capacities are represented to examine the effects of heavy volumes of traffic attempting to evacuate a particular geographic region; its key metric is the time required to evacuate the area.

Simulation of Urban Mobility, or SUMO, is a free and open source simulation tool that is used for traffic analysis and is capable of modeling intersections to highway interchanges and a variety of other vehicular traffic (such as bicycles or pedestrians). SUMO will serve as the microscopic model in this experiment, offering higher fidelity and narrower scope.

3.5.2 Data Organization and Alignment

Each model will be decomposed using the OPR method above into its constituent conceptual components. Using available documentation, to include users' manuals for both and the VMASC-sponsored Validation &Verification study on RtePM, each conceptual models' conceptual components will be tabulated and justified.

Using the alignment method, and inversely, the alignment method discussed in section 2.1.3, each model will be compared to the other to arrive at an alignment assessment.

3.5.3 Risk Assessment

The first major step of the risk assessment will be to determine what conceptual elements are included in each of the models. This will be done by carefully examining model documentation and validation studies, when available.

The next major step in the risk assessment is to determine the misalignments between the models with respect to each of their Objects, Processes, and Relationships. Set theory will be used to compare the sets of concepts across the models depending upon their class of misalignment. Value hierarchies will be used to interpolate similar concepts and their elements.

A series of sets of potential nominal model metrics' changes will be developed with a design of experiments. The combinations of metrics changes will be combined with the calculated alignment values to develop integration risk curves.

CHAPTER 4

THEORY DEVELOPMENT

4.1 Risk Scenarios

A single model used for decision making presents an inherent risk to the decision(s) at hand [3, 40]. Risk to the quality of the decision is compounded when models are integrated with one another. As discussed in section 2.5, Risk is a function of scenario, probability and consequence. The first major step in exploring Risk of integrating two models is to enumerate those scenarios. Reference [42] enumerates three major means by which models can differ from one another. They are a misalignment of scope, misalignment of resolution, and a misalignment of structure. Each is summarized in turn below. These major misalignment categories will be used to define the risk scenarios of two models integrated with one another. In the paragraphs that follow, the concepts that are used to highlight the differences between models can be either objects, processes or relationships. The OPR taxonomy that is used in this dissertation will apply these risk scenarios to all three conceptual dimensions—objects, processes, and relationships—in order to *define* each dimension of misalignment in turn, to allow for the Risk tuple that is a function of Scenario, Consequence, Objects' Alignment, Processes' Alignment, and Relationships' Alignment.

The first major risk scenario is misaligned scope. Scope refers to the quantity of concepts that are included in each model. It can be thought of as the "breadth" of the model, and is a count of the concepts that are included in the model either by design or by assumption. As depicted in the Fig. 5 below, one model may contain a set of concepts while a second model has a different set of concepts. The two models may have significant overlap or very little overlap. At least one modeling system contains a major concept not found in the other system. In the image below, each system contains a major concept not found in the other, while the share two major concepts between them.

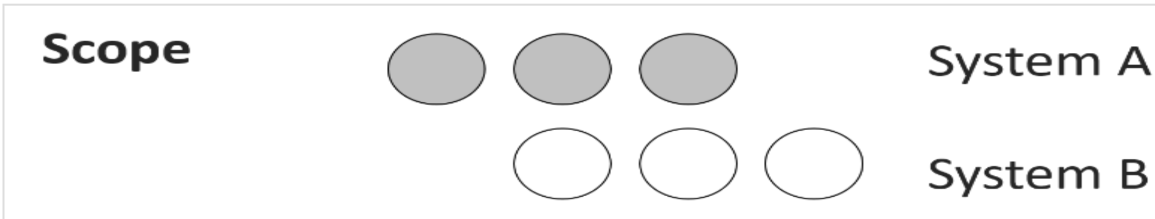


Fig. 5. Model concepts misaligned by scope. [42]

The second major risk scenario is misaligned resolution. Resolution refers to the level of precision that is incorporated into the model to describe each concept. Where one model may have a succinct description for its own purposes, a second model may have a more detailed description of the same concept. The detail used to describe the components may be by explicit design or may be implicit assumptions in the model. Fig. 6 below depicts System A as having 3 major concepts where System B has 4 concepts in place of each concept in A, for a total of 12 concepts. The ratio of concepts in B to concepts in A need not be fixed, nor need be consistent from one concept to another.

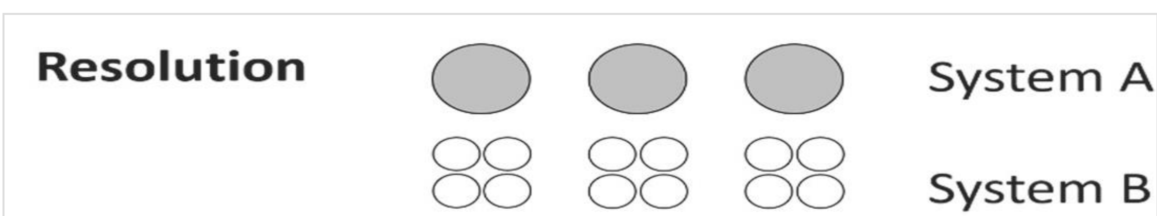


Fig. 6. Model concepts misaligned by resolution. [42]

The third major risk scenario is misaligned model structures. Structure refers to the grouping of one or more sub-concepts in describing a larger concept. These groupings of subcomponents may not mirror one another across multiple models. To complicate matters, sub-concepts may be included in the grouping of another major concepts in another model. In the Fig. 7 below, System A includes two entities, each with two descriptive components. Likewise, System B has two entities, each with two differing descriptive components, though some of those sub components have be swapped between major conceptual entities.

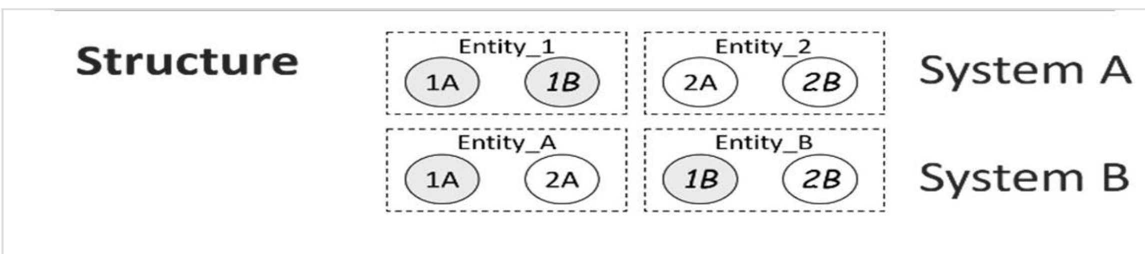


Fig. 7. Model concepts misaligned by structure. [42]

Beyond these “basic” risk scenarios are combinations of those scenarios which include 1) a misalignment of both scope and of resolution, 2) a misalignment of both scope and structure, 3) a misalignment of both resolution and structure, and 4) a misalignment of scope, resolution, and structure. Fig. 8 through Fig. 11 below graphically depict these scenarios. Integrating two models together will demonstrate a misalignment in at least one of these dimensions, and very likely in multiple dimensions.

In Fig. 8 below, the scenario where both scope and resolution are misaligned is depicted. System A may have any number of concepts describing its scope breadth—the figure shows three as an example. System B may share some non-zero number of major concepts with System A but replaces some number of System A's concepts with higher levels of detail. The concepts shared between Systems A and B is non-zero because if there were no overlap of the two models, the models would simply not be compatible with one another.

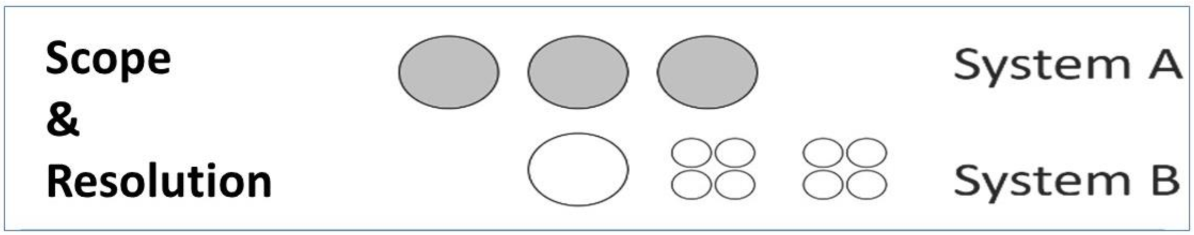


Fig. 8. Misaligned scope and resolution. [42]

Fig. 9 below depicts models that are conceptually misaligned in both scope and resolution. At least one of the two systems contain a major concept not included in the other system. In the example below, System A contains "concept 1" which has no corollary in System B while System B contains "concept 4" which has no mapping in System A. In the major concepts that are shared between the models, there is a mismatch of which sub-components are included in each major concepts' definition. It is possible that one or more sub-concepts may exist in one model with no mapping to the other model, as depicted in sub-component 2C.

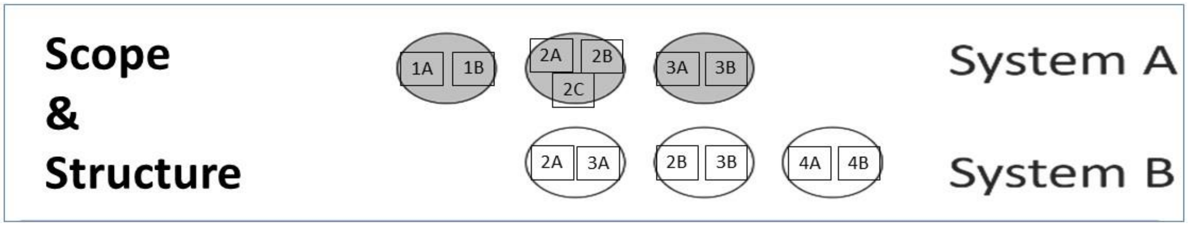


Fig. 9. Misaligned scope and structure. [42]

Fig. 10 below depicts the next major Risk scenario where two models are misaligned in both resolution and structure. Both modeling systems include the same major concepts, but at least one of the two models—in this case System B—includes greater detail in one or more of the concepts. Where major concepts are shared in each model, there may be different structures of supporting detail. In the example below, concept 2A moved from describing one major concept in System A to describing another major concept in System B. Likewise, concepts 1B and 3B describe different major concepts between the two models.

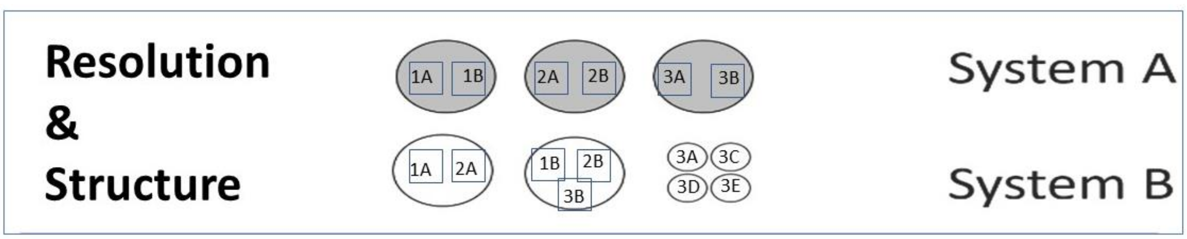


Fig. 10. Misaligned resolution and structure. [42]

The final Risk scenario is a misalignment across all three major definitions of misalignment – scope, resolution, and structure. Each model may have different major concepts from one another, supporting sub-concepts may be grouped differently in each model to describe different major concepts, and one model may have more detail in place of simplified assumptions in the other model. In practicality, this is the most likely risk scenario, where models have been developed and applied independently with different assumptions, different levels of detail, and different purposes—perhaps nuancedly different, but different nonetheless. Fig. 11 below depicts this complex misalignment where:

- System A contains concept 1 that has no matching component in System B for a misaligned resolution.
- System B contains concept 4 that has no matching component in System A for a misaligned resolution.
- System A's contains concept 2 with 3 elements. These elements are divided between concepts 2 and 4 in System B for a misaligned structure.
- System A contains concept 3 which is divided between concepts 2 and 4 in System B for a misaligned structure.
- System B contains concept 2 with 4 elements whereas System A contains concept 2 with only 3 elements for a misaligned resolution.

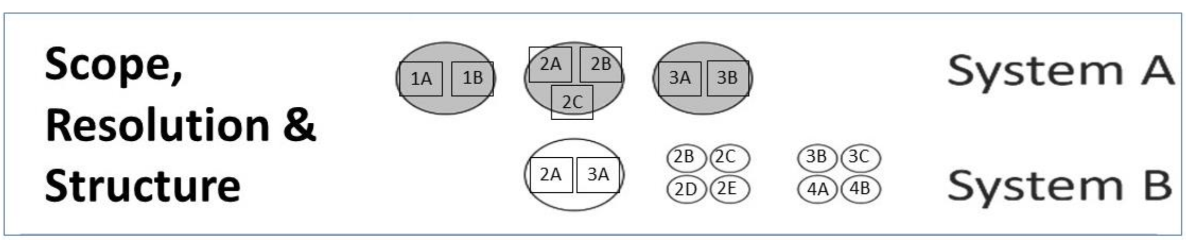


Fig. 11. Misaligned scope, resolution, and structure. [42]

4.2 Defining Model Alignments Across Three Axes

In all of these risk scenarios, the definitions of misalignment can apply to objects, to processes, and to relationships. So, it is possible, if not likely, that each of those three major categories of conceptual components will have different levels of misalignment. Thus, the Risk tuple needs to consider each dimension—Objects, Processes, and Relationships—independently of one another. The misalignment of any two models is the measure of their dissimilarity and can be calculated on a value from 0 to 1, where 0 means no alignment and 1 means perfect alignment, as will be shown.

4.2.1 Model Alignment for Objects

The conceptual element of Objects are those entities within a simulation with a distinct identity and persist during the course of the model's execution [1]. Objects are collections of attributes, and those attributes are what differ Objects from one another. Differing values within the same attribute distinguish similar Objects from one another. As an example, they could represent two vehicles in the same model with different levels of fuel remaining. Differing sets of attributes will distinguish different classes of objects from one another. With each Object being a collection of attributes, set notation can be used to define the mathematics of model alignment.

Let:

M_A indicate Model A

M_B indicate Model B

Further, let:

$M_{A,O}$ indicate the set of Objects in Model A

$M_{B,O}$ indicate the set of Objects in Model B

$M_{A, O(n)}$ indicate Object n in Model A

$M_{B, O(l)}$ indicate Object l in Model B

$M_{A, O(n), A}$ indicate the set of Attributes in Object n in Model A

$M_{B, O(n), A}$ indicate the set of Attributes in Object n in Model B

$M_{A, O(n), A(m)}$ indicate attribute m in Object n in Model A

$M_{B, O(l), A(k)}$ indicate attribute k in Object l in Model B

$M_{A, A}$ indicate the set of all Attributes across all Objects in Model A

$M_{B, A}$ indicate the set of all Attributes across all Objects in Model B

To compare models with one another for integration, the models' objects and their defining attributes are the first basis of comparison. To use the risk scenarios defined in the previous section, two models may differ in:

1) Scope
2) Resolution
3) Structure
4) A combination of scope and resolution
5) A combination of scope and structure
6) A combination of resolution and structure, or
7) A combination of scope, resolution, and structure.

The following subsections calculate the alignment of models for the previously defined seven risk scenarios. The alignment calculation is done on a model-to-model comparison.

4.2.1.1 Objects' Misaligned Scope

In the case of misaligned scope, it is expected that at least one of the input models has one or more Objects not contained in the other model. However, there is a subset of Objects that are common to both models and a superset of all Objects in both models. In this scenario, the Objects are presumed to be compatible with one another and the Attributes that define each Object are assumed to be the same. When this assumption is not true, there is also a misalignment of resolution or structure, which are addressed in other Risk scenarios. Therefore, the misalignment of Objects is binary – either the Objects in question are contained in both models, or they are not. In this simple scenario, and a direct comparison of each model's conceptual Object is made to the conceptual Objects of the other model. The value hierarchy for Objects in a misaligned scope then is shown in Table 17.

TABLE 17

VALUE HIERARCHY FOR CONCEPTUAL OBJECTS IN A MISALIGNED SCOPE SCENARIO

Definition	Value
Conceptual Object in Model A has an unambiguous mapping to a Conceptual Object in Model B. All attributes in the Object are consistent in both models with no additional or missing attributes.	1.00
Conceptual Object in Model A has no comparable conceptual Object in Model B	0.00

To make a comparison of the two models in the Objects dimension, then is an average of the values across the union of the sets of Objects in both Models A and B yields the overall alignment of the model.

Let O = total number of Object elements in both models. Mathematically,

$$O = M_{A,O(n)} \cup M_{B,O(l)} \tag{9}$$

Let T be the total number of Objects in the union of both models.

$T = |O|$

The function

$$\forall M_{A}, O_{(n)} \in O, \sum x \tag{10}$$

Where

$x = \begin{cases} 1, \text{if } M_{A,O} \exists \text{ O} \\ 0, \text{otherwise.} \end{cases}$

The final alignment value of Objects from Model A to Model B is the ratio of the sum of all common elements to total elements, or

$$A(Obj) = x \,/\, T \tag{11}$$

Where $A(Obj)$ is the alignment value of the Objects between models.

4.2.1.2 Objects' Misaligned Resolution

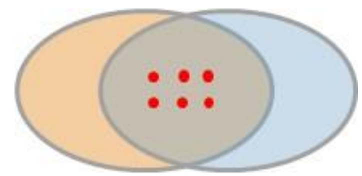
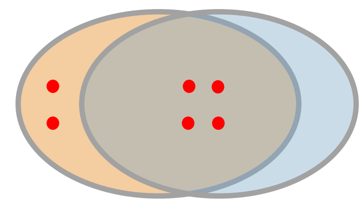
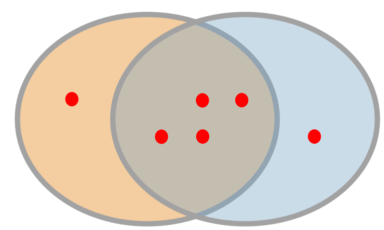
In the scenario of misaligned Resolution, one or more conceptual Objects contained in the one model is compared to a collection of Objects in the other model. The comparison is one to many, meaning that the second model has more than one Object in place of a single Object in the original model. To make a meaningful comparison between a larger or coarser Object to smaller or finer Objects, we need to decompose them into their constituent Attributes. Attributes are the defining qualities of Objects. The comparison of two models is not meant to explore the specific values of such Objects' Attributes, but rather the type of Attribute that is contained in each Object. In this scenario, it is unlikely that Attributes will be unambiguously mapped to one another. In order to make comparisons from an Attribute in one model to an Attribute in another model a value hierarchy will be required to evaluate the models' alignment with a deeper look at each models' objects' attributes. There are three permutations of misaligned resolution (Table 18).

$M_{A, O(n), A(m)}$ indicates attribute *m* in Object *n* in Model *A*

$M_{B, O(l), A(k)}$ indicates attribute *k* in Object *l* in Model *B*

TABLE 18

MODEL OBJECTS MISALIGNED BY RESOLUTION

Definition	Value	Description	Venn Diagram
The conceptual Object in Model A has all of its Attributes accounted for in the same conceptual Object in Model B. Model B has no Attributes beyond those captured in Model A	1.00	This is a perfect match where the contents of Model B can simply be used to replace the contents of Model A	
The conceptual Object in Model A has a subset of its Attributes accounted for in a set of Objects in Model B, but has Attributes not found in Model B. Model B has no Attributes beyond those captured in Model A	$\frac{(M_{A,O(n),A(m)}) \cap (M_{\mathrm{A,O(n),A(k)}})}{(M_{A,O(n),A(m)})}$	The shared set of attributes as a ratio to the contents of Model A	
Both Models A and B have attributes that are shared in the same conceptual Object and attributes that are unique.	$\frac{(M_{A,O(n),A(m)}) \cap (M_{\mathrm{A,O(n),A(k)}})}{(M_{A,O(n),A(m)})} \times \frac{(M_{A,O(n),A(m)}) \cap (M_{\mathrm{A,O(n),A(k)}})}{(M_{A,O(n),A(m)})}$	This comparison is the product of Model A's Object's Attributes that are shared in the entire set of Attributes and Model B's Object's Attributes that are shared in the entire set of Attributes	

To make a comparison of the two models in the entirety of the Objects dimension, then is an average of the values across the union of the sets of Objects in both Models A and B yields the overall alignment of the model in the Objects dimension. This simple case presumes that the number of Objects in Model A is the same as the number of Objects in Model B.

Recalling that in this scenario, both models have the same number of Objects,

Let N = the number of Objects in either model. $N = |M_A, o_{(n)}| = |M_B, o_{(l)}|$

The function

$\forall M_A, o \in N, \sum x$ sums the score of each individual Object in the collection

where *x* is determined from the tabulated formulae above.

The final alignment value of Objects from Model A to Model B is the ratio of the sum of all common elements to total elements, or

$A(Obj) = x / N$

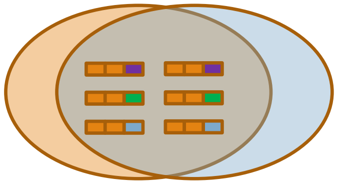
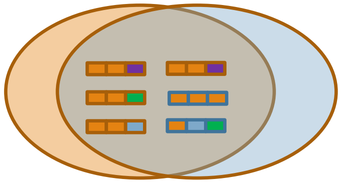
Where $A(Obj)$ is the alignment value of the Objects between models.

4.2.1.3 Objects Misaligned Structure

In the scenario of misaligned Structure, one or more Attributes of one or more conceptual Objects in a model are associated with different conceptual Objects in another model. In this particular scenario, it is assumed that all Attributes of all Objects are contained in both models but arranged differently in their descriptions of Objects than one another. More complex scenarios where there are different Objects or different Attributes are considered later in this dissertation. The misaligned structures are depicted in Table 19 below. What this means is that conceptual Objects while seemingly the same by name or gross description are different from one another, and the comparison of alignment needs to be made at each Object in the model to account for attributes that are found in different Objects in another model. To measure the alignment here, the Objects can no longer be considered a one for one match, and the individual attributes of each Object need to be considered as fundamental to the definition of the Object. Borrowing from the method presented by Wartik et al. in reference [15], each Objects' individual alignment must be considered against the Object of similar definition in the second model.

TABLE 19

MODEL OBJECTS MISALIGNED BY STRUCTURE

Definition	Value	Description	Venn Diagram
All Objects in Model A have all their Attributes in the same Objects in Model B	1.00	There is no misalignment of individual Objects and their descriptive Attributes	
Partial match. The total number of Attributes in an Object in Model A that is shared with the same Object in Model B divided by total number of Attributes used to define the Object across both Models	$\frac{(M_{A,O(n),A}) \cap (M_{B,O(n),A})}{(M_{A,O(n),A}) \cup (M_{B,O(n),A})}$	The Attributes that define an Object of one model exist in the description of an Object in the second model	

To make a comparison of the two models in the entirety of the Objects dimension, then is an average of the values across the union of the sets of Objects in both Models A and B yields the overall alignment of the model in the Objects dimension. This simple case presumes that the number of Objects in Model A is the same as the number of Objects in Model B.

Recalling that in this scenario, both models have the same number of Objects,

Let N = the number of Objects in either model. $N = |M_{A}, o_{(n)}| = |M_{B}, o_{(l)}|$

The function

$\forall\, M_{A}, o \in N, \sum x$ sums the score of each individual Object in the collection

where x is determined from the tabulated formulae above.

The final alignment value of Objects from Model A to Model B is the ratio of the sum of all common elements to total elements, or

$A(Obj) = x / N$

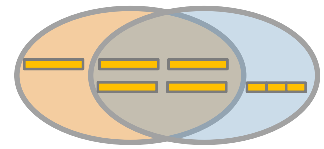
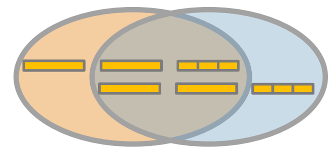
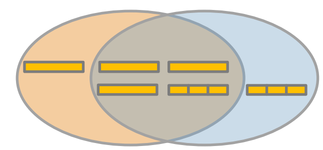
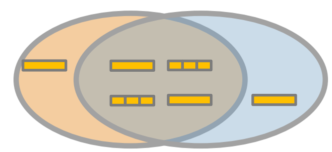
Where $A(Obj)$ is the alignment value of the Objects between models.

4.2.1.4 Objects Misaligned in both Scope and Resolution

A misalignment due to both misaligned scope and resolution is a case where there are not only unique Objects in each model, but the shared Objects differ in the level of detail that defines them. This case must be examined at the Attribute level, since the Attributes account for the difference in detail between the Objects in each of the models. As presented earlier, misaligned scope means that one or both of the models contain concept with no concept element in the other model. Misaligned resolution is where one or more Objects have more details in one model than in the other. The combination of these two misalignments simply means that one model may contain an Attribute or Attributes in one or more Objects that cannot be aligned to an Object in the other model and permutations of this misalignment are depicted in Table 20. Furthermore, those additional Attributes are not aligned to Attributes in other Objects, which would account for a misaligned structure between the models.

TABLE 20

MODEL OBJECTS MISALIGNED BY SCOPE AND RESOLUTION

Definition	Value	Description	Venn Diagram
All Objects shared by Model A and Model B are identical in their attributes' definitions	1.00	There is no misalignment of individual Objects and their component Attributes	
The Object shared by Model A and Model B differ in the Attributes assigned to them (A has more Attributes than B)	$\frac{(M_{A,O(n),A}) \cap (M_{B,O\ (n),A})}{(M_{A,O(n),A}) \cup (M_{B,O\ (n),A})}$	One or more Objects in Model A have more detail than a matching Object in Model B. There are Objects in one or both models that do not map to Objects in the other model.	
The Objects shared by M_a and M_b differ in the Attributes assigned to them (Model B has more Attributes than Model A)	$\frac{(M_{A,O(n),A}) \cup (M_{B,O\ (n),A})}{(M_{A,O(n),A}) + \frac{(M_{A,O(n),A}) \cap (M_{B,O\ (n),A})}{(M_{A,O(n),A})}}$	One or more Objects in Model B has more detail than a matching Object in Model A	
Both M_a's and M_b's Objects have unique Attributes	$M_{A,O(n),A} \div (M_{A,O(n),A} \cap M_{B,O(n),A})$	Each Model has one or more Objects with more detail than its corresponding Object in in the other Model. It is a ratio of Model A's Objects' Attributes to the union of total Attributes across both models that define that Object	

To make a comparison of the two models in the entirety of the Objects dimension, then is an average of the values across the union of the sets of Objects in both Models A and B yields the overall alignment of the model in the Objects dimension. This simple case presumes that the number of Objects in Model A is the same as the number of Objects in Model B.

In this scenario, one model has more Objects than the other model, and the calculation must be made from the perspective of one model.

Let N = the number of Objects in Model A. $N = |M_A, o_{(n)}|$

The function

$\forall M_A, o \in N, \sum x$ sums the score of each individual Object in the collection

where x is determined from the tabulated formulae above.

The final alignment value of Objects from Model A to Model B is the ratio of the sum of all common elements to total elements, or

$A(Obj) = x / N$

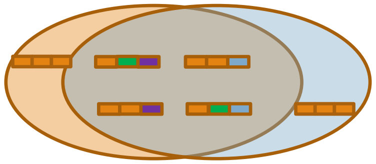
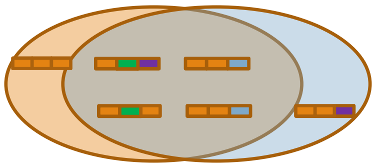
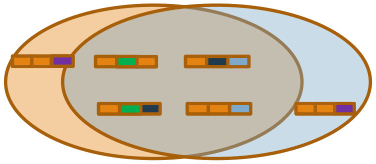
Where $A(Obj)$ is the alignment value of the Objects between models.

4.2.1.5 Objects Misaligned in both Scope and Structure

A misalignment due to both misaligned scope and structure is a case where at least one of the two models in question offers more Objects than the other model, meaning a misalignment of scope. Additionally, the Objects that are shared between the two models include the same set of Attributes, but in different Objects' definitions, meaning a misalignment of structure. Permutations of this misalignment are depicted in Table 21 below.

TABLE 21

MODEL OBJECTS MISALIGNED BY SCOPE AND STRUCTURE

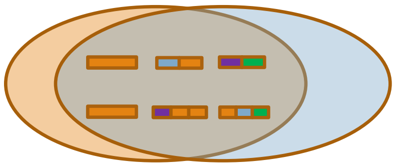
Definition	Value	Description	Venn Diagram
Attributes of Objects within the intersected space are all accounted for within the intersected space, albeit in differing Objects	$\frac{(M_{A,O} \cap M_{B,O})}{(M_{A,O} \cup M_{B,O}) + (M_{A,O} \cap M_{B,O})} + \frac{(M_{A,O} \cap M_{B,O'})}{(M_{A,O} \cup M_{B,O'}) + (M_{A,O} \cap M_{B,O'})}$	The attributes of Models' shared Objects are included among other shared Objects	
Attributes of Objects within the intersected space are all accounted for, but at least one Attribute is outside the intersected space. Additional terms are for specific attributes outside the intersected Objects' space.	$\frac{(M_{A,O} \cap M_{B,O})}{(M_{A,O} \cup M_{B,O}) + (M_{A,O} \cap M_{B,O})} + \frac{(M_{A,O} \cap M_{B,O'})}{(M_{A,O} \cup M_{B,O'}) + (M_{A,O} \cap M_{B,O'})} + \frac{(M_{B,O'\|A(m)})}{(M_{A,O\|A(m)} \cup M_{B,O'\|A(m)}) + (M_{B,O\|A(m)})}$	Among the shared Objects, there are no unique Attributes, though the defining Attributes are found outside the shared space of Objects; the Attributes are used in the definition of Objects not shared between the Models	
Compound; Different Objects outside the intersected space have Attributes misaligned	$\frac{(M_{A,O} \cap M_{B,O})}{(M_{A,O} \cup M_{B,O}) + (M_{A,O} \cap M_{B,O})} + \frac{(M_{A,O} \cap M_{B,O})}{(M_{A,O} \cup M_{B,O}) + (M_{A,O} \cap M_{B,O})} + \frac{(M_{A,O} \cap M_{B,O'})}{(M_{A,O} \cup M_{B,O'}) + (M_{A,O} \cap M_{B,O'})} + \frac{(M_{B,O'\|A(m)})}{(M_{A,O\|A(m)} \cup M_{B,O'\|A(m)}) + (M_{B,O\|A(m)})} + \frac{(M_{A,O'\|A(m)} \cup M_{B,O'\|A(m)})}{(M_{A,O\|A(m)}) + (M_{B,O\|A(m)})}$	Attributes that are used to define Objects unique to one of the Models are found within the definition of Objects that are shared between the Models.	

4.2.1.6 Objects Misaligned in both Resolution and Structure

In this scenario, misalignment occurs when an Object or multiple Objects in one model have additional Attributes that define them in another model, causing a misalignment of the models' resolutions. Additionally, the Attributes defining the Object or Objects from one model are found in the definitions of different Objects in the second model. Permutations of this misalignment are depicted in Table 22 below.

TABLE 22

MODEL OBJECT MISALIGNED BY RESOLUTION AND STRUCTURE

Definition	Value	Description	Venn Diagram
Attributes from multiple low -resolution Objects are contained in multiple Objects of higher resolution	$\frac{(M_{A,O} \cap M_{B,O})}{(M_{A,O} \cup M_{B,O}) + (M_{A,O} \cap M_{B,O})} + \frac{(M_{A,O} \cap M_{B,O'\|A(m)})}{(M_{A,O} \cup M_{B,O'\|A(m)}) + (M_{A,O} \cap M_{B,O'\|A(m)})}$	The Attributes that define Objects in one Model are used to define different Objects in the other Model	

To make a comparison of the two models in the entirety of the Objects dimension, then is an average of the values across the union of the sets of Objects in both Models A and B yields the overall alignment of the model in the Objects dimension. This simple case presumes that the number of Objects in Model A is the same as the number of Objects in Model B.

In this scenario, one model has more Objects than the other model, and the calculation must be made from the perspective of one model.

Let N = the number of Objects in Model A. $N = |M_{A}, o_{(n)}|$

The function

$\forall M_{A}, o \in N, \sum x$ sums the score of each individual Object in the collection

where x is determined from the tabulated formulae above.

The final alignment value of Objects from Model A to Model B is the ratio of the sum of all common elements to total elements, or

$A(Obj) = x / N$

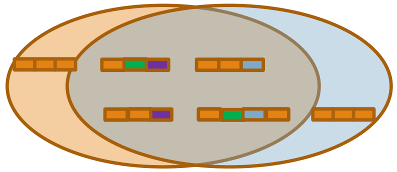
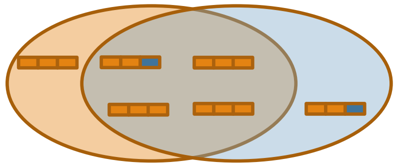
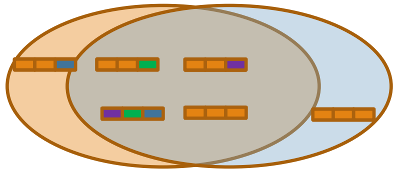
Where $A(Obj)$ is the alignment value of the Objects between models.

4.2.1.7 Objects Misaligned in Scope, Resolution, and Structure

In the final risk scenario, a misalignment occurs between model conceptual Objects due to differences in the scope of the Objects, their resolution, and their structure. As in the case of scope misalignment, Objects in one model include Attributes that are not found in the second model. As in misalignment of resolution, comparable Objects between the models will have different numbers of Attributes that define them. And as in the case where models differ in resolution, the contained Objects will have similar Attributes, but as part of the definition of different Objects in each model. The combination of these misalignments is the basis of this last scenario of misalignments. Table 23 below depicts the alignment calculations for this scenario.

TABLE 23

MODELS OBJECTS MISALIGNED BY SCOPE, RESOLUTION, AND STRUCTURE

Definition	Value	Description	Venn Diagram
All Objects in Model A that are shared between Models A and B have all their Attributes contained within the intersected space	$\frac{(M_{A,O} \cap M_{B,O}) + (M_{A,O} \cap M_{B,O'\mid A(m)})}{(M_{A,O} \cup M_{B,O'\mid A(m)}) + (M_{A,O} \cap M_{B,O})}$	Attributes used to define an Object in one Model are also used to define an Object in another model, albeit in different Objects	
One or more Objects outside the intersection of Models and B have one or more Attributes within an Object contained in both models	$\frac{(M_{A,O} \cap M_{B,O}) + (M_{A,O} \cap M_{B,O'\mid A(m)})}{(M_{A,O} \cup M_{B,O'\mid A(m)}) + (M_{A,O} \cap M_{B,O})} + \frac{(M_{B,O'\mid A(m)})}{(M_{A,O\mid A(m)} \cup M_{B,O'\mid A(m)})}$	An Object's Attribute in one Model is used to define a different and unique Object in another Model	
One or more Objects outside the intersection of Models and B have one or more Attributes within a shared Object as well as Attributes within the intersected space ascribed to different Objects	$\frac{(M_{A,O'\mid A(m)} \cap M_{B,O})}{(\sum M_{B,O\mid A(l) \in (M_{A,O'})})}$ x $\frac{((M_{A,O'\mid Attr}) - (M_{B,O}))}{(M_{A,O'})}$	An Object that is unique to one Model has an Attribute used to define an Object that is similar among the two Models.	

To make a comparison of the two models in the entirety of the Objects dimension, then is an average of the values across the union of the sets of Objects in both Models A and B yields the overall alignment of the model in the Objects dimension. This simple case presumes that the number of Objects in Model A is the same as the number of Objects in Model B.

In this scenario, one model has more Objects than the other model, and the calculation must be made from the perspective of one model.

Let N = the number of Objects in Model A. $N = |M_{A}, O_{(n)}|$

The function

$\forall M_{A}, O \in N, \sum x$ sums the score of each individual Object in the collection

where x is determined from the tabulated formulae above.

The final alignment value of Objects from Model A to Model B is the ratio of the sum of all common elements to total elements, or

$A(Obj) = x / N$

Where $A(Obj)$ is the alignment value of the Objects between model.

4.2.2 Model Alignment for Processes

The next major axes of model conceptual elements are Processes. Processes are the dynamic conceptual components of a model and represent changes in the models' states. Process also capture the nature of cause and effect in a model. Processes are collections of characteristics, and those specific characteristics are what differ Processes from one another. Differing values within the same characteristic distinguish similar Processes from one another. As an example, they could represent two vehicles in the same model with different levels of fuel remaining. Differing sets of characteristics will distinguish different classes of Processes from one another. With each Process being a collection of characteristics, set notation can be used to define the mathematics of model alignment.

As before, let:

M_A indicate Model *A*

M_B indicate Model *B*

Further, let:

$M_{A, P}$ indicate the set of Processes in Model *A*

$M_{B, P}$ indicate the set of Processes in Model *B*

$M_{A, P(n)}$ indicate Process *n* in Model *A*

$M_{B, P(l)}$ indicate Process *l* in Model *B*

$M_{A, P(n), A}$ indicate the set of characteristics in Process *n* in Model *A*

$M_{B, P(n), A}$ indicate the set of characteristics in Process *n* in Model *B*

$M_{A, P(n), C(m)}$ indicate characteristic *m* in Process *n* in Model *A*

$M_{B, P(l), C(k)}$ indicate characteristic *k* in Process *l* in Model *B*

$M_{A, C}$ indicate the set of all Characteristics across all Processes in Model *A*

$M_{B, C}$ indicate the set of all Characteristics across all Processes in Model *B*

To compare models with one another for integration, the models' Processes and their defining characteristics are the first basis of comparison. To use the risk scenarios defined in the previous section, two models may differ in:

1) Scope
2) Resolution
3) Structure
4) A combination of scope and resolution

5) A combination of scope and structure
6) A combination of resolution and structure, or
7) A combination of scope, resolution, and structure.

Similar to the alignment of objects, the following subsections calculate the alignment of models with respect to both their shared and divergent Processes for the previously defined seven risk scenarios.

4.2.2.1 Processes' Misaligned Scope

In the case of misaligned scope, it is expected that at least one of the input models has one or more Processes not contained in the other model. However, there is a subset of Processes that are common to both models and a superset of all Processes in both models. In this scenario, the Processes are presumed to be compatible with one another and the Characteristics that define each Process are assumed to be the same. When this assumption is not true, there is also a misalignment of resolution or structure, which are addressed in other Risk scenarios. Therefore, the misalignment of Processes is binary—either the Processes in question are contained in both models, or they are not. In this simple scenario, and a direct comparison of each model's conceptual Process is made to the conceptual Processes of the other model. The value hierarchy for Processes in a misaligned scope then is as shown in Table 24.

TABLE 24

VALUE HIERARCHY FOR CONCEPTUAL PROCESSES IN A MISALIGNED SCOPE SCENARIO

Definition	Value
Conceptual Process in Model A has an unambiguous mapping to a Conceptual Process in Model B. All characteristics in the Process are consistent in both models with no additional or missing characteristics.	1.00
Conceptual Process in Model A has no comparable conceptual Process in Model B	0.00

To make a comparison of the two models in the Processes dimension, then is an average of the values across the union of the sets of Processes in both Models A and B yields the overall alignment of the model.

Let P = total number of Process elements in both models. Mathematically,

$$P = M_{\mathrm{A,\,P(n)}} \cup M_{\mathrm{B,\,P(l)}} \tag{12}$$

Let T be the total number of Processes in the union of both models.

$T = |P|$

The function

$$\forall\, M_{\mathrm{A}}, P_{(\mathrm{n})} \in P, \sum x \tag{13}$$

Where

$$x = \begin{cases} 1, \text{if } M_{\mathrm{A,P}} \exists\, 0 \\ 0, \text{otherwise.} \end{cases}$$

The final alignment value of Processes from Model A to Model B is the ratio of the sum of all common elements to total elements, or

$$A(Proc) = x / T \tag{14}$$

Where $A(Proc)$ is the alignment value of the Processes between models.

4.2.2.2 Processes' Misaligned Resolution

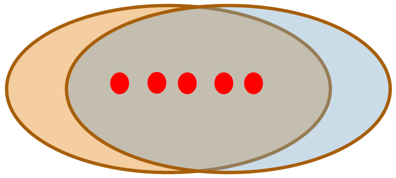
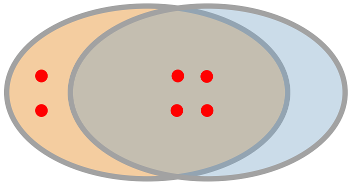
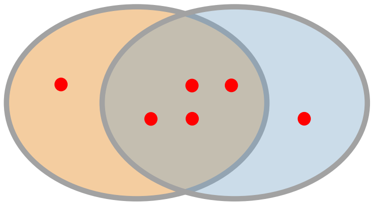
In the scenario of misaligned Resolution, one or more conceptual Processes contained in the one model is compared to a collection of Processes in the other model. The comparison is one to many, meaning that the second model has more than one Process in place of a single Process in the original model. To make a meaningful comparison between a larger or coarser Process to smaller or finer Processes, we need to decompose them into their constituent Characteristics. Characteristics are the defining qualities of Processes. The comparison of two models is not meant to explore the specific values of such Processes' Characteristics, but rather the type of Characteristic that is contained in each Process. In this scenario, it is unlikely that Characteristics will be unambiguously mapped to one another. In order to make comparisons from a Characteristic in one model to a Characteristic in another model a value hierarchy will be required to evaluate the models' alignment with a deeper look at each models' Processes' characteristics. There are three permutations of misaligned resolution, depicted in Table 25.

$M_{A, P(n), C(m)}$ indicates characteristic *m* in Process *n* in Model *A*

$M_{B, P(l), C(k)}$ indicates characteristic *k* in Process *l* in Model *B*

TABLE 25

PROCESSES MISALIGNED IN RESOLUTION

Definition	Value	Description	Venn Diagram
The conceptual Process in Model A has all of its Characteristics accounted for in the same conceptual Process in Model B. Model B has no Characteristics beyond those captured in Model A	1.00	This is a perfect match where the contents of Model B can simply be used to replace the contents of Model A	
The conceptual Process in Model A has a subset of its Characteristics accounted for in a set of Processes in Model B, but has Characteristics not found in Model B. Model B has no Characteristics beyond those captured in Model A	$\frac{(M_{A,P(n),C(m)}) \cap (M_{B,P(n),C(k)})}{(M_{A,P(n),C(m)})}$	The shared set of characteristics as a ratio to the contents of Model A	
Both Models A and B have characteristics that are shared in the same conceptual Process and characteristics that are unique.	$\frac{(M_{A,P(n),C(m)}) \cap (M_{\mathrm{B,P(n),CA(k)}})}{(M_{A,P(n),C(m)})} \times$ $\frac{(M_{A,P(n),C(m)}) \cap (M_{B,P(n),C(k)})}{(M_{B,P(n),C(k)})}$	This comparison is the product of Model A's Process's Characteristics that are shared in the entire set of Characteristics and Model B's Process's Characteristics that are shared in the entire set of Characteristics	

To make a comparison of the two models in the entirety of the Processes dimension, then is an average of the values across the union of the sets of Processes in both Models A and B yields the overall alignment of the model in the Processes dimension. This simple case presumes that the number of Processes in Model A is the same as the number of Processes in Model B.

Recalling that in this scenario, both models have the same number of Processes,

Let N = the number of Processes in either model. $N = |M_{A}, P_{(n)}| = |M_{B}, P_{(l)}|$

The function

$\forall M_{A}, P \in N, \sum x$ sums the score of each individual Process in the collection

where x is determined from the tabulated formulae above.

The final alignment value of Processes from Model A to Model B is the ratio of the sum of all common elements to total elements, or

$A(Proc) = x / N$

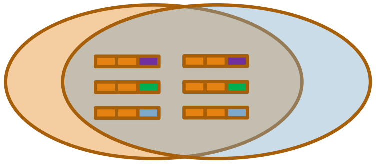
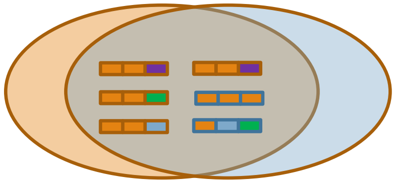
Where $A(Proc)$ is the alignment value of the Processes between models.

4.2.2.3 Processes Misaligned Structure

In the scenario of misaligned Structure, one or more Characteristics of one or more conceptual Processes in a model are associated with different conceptual Processes in another model. In this particular scenario, it is assumed that all Characteristics of all Processes are contained in both models, but arranged differently in their descriptions of Processes than one another. More complex scenarios where there are different Processes or different Characteristics are considered later in this dissertation. What this means is that conceptual Processes while seemingly the same by name or gross description are different from one another, and the comparison of alignment needs to be made at each Process in the model to account for characteristics that are found in different Processes in another model. To measure the alignment here, the Processes can no longer be considered a one-for-one match, and the individual characteristics of each Process need to be considered as fundamental to the definition of the Process. Borrowing from the method presented in reference [15], each Processes' individual alignment must be considered against the Process of similar definition in the second model. The permutations and calculations of this misalignment are depicted below in Table 26.

TABLE 26

PROCESSES MISALIGNED BY STRUCTURE

Definition	Value	Description	Venn Diagram
All Processes in Model A have all their Characteristics in the same Processes in Model B	1.00	There is no misalignment of individual Processes and their descriptive Characteristics	
Partial match. The total number of Characteristics in a Process in Model A that is shared with the same Process in Model B divided by total number of Characteristics used to define the Process across both Models	$\frac{(M_{A,P(n),C}) \cap (M_{B,P(n),C})}{(M_{A,P(n),C}) \cup (M_{B,P(n),C})}$	The Characteristics that define a Process of one model exist in the description of a Process in the second model	

To make a comparison of the two models in the entirety of the Processes dimension, then is an average of the values across the union of the sets of Processes in both Models A and B yields the overall alignment of the model in the Processes dimension. This simple case presumes that the number of Processes in Model A is the same as the number of Processes in Model B.

Recalling that in this scenario, both models have the same number of Processes,

Let N = the number of Processes in either model. $N = |M_{A,P(n)}| = |M_{B,P(l)}|$N

The function

$\forall M_{A,P} \in N, \sum x$ sums the score of each individual Process in the collection

where x is determined from the tabulated formulae above.

The final alignment value of Processes from Model A to Model B is the ratio of the sum of all common elements to total elements, or

$A(Proc) = x / N$

Where $A(Proc)$ is the alignment value of the Processes between models.

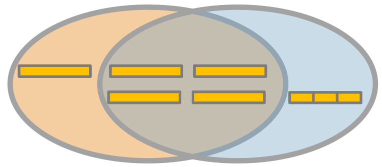
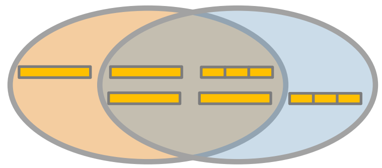
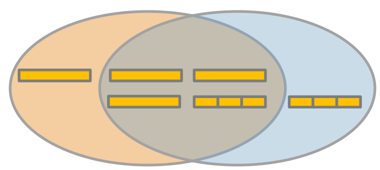
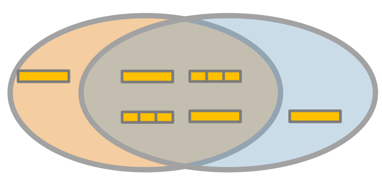
4.2.2.4 Processes Misaligned in both Scope and Resolution

A misalignment due to both misaligned scope and resolution is a case where there are not only unique Processes in each model, but the shared Processes differ in the level of detail that defines them. This case must be examined at the Characteristic level, since the Characteristics account for the difference in detail between the Processes in each of the models. As presented earlier, misaligned scope means that one or both of the models contain concept with no concept element in the other model. Misaligned resolution is where one or more Processes have more details in one model than in the other. The combination of these two misalignments simply means that one model may contain a Characteristic or Characteristics in one or more Processes that cannot be aligned to a Process in the other model. Furthermore, those additional

Characteristics are not aligned to Characteristics in other Processes, which would account for a misaligned structure between the models. This misalignment is depicted in Table 27 below.

TABLE 27

PROCESSES MISALIGNED IN SCOPE AND RESOLUTION

Definition	Value	Description	Venn Diagram
All Processes shared by Model A and Model B are identical in their characteristics' definitions	1.00	There is no misalignment of individual Processes and their component Characteristics	
The Process shared by Model A and Model B differ in the Characteristics assigned to them (A has more Characteristics than B)	$\frac{(M_{A,P(n),C}) \cap (M_{B,P\,(n),C})}{(M_{A,P(n),C}) \cup (M_{B,P\,(n),C})}$	One or more Processes in Model A have more detail than a matching Process in Model B. There are Processes in one or both models that do not map to Processes in the other model.	
The Processes shared by M_a and M_b differ in the Characteristics assigned to them (Model B has more Characteristics than Model A)	$\frac{(M_{A,P(n),C}) \cap (M_{B,P\,(n),C})}{(M_{A,P(n),C}) + \frac{(M_{A,P(n),C}) \cup (M_{B,P\,(n),C})}{(M_{A,P(n),C})}}$	One or more Processes in Model B has more detail than a matching Process in Model A	
Both M_a's and M_b's Processes have unique Characteristics	$\frac{M_{A,P(n),C}}{(M_{A,P(n),C}) \cup (M_{B,P\,(n),C})}$	Each Model has one or more Processes with more detail than its corresponding Process in in the other Model. It is a ratio of Model A's Processes' Characteristics to the union of total Characteristics across both models that define that Process	

To make a comparison of the two models in the entirety of the Processes dimension, then is an average of the values across the union of the sets of Processes in both Models A and B yields the overall alignment of the model in the Processes dimension. This simple case presumes that the number of Processes in Model A is the same as the number of Processes in Model B.

In this scenario, one model has more Processes than the other model, and the calculation must be made from the perspective of one model.

Let N = the number of Processes in Model A. $N = |M_{A,P(n)}|$

The function

$\forall M_{A,P} \in N, \sum x$ sums the score of each individual Process in the collection

where x is determined from the tabulated formulae above.

The final alignment value of Processes from Model A to Model B is the ratio of the sum of all common elements to total elements, or

$A(Proc) = x / N$

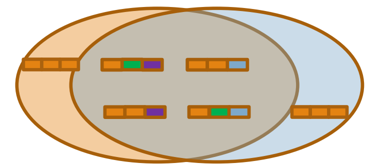
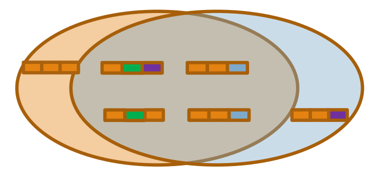
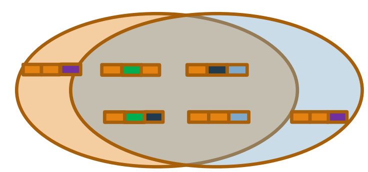
Where $A(Proc)$ is the alignment value of the Processes between models.

4.2.2.5 Processes Misaligned in both Scope and Structure

A misalignment due to both misaligned scope and structure is a case where at least one of the two models in question offers more Processes than the other model, meaning a misalignment of scope. Additionally, the Processes that are shared between the two models include the same set of Characteristics, but in different Processes' definitions, meaning a misalignment of structure. The permutations of this misalignment are depicted in Table 28.

TABLE 28

PROCESSES MISALIGNED IN SCOPE AND STRUCTURE

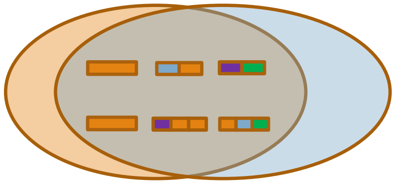
Definition	Value	Description	Venn Diagram
Characteristics of Processes within the intersected space are all accounted for within the intersected space, albeit in differing Processes	$\frac{(M_{A,C} \cap M_{B,C})}{(M_{A,C} \cup M_{B,C}) + (M_{A,C} \cap M_{B,C})} + \frac{(M_{A,C} \cap M_{B,C'})}{(M_{A,C} \cup M_{B,C'}) + (M_{A,C} \cap M_{B,C'})}$	Characteristics of shared Processes in one Model are found in different, but still shared Processes in the other Model	
Characteristics of Processes within the intersected space are all accounted for, but at least one Characteristic is outside the intersected space.	$\frac{(M_{A,P} \cap M_{B,P})}{(M_{A,P} \cup M_{B,P}) + (M_{A,P} \cap M_{B,P})} + \frac{(M_{A,P} \cap M_{B,P'})}{(M_{A,P} \cup M_{B,P'}) + (M_{A,P} \cap M_{B,P'})} + \frac{(M_{B,P'\|C(m)})}{(M_{A,P\|C(m)} \cup M_{B,P'\|C(m)}) + (M_{B,P\|C(m)})}$	The Characteristics of a Process shared between the Model are found in different Processes in the other Model that are not necessarily similar Processes to the first Model.	
Compound; Different Processes outside the intersected space have Characteristics misaligned	$\frac{(M_{A,P} \cap M_{B,P})}{(M_{A,P} \cup M_{B,P}) + (M_{A,P} \cap M_{B,P})} + \frac{(M_{A,P} \cap M_{B,P'})}{(M_{A,P} \cup M_{B,P'}) + (M_{A,P} \cap M_{B,P'})} + \frac{(M_{B,P'\|C(m)})}{(M_{A,P\|C(m)} \cup M_{B,P'\|C(m)}) + (M_{B,P\|C(m)})} + \frac{(M_{A,P'\|C(m)} \cup M_{B,P'\|C(m)})}{(M_{A,P\|C(m)}) + (M_{B,P\|C(m)})}$	The Characteristics of Processes that are unique to a Model are found within the Characteristics of a Process shared between the Models	

4.2.2.6 Processes Misaligned in both Resolution and Structure

In this scenario, misalignment occurs when a Process or multiple Processes in one model have additional Characteristics that define them in another model, causing a misalignment of the models' resolutions. Additionally, the Characteristics defining the Process or Processes from one model are found in the definitions of different Processes in the second model. This misalignment is depicted in Table 29 below.

TABLE 29

PROCESSES MISALIGNED IN RESOLUTION AND STRUCTURE

Definition	Value	Description	Venn Diagram
Characteristics from multiple low - resolution Processes are contained in multiple Processes of higher resolution	$\frac{(M_{A,\,P} \cap M_{B,\,P})}{(M_{A,\,P} \cup M_{B,\,P}) + (M_{A,\,P} \cap M_{B,\,P})} + \frac{(M_{A,\,P} \cap M_{B,\,P'\|\,C(m)})}{(M_{A,\,\mathrm{P}} \cup M_{B,\,\mathrm{P}'\|C(m)}) + (M_{A,\,\mathrm{P}} \cap M_{B,\,\mathrm{P}'\|C(m)})}$	The Characteristics that define Processes in one Model are used to define different Processes in the other Model	

To make a comparison of the two models in the entirety of the Processes dimension, then is an average of the values across the union of the sets of Processes in both Models A and B yields the overall alignment of the model in the Processes dimension. This simple case presumes that the number of Processes in Model A is the same as the number of Processes in Model B.

In this scenario, one model has more Processes than the other model, and the calculation must be made from the perspective of one model.

Let N = the number of Processes in Model A. $N = |M_{A,P(n)}|$

The function

$\forall M_{A,P} \in N, \sum x$ sums the score of each individual Process in the collection

where x is determined from the tabulated formulae above.

The final alignment value of Processes from Model A to Model B is the ratio of the sum of all common elements to total elements, or

$A(Proc) = x / N$

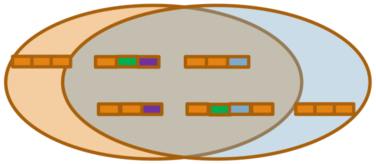
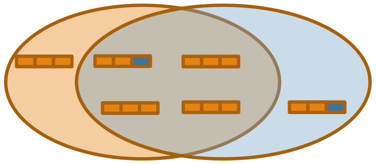
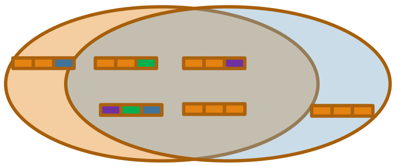
Where $A(Proc)$ is the alignment value of the Processes between models.

4.2.2.7 Processes misaligned in Scope, Resolution, and Structure

In the final risk scenario, a misalignment occurs between model conceptual Processes due to differences in the scope of the Processes, their resolution, and their structure. As in the case of scope misalignment, Processes in one model include Characteristics that are not found in the second model. As in misalignment of resolution, comparable Processes between the models will have different numbers of Characteristics that define them. And as in the case where models differ in resolution, the contained Processes will have similar Characteristics, but as part of the definition of different Processes in each model. The combination of these misalignments is the basis of this last scenario of misalignments. Table 30 below depicts the alignment calculations for this scenario.

TABLE 30

PROCESSES DIFFERING BY SCOPE, RESOLUTION, AND STRUCTURE

Definition	Value	Description	Venn Diagram
All Processes in Model A that are shared between Models A and B have all their Characteristics contained within the intersected space	$\frac{(M_{A,P} \cap M_{B,P}) + (M_{A,P} \cap M_{B,P'\mid C(m)})}{(M_{A,P} \cup M_{B,P'\mid C(m)}) + (M_{A,P} \cap M_{B,P})}$	The Characteristics that define shared Processes are found in different shared Processes among the two Models	
One or more Processes outside the intersection of Models and B have one or more Characteristics within a Process contained in both models	$\frac{(M_{A,P} \cap M_{B,P}) + (M_{A,P} \cap M_{B,P'\mid C(m)})}{(M_{A,P} \cup M_{B,P'\mid C(m)}) + (M_{A,P} \cap M_{B,P})} + \frac{(M_{B,P'\mid C(m)})}{(M_{A,P\mid C(m)} \cup M_{B,P'\mid C(m)})}$	Characteristics that define a unique Process of one Model are found within the Characteristics of a Process shared between the Models	
One or more Processes outside the intersection of Models and B have one or more Characteristics within a shared Process as well as Characteristics within the intersected space ascribed to different Processes	$\frac{(M_{A,P'\mid C(m)} \cap M_{B,P})}{(M_{B,P\mid C(l) \in (M_{A,P'})})} \times \frac{((M_{A,P'\mid C}) - (M_{B,P)}))}{(M_{A,C'})}$	Characteristics of unique Processes of a Model are found within Model are found within a shared Process of both Models. Characteristics of shared Processes are found in the definition of a different Process in the other Model.	

To make a comparison of the two models in the entirety of the Processes dimension, then is an average of the values across the union of the sets of Processes in both Models A and B yields the overall alignment of the model in the Processes dimension. This simple case presumes that the number of Processes in Model A is the same as the number of Processes in Model B.

In this scenario, one model has more Processes than the other model, and the calculation must be made from the perspective of one model.

Let N = the number of Processes in Model A. $N = |M_{A,P(n)}|$

The function

$\forall M_{A,P} \in N, \sum x$ sums the score of each individual Process in the collection

where x is determined from the tabulated formulae above.

The final alignment value of Processes from Model A to Model B is the ratio of the sum of all common elements to total elements, or

$A(Proc) = x / N$

Where $A(Proc)$ is the alignment value of the Processes between models.

4.2.3 Model Alignment for Relationships

Relationships are the third category of model conceptual components that present opportunities for models to misalign with one another. They are unique, however, in that they cannot be treated independently of the other conceptual components. Relationships are dependent upon Objects and Processes that are present in the model. Relationships link two or more other conceptual components together; be they Objects or Processes. While Relationships are decomposed by the Rules that govern the relationship [1], those rules and relationships are defined by and define the linkage between other conceptual components.

To compare conceptual Relationships between two models first means that all components linked by a Relationship need to be the same in both models. A Relationship that does not link the same Objects, Processes and their component Attributes and Characteristics cannot be compared to a Relationship with different linkages. This drastically simplifies the set of Relationships that need to be considered between two models as primarily a binary decision. That is, either the Relationship is consistent between models or it is not. A value of 0 would mean that the Relationship is not consistently present in both models whereas a value of 1 would mean that the relationship is present in both models.

The overall alignment score between one model and another is the average overall Relationships in each model as either a zero or a one. As before, the comparison from one model to another is not a commutative one—the count of Relationships is dependent upon which model acts as the frame of reference for the comparison, and the denominator value of this average would change from one model to another.

4.3 Defining Consequences

In order for there to be a risk to the quality of the decision or decisions made by using models in concert with one another, there must be a negative consequence to doing so. Such a consequence is most directly measurable from changes in the outputs of the model, specifically the measures of effectiveness (MOEs) and measures of performance (MOPs) that are used as the basis of the decision. Collectively, these metrics are measures of merit (MOMs). Changes in the MOMs can result from the structural differences of the models and can alert savvy analysts and decision makers to issues that might warrant additional scrutiny.

MOEs are those metrics that are directly for the decision being made, and as the name suggests, indicates how effectively the system or systems under scrutiny meet their objectives.

MOPs can be likened to efficiency, not a measure of meeting intent, but how well the system meets that intent. MOPs can provide additional insight to the problem space by measuring the utility or efficiency of the system. In any given modeling and simulation analysis, or even engineering analysis, the measures themselves may be few or may be many, depending on the complexity of both the system and the decision to be made. For a simulation to be useful for its intended purpose, it needs to provide measures that are meaningful in the tradespace of the decision, and decision makers must have some confidence in its validity. Where there are changes in model outputs due to model integration, the validity of the model composition is in need of further consideration as well. As discussed earlier, face validity is a more informal method of model validation, and demonstrable changes in model outputs can trigger at least an informal review of model outputs, such as a face validation.

Measures of merit are a specific value or a calculation of a model's state. A model's state is the values across Object's Attributes that are germane to the model's purpose and decision to be made. When a second model is used in that same decision space, the introduction of new or different Objects or supporting Attributes or absence of others can change the values of these measures. Even with consistent Objects and Attributes defining the measures, there is the possibility of influences on those Objects from other conceptual components in the second model.

Changes in the measures of merit, both MOEs and MOPs can also occur to changes or differences in the Processes that influence the model attributes the metrics require. Even similar Objects or Processes may have different Relationships in their respective models that are not aligned to one another. As discussed in section 2.4, the inclusion or exclusion of system concepts into a model may be deliberate or implicit in defining the model at its conceptual stages. The introduction of new concepts into the decision space is due a difference in the models' assumptions, purposes, or both. A means to evaluate a consequence of integrating multiple models is the changes, additions or deletions of critical metrics as they are the basis for the decision or decisions being made.

Changes in both MOEs and MOPs could range from minor to significant. The introduction of a new conceptual components from an additional model may augment, change, or contradict the metrics of a single model on its own. Cases where metrics change significantly or new metrics contradict previous metrics are the scenarios of highest consequence to the overarching purposes of the models and resulting decisions. Developing a hierarchy of preferences for MOEs and MOPs as consequences is relatively straightforward. The potential scenarios are listed in order of increasing gravity. The levels are weighted using the principle of Maximum Information Entropy that was presented in section 2.5.1. The same principle will be applied here to determine weightings for MOEs and MOPs in the Consequence component. When the only piece of information is a general preference order of categories, we will equally divide the consequence space from zero to one and take the centroid value of each subspace. To develop a hierarchy table of MOPs and MOEs, we need only list a preference order of categories. Characteristics of these categories are the significance of changes in MOP values upon the integration of an input model – minor, moderate, or significant, the introduction of new attributes as part of the MOPs, and if new attributes exist whether they contradict the original model's MOPs or not yields nine categories to measure consequences of model integration. This value hierarchy is depicted below in Table 31.

Similarly, a value hierarchy for MOEs can be constructed with the same categories and definitions of categories, shown below in Table 32.

TABLE 31

VALUE HIERARCHY FOR MOPS

Preference Order	Change in MOPs' Values	New Attributes in the MOPs	Conflicting Attributes	Upper bound of level	Centroid Weighting
1	Minor	No	NA	0.111	0.056
2	Minor	Yes	No	0.222	0.167
3	Minor	Yes	Yes	0.333	0.278
4	Moderate	No	NA	0.444	0.389
5	Moderate	Yes	No	0.556	0.500
6	Moderate	Yes	Yes	0.667	0.611
7	Significant	No	NA	0.778	0.722
8	Significant	Yes	No	0.889	0.833
9	Significant	Yes	Yes	1.000	0.944

TABLE 32

VALUE HIERARCHY FOR MOES

Preference Order	Change in MOEs' Values	New Attributes in the MOEs	Conflicting Attributes	Upper Bound of Level	Centroid Weighting
1	Minor	No	NA	0.111	0.056
2	Minor	Yes	No	0.222	0.167
3	Minor	Yes	Yes	0.333	0.278
4	Moderate	No	NA	0.444	0.389
5	Moderate	Yes	No	0.556	0.500
6	Moderate	Yes	Yes	0.667	0.611
7	Significant	No	NA	0.778	0.722
8	Significant	Yes	No	0.889	0.833
9	Significant	Yes	Yes	1.000	0.944

The Maritime Security Risk Model allows for the combination of different consequence types for use in U.S. Coast Guard applications and analysis to allocate resources for port security and various other operations [43]. In that specific case, the consequences are "primary" but also a "secondary economic impact" that accounts for concepts such as redundancy or recoverability of the asset under analysis. In the general case of model combinations, the impacts of MOEs and MOPs can be combined to arrive at a single measure of consequence.

Both MOEs and MOPs have potential changes in their values that are part of the consequence portion of risk. The changes in their values are measurable, proving useful to a risk calculation. Any number of MOEs can be combined with one another, and any number of MOPs can be combined with one another. The *consequence* portion of risk is extended as tuple for both MOEs and MOPs to

$$C = D(MOE), D(MOP) \tag{15}$$

Where C is the consequence,

$D(MOE)$ is the difference in the values of Measures of Effectiveness

$D(MOP)$ is the difference in the values of Measures of Performance

Where the alignment of model Objects, Processes, and Relationships examine the structural differences among models, the change of MOEs and MOPs addresses the impact of the models' integration.

4.4 Model Integration

Model Integration is a meta-modeling concept wherein two or more models are joined together after execution. The models are assumed to have their own unique, stand-alone purposes. Reference [44] notes there is not a formal definition of model integration, but that it is practiced for several reasons. Reference [45] identifies several types of model interaction that require a view of the models' semantics and identify some of the reasons this form of model integration may be done:

1) Concatenation: The models being examined share representations and can get instances from one another.
2) Amplification: A model adds or augments to the representation in another model.
3) Parameter Discovery: One model is used to develop parameters as inputs into another model.
4) Model Construction: One model is used as the basis to construct a model of a different type
5) Model Merging: Meta-modeling wherein a wholly different model is created by the merger of one model's structure with the methodology of a second model.

This dissertation has primarily focused on the third type of model integration, *Parameter Discovery*. That is, where a model of higher fidelity and smaller scope is used to inform a model of broader scope and lower fidelity.

Model Integration is a practice that extends beyond model construction, but into model management as well. The larger scope of model management, which includes model integration introduces complexities beyond those that may be found in a single model alone [44]. The complexities are many and have been discussed in this dissertation. They act of model integration is a state of practice wherein one model can act as the data source of parameters for another model.

For this dissertation, model integration will be taken to mean the practice of developing parameters of a model based upon the metrics—both MOEs and MOPs—of another model.

Def. 7 *Model Integration*: The mapping of one or more outputs of a model to one or more input parameters of a second model.

Model integration is fundamentally a human activity and is left to savvy analysts and model users to understand the implications and purposes of the models they are using. As models will be dependent upon data or data sets, a model's output can act as the data inputs for another. As has been discussed, however, the outputs of a model are subject to the conceptual components and the context of that model. When those components are hidden or unaccounted for in the integration activities, the mapping of a model's outputs into another's inputs may influence the outputs.

In reference [45], Levis and Jbara further identify the workflow practice of integrating models as a modeling activity itself. As had been presented in section 2.5, models carry with them a certain intrinsic risk in their usage. Understanding and appreciating the risk of model integration is the purpose of this dissertation.

The practice of model integration is not arbitrary as it is a state of practice in large enterprises with disparate modeling and analysis needs. The emphasis on reusing models and insights garnered from them is an important concept for knowledge management and savings of both time and cost. However, as has been shown, this practice is not well-defined.

4.5 Risk Calculation Theory

From the previous subsections on the dimensions of Risk, there are a number of model misalignments that can generate risk in model integration. The risk tuple

$$R = \langle S, D(Obj), D(Proc), D(Rel), D(MOE), D(MOP) \rangle \tag{16}$$

can help identify the risk profile of permutations among the alignment of conceptual elements and the potential changes in metrics from the model. In plain words, the risk tuple reads that integration risk is a function of the alignment scenario, the differences of each of three conceptual dimensions of the model, and the impact the integration of the models has on the outputs of the modeling process. Across the various scenarios of misalignment, set theory was applied to measure the differences among three different conceptual components of models in section 4.2. Section 4.3 then provided a weighted centroid method for domain and subject matter experts to categorize the changes in outputs that a model would offer. Using these alignments and these impacts, the major components of risk are available for examination.

The calculation of alignments across the dimensions of objects, processes, and relationships within the model can be treated as average of the three conceptual elements. The calculations of alignment in each conceptual element were themselves a calculation of set theory and represent the total alignment between the models on each conceptual dimension. A perfect alignment between two models would a 1.0 value, so all alignments that the models share is deducted from 1.0, representing the misalignment of the models. The values are the results of sets that are decision criteria for model developers and analysts. The misalignment in each conceptual dimension are derived from value hierarchies [15], and averaging these values is treated as probabilistic calculation. The calculation of the misalignment between two models follows the general form of:

$$P(misalignment) = p(misalignment\ of\ Objects) \cup p(misalignment\ of\ Processes) \cup p(misalignment\ of\ Relationships) \quad (17)$$

$$= p(misalignment\ of\ Objects) + p(misalignment\ of\ Processes)$$
$$+ p(misalignment\ of\ Relationships)$$
$$- p(misalignment\ of\ Objects) \times p(misalignment\ of\ Processes)$$
$$- p(misalignment\ of\ Objects) \times p(misalignment\ of\ Relationships)$$
$$- p(misalignment\ of\ Processes) \times p(misalignment\ of\ Relationships)$$
$$+ p(misalignment\ of\ Objects) \times p(misalignment\ of\ Processes)$$
$$\times p(misalignment\ of\ Relationships)$$

The total misalignment of the two models is:

$$D(ModelMisalignment)$$
$$= D(Obj) + D(Proc)$$
$$+ D(Rel) - [D(Obj) \times D(Proc)] - [D(Obj) \times D(Rel)] - [D(Proc)$$
$$\times D(Rel)] + [D(Obj) \times D(Proc) \times D(Rel)]$$

Where

$D(Obj) = 1 - A(Obj)$, representing the misalignment between models' Objects.

$D(Proc) = 1 - A(Proc)$, representing the misalignment between models' Processes.

$D(Rel) = 1 - A(Rel)$, representing the misalignment between models' Relationships.

Changes in the integrated model's MOEs and MOPS is also derived from value hierarchies as presented in section 4.3. Changes in both MOEs and MOPs are likewise combined using probability statements. In a simple case of two values, the numbers can simply be averaged. In more complex situations with multiple MOEs or MOPs, the combination of MOEs and MOPs follow the general form combing metrics:

$$P(metrics) = p(change\ in\ MOE1) \cup p(change\ in\ MOE2) \cup \ldots \cup p(change\ in\ MOEn)$$
$$\cup\ p(change\ in\ MOP1) \cup p(change\ in\ MOP2) \cup \ldots \cup p(change\ in\ MOEn).$$

With methods to calculate probabilities of misalignment and consequences available, overall risk can be calculated. With the changes of model metrics—both MOEs and MOPs—a result of the inclusion of an additional feeder model, then the Risk due to Model Integration is defined as Model Results will adversely affect the decision because of Model Integration *and* that Model Results are worsened because of Model Integration *and* that Model Results adversely affect the decision. This is mathematically defined as:

$$Integration\ Risk = p(misalignment)\ x\ p(metrics)\ x\ [1 - p(misalignment) + p(misalignment)\ x\ p(metrics)] \quad (18)$$

Values for the misalignment, the metrics, and therefore the overall risk will range between 0 and 1. Higher values of misalignment indicate that the models have relatively poor alignment in their conceptual components. Higher values in the metrics mean that there are significant changes to the model's outputs. Unsurprisingly, there is higher risk to the decision from model integration when alignment is poor and when metrics change significantly. Likewise, there is lower risk when the models are well-aligned and the changes to metrics are small. However, the value of this analysis is identifying risk values for moderate changes in either alignments or in metrics.

Fig. 12 presents the risk surface response to changing combinations of models' alignments and changes to model outputs. The X axis represents changing values of the aggregate of misalignments, scaled from 0 to 1 where 0 represents a perfect alignment between the two models and 1 means complete misalignment. The Y axis represents changes to models' outputs in both MOEs and MOPs, ranging from 0 to 1 where 0 means no change and 1 means significant change. The Z axis represents the calculated integration risk where 0 represents no risk and 1 represents a high risk to the quality of the decision and model credibility. The surface area of this curve is larger in regions of lower risk, and smaller in regions of higher risk. This indicates that the risk of model integration may in fact be skewed towards smaller risks.

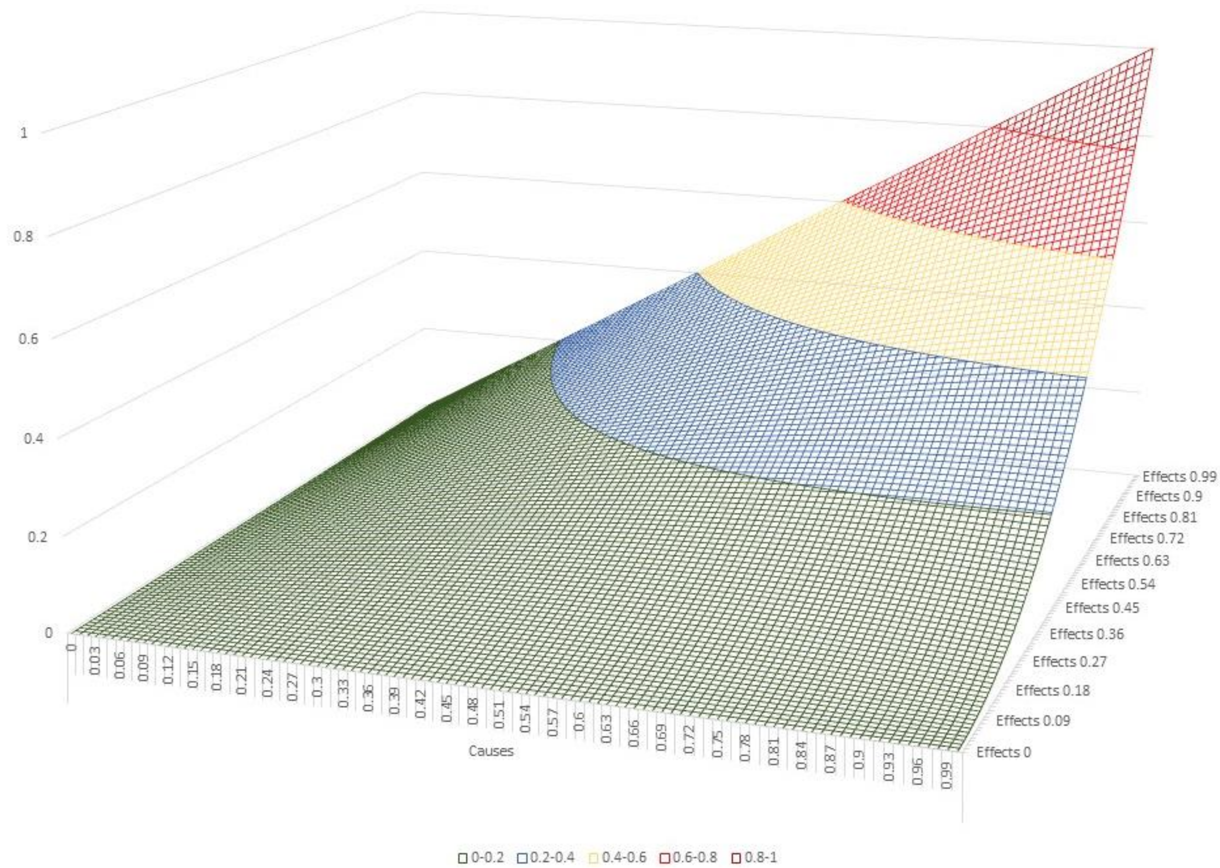


Fig. 12. Integration risk as a function of causes and effects.

The image above depicts break points along the surface of the curve at 20% intervals of integration risk. The green region is the lowest risk portion of the curve and accounts for 63.38% of the surface area. The blue region is the region where integration risk ranges from 20% to 40% and accounts for 21.02% of the surface area. The yellow region is the portion of the curve where risk is between 40% and 60%, representing 9.86% of the surface area. The light red is the region where risk is between 60% and 80%, representing 4.45% of the surface area. The upper most, dark red, region depicts the portion of the curve where integration risk exceeds 80%, and represents 1.29% of the total surface area.

CHAPTER 5

DEMONSTRATION WITH TRANSPORTATION MODELS

To show a risk to the quality of a decision, two models in the transportation field are used to demonstrate the connection of model alignment to integration risk. The selection of transportation models is arbitrary, and the alignment and risk process can be applied to models in other domains as well. Specifically, the models are RtePM and SUMO where selected because of their relative availability. RtePM stands as a macroscopic model that provides high-level insights to decision makers regarding the ability to evacuate citizens from a given area whereas SUMO is a microscopic model that contains significantly more detail and is used to support traffic engineering decisions. Both are transportation models and explicitly model vehicular traffic on road networks. This chapter will discuss the development of the models, their alignment, and their integration to derive a risk to the quality of decisions.

5.1 RtePM

5.1.1 Overview

RtePM is a macroscopic transportation model that makes estimates of total time to evacuate vehicles from a given region [46]. Its overall purpose is to “enable emergency managers to gain insights from testing various evaluation scenarios” [46]. The model user has several parameters available to manipulate, such as the region size itself, the time of day, people per vehicle, and the population ratio that heeds the evacuation warning. It explicitly models vehicles in a stochastic, time-stepped simulation. Calculated or implicit elements include the numbers of and speeds of vehicles in the model.

RtePM was selected for this analysis to serve as the macroscopic, or more-broadly scoped model. It is heavily reliant on transportation model with the primary function of calculating the entire time to clear a region of its population. It is free to the public and maintained by Old Dominion University and VMASC. It was developed to support planning efforts of the U.S. Department of Homeland Security and enhance by VMASC to support requirements from the Virginia Department of Emergency Management.

5.1.2 Conceptual Components of RtePM

Recalling from the OPR method in reference [1], discussed in section 2.4, objects are the elements of a model that have a persistent existence. They are defined by *qualities* or attributes that distinguish the object from other objects. Values of these attributes can change dynamically over time define each objects' state at different times in the simulation. The qualities that are available to define any given object are also important to note as they will aid in the similarity metrics presented in section 4.2. In order to compare one model's concepts to another model's concepts for the eventual purposes of model similarity, a clear understanding of the concepts and their attributes needs to be defined.

5.1.2.1 RtePM Objects

The first step in identifying the Objects in RtePM are to begin with the user guide or model documentation from VMASC and the independent validation assessment conducted by Omni engineering on behalf of VMASC [47]. These documents are not structured to explicitly list model object or entities. However, each in turn enumerates certain concepts that shall be categorized as objects, processes, or relationships. The concepts identified are those necessary to define a complete executable scenario in RtePM, and do not include metadata, such as scenario descriptions, save data, or user data. The Object concepts and their supporting attributes or qualities are depicted in the series of images that follow and briefly described after each depiction.

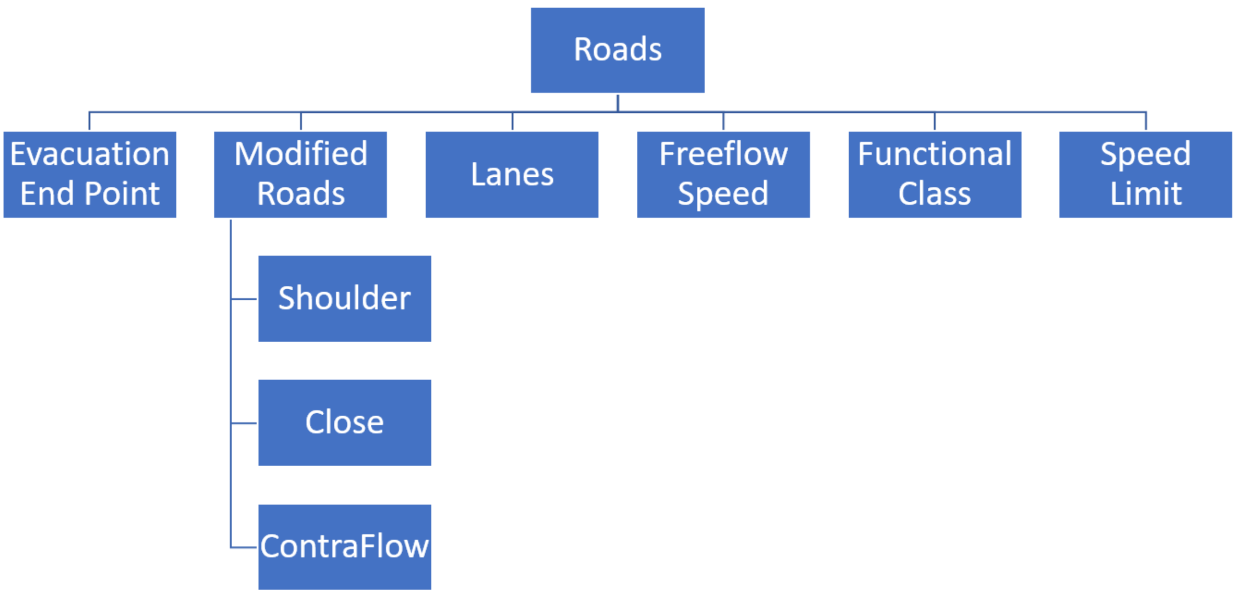


Fig. 13. RtePM roads objects.

As an evacuation model with significant reliance on transportation, *roads* are a prominent object in RtePM. The roads themselves are collected geographically from a proprietary road network. The proprietary nature of the roads makes it difficult to define them. However, the user guide provides qualities of roads that are sufficient for this dissertation. The breakdown of this Object is depicted above in Fig. 13 and described as follows:

1) Evacuation End Point which is particular to roads at the end of the network and designate an egress from the system.

2) Modified Roads that indicates whether a road has been modified from its most basic instantiation. This is further refined with qualities for the (A) Shoulder—toggled to allow traffic to drive on the road's shoulder, (B) Close—toggled to disallow any traffic to use this road, and (C) ContraFlow—toggled to allow traffic to flow in a reverse direction.

3) Lanes, indicating the number of lanes on the road.

4) Freeflow Speed, the average speed in miles per hour on this road.

5) Functional Class, a textual description of roads that generally relates numbers of lanes and speed to either a "highway," "major artery," minor artery," or "smaller." These descriptions allow for refined searching and editing in the road network.

The second object concept in RtePM is the *evacuation area* which generally defines the population as entities in the system. This concept also leads to the derivation of the number of vehicles in the system. The major qualities and minor qualities defining the object depicted below in Fig. 14 are:

1) The Population Block, which is further defined by the daytime population, the nighttime population, the number of households, an implicit geographic location, and an implicit shape. These latter two qualities are implicit because they are not directly observable by a model user but are inherently part of the data provided by the U.S. Census Bureau that defines the population blocks. There can be many population blocks in an evacuation zone.
2) An optional Seasonal factor that is particularly useful for regions that have significant tourist or visiting populations. The layer is defined by a name, an additional population number, a vehicle occupancy of the additional population, and a geographic shape of the region.

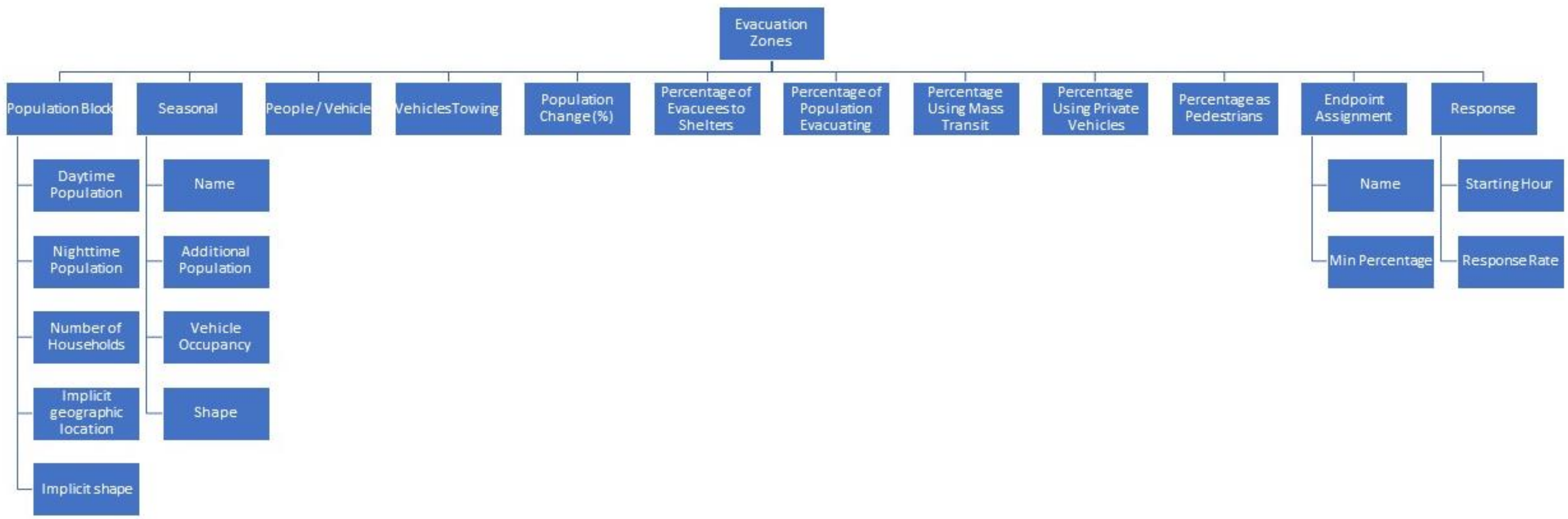


Fig. 14. RtePM evacuation zones objects.

3) The number of people per vehicle. Coupled with the time of day and the population of a given block, the number of vehicles is calculated from this information.
4) The percentage of vehicles towing trailers or other items.
5) The percentage of population change, which allows for a scalar multiplier of the underlying census data.
6) The percentage of evacuees to shelters defines what portion of the evacuation zone will go to a shelter rather than egress from the zone.
7) The percentage of population evacuating is the portion of the population that will attempt to evacuate, either by egress or to a shelter.
8) The percentage of population using mass transit, which is one of three modes of evacuation.
9) The percentage of population using private vehicles, which is one of three modes of evacuation.
10) The percentage of population as pedestrians, which is one of three modes of evacuation.
11) Endpoint Assignments are qualities that direct evacuees to which edge of the network they will attempt to use. It is refined by a name and by a minimum percentage, which is the minimal proportion of evacuees that will attempt to use this end point.
12) The response is refined by a starting hour and a response rate. The starting hour defines what time the evacuation begins, and the response rate provides a probability distribution that determines what portion of evacuees being their individual evacuations.

The third object concept in RtePM is the *shelter* which generally defines points within the system for evacuees to congregate instead of evacuating. It is defined by four qualities shown below in Fig. 15:

1) A toggle to activate or deactivate the shelter for inclusion in the system.
2) The text name of the sheltering facility.
3) The capacity of evacuees the shelter can accommodate.
4) A toggle for last resort, meaning that the shelter only becomes available during the simulation when evacuees have no other egress or shelter available to them.

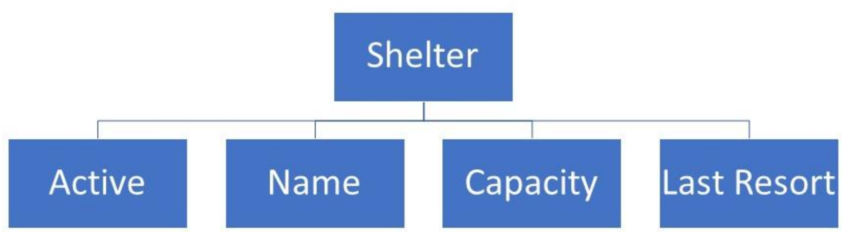


Fig. 15. RtePM shelters objects.

The fourth object concept in RtePM is vehicles. Vehicles are the major dynamic element that move about the system and influences many of metrics that RtePM provides to decision makers. It is a derived Object, meaning that the model user does not directly manipulate parameters of vehicles. However, the breakdown of the object is depicted in Fig. 16 and the Attributes are defined as:

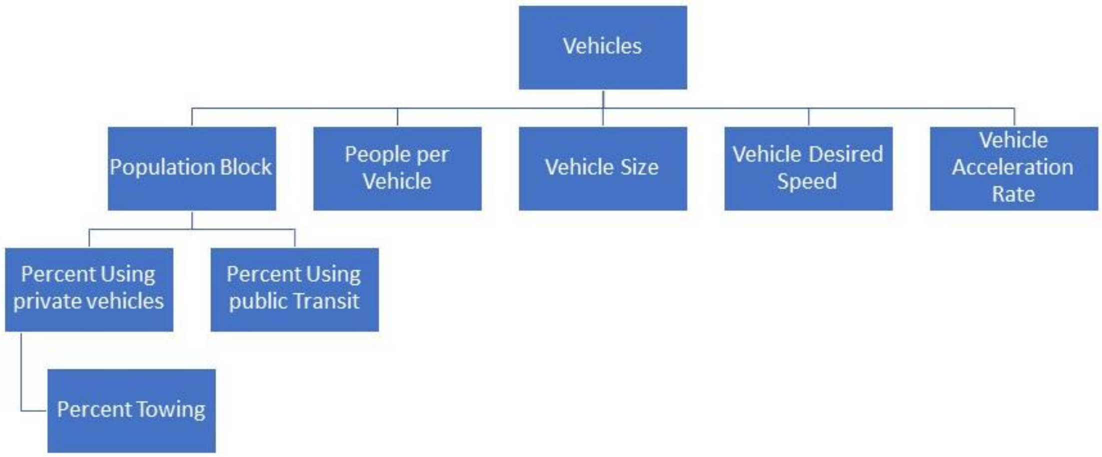


Fig. 16. RtePM vehicles object.

1) The population block, which is a previously defined object. In the population block is the number of persons required to evacuate. Vehicles generated in the system are defined by the further discriminating qualities of the percentage of population using private vehicles and the percentage of population using public transit. Each of these qualities allows a calculation for vehicles of different types, such as cars and busses.
2) The number of people per vehicle, which is a previously defined quality of the evacuation zone.
3) The vehicle size is not determined directly by the model user, but rather is either defined deterministically or stochastically based upon the simulation's runtime configuration.
4) The vehicles' desired speeds are not determined directly by the model user, but rather is either defined deterministically or stochastically based upon the simulation's runtime configuration.

5) The vehicles' acceleration rates are not determined directly by the model user, but rather is either defined deterministically or stochastically based upon the simulation's runtime configuration.

These two major qualities together allow for a calculated number of vehicles in the modeled system.

5.1.2.2 RtePM Processes

RtePM's Processes are proprietary, complicating their decomposition and assessment. VMASC's developmental work on introducing randomness from an earlier version of RtePM is well-documented, but some of the initial developmental work is unavailable. The relative opacity of the underlying processes / algorithms is illustrative of the overarching problem statement; oftentimes, model users and analysts do not have access to the complete details of a model's detailed calculations. As in the case of RtePM, there is frequently limited access to understanding an existing model's capabilities.

VMASC's developmental work makes clear there are several processes present that were improved and developed in the course of VMASC's management of RtePM. The first mentioned is traffic congestion modeling. This process is influenced by vehicles' length to which VMASC has introduced a stochastic distribution to define vehicle length [48].

Congestion affects the speed, density, and throughput of each road segment, which can be viewed as model output. This data is not directly exportable from RtePM, but its visualization is available and its simplicity is one of RtePM's touted advantages. The data can be exported on the RtePM server in short intervals by special request. Fig. 17 below depicts the breakdown of RtePM processes.

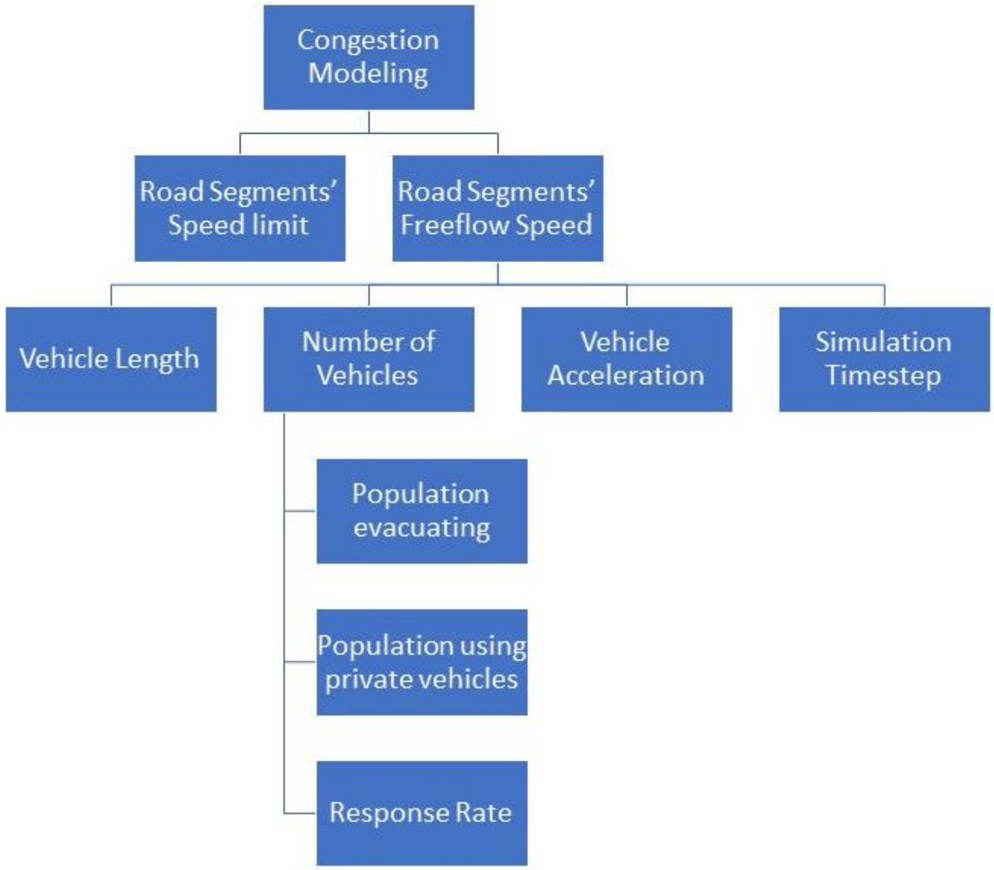


Fig. 17. RtePM's congestion modeling process.

The congestion that is depicted in RtePM is a selection between each road segments' speed limit and its actual freeflow speed. Freeflow speed is calculated by the distance to any vehicles ahead of an individual vehicle and its ability to accelerate to close that distance within a time step of the simulation. The number of other vehicles on the road segment is determined by the overall evacuation demand and response rate show in the user parameters of population evacuating, population using private vehicles, and overall response rate.

The next important process in RtePM is traffic signal phasing. Traffic signals are implicitly modeled at every major intersection of the model, regardless of the presence of an actual traffic signal at the intersection. The phasing of traffic lights refers to the length of time that traffic can flow through the intersection in a particular direction. Traffic signal phasing at each intersection is determined by the sets of non-conflicting traffic flowing into an intersection and by the relative proportions or traffic flow within those sets. Fig. 18 below depicts the components of traffic signal phasing. The resulting timing allows traffic to cross the intersection without interfering with cross traffic.

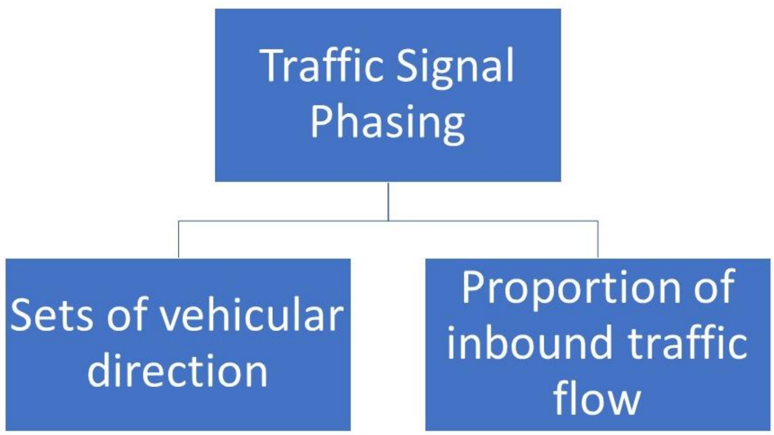


Fig. 18. RtePM traffic signal phasing.

A validation report of RtePM conducted by Massachusetts Institute of Technology Lincoln Laboratories explicitly identifies four additional processes that are perhaps intuitively necessary for a transportation focused model [49]. They are a car following algorithm, vehicle entry, path determination, and clearance of all vehicles from the system.

Fig. 19 below depicts the structure of the car-following model with descriptions of its supporting characteristics. This Process defines how vehicles traverse the network and maintain distance from one another.

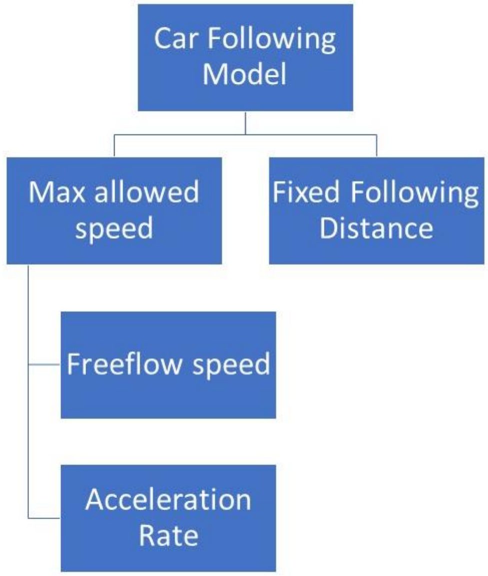


Fig. 19. RtePM car-following model.

Cars will traverse the system at the maximum of the highest speed allowed or be remaining a fixed distance behind the vehicle ahead of them. The maximum allowed speed is decomposed into a freeflow speed and the speed to which each vehicle can accelerate in a one-second time interval.

The next Process is the vehicle entry process. It is decomposed in Fig. 20 below. This Process defines the entry of Vehicles onto the road network from each population block.

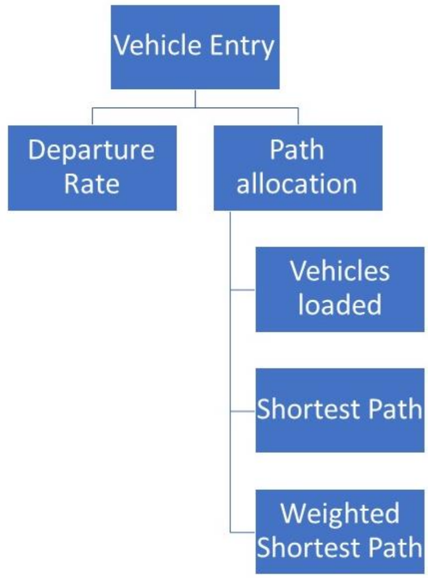


Fig. 20. Vehicle entry.

The departure rate Characteristic is set by a user-defined S-curve setting the response rate for evacuees in the system – it defines the probability of evacuees at any given time in the simulation. The vehicles are generated as a product of the response rate and by the population using vehicles. As vehicles enter the road network, they are allocated a path. A portion of vehicles follow a shortest path and a portion will dynamically calculate the weighted shortest path given present conditions. Vehicles are loaded onto the roadway at regular time intervals.

Vehicles determine their path through the network when they are loaded onto the network and do not dynamically adjust their routes. The path determination Process is depicted in Fig. 21 below.

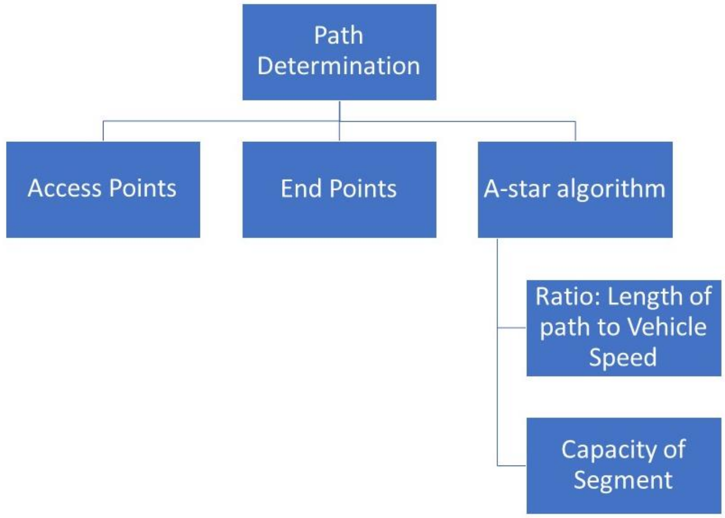


Fig. 21. Vehicle path determination.

Each Vehicle's path is determined by the access points available to it at its sourcing population block, its end point, and a shortest path A* algorithm. The A* algorithm determines the lowest cost of each road segment as a ratio of the length of each segment to the vehicle's speed and is weighted by the segment's vehicular capacity.

Lastly, a periodic global Process checks to see if all the evacuating vehicles have left the road network. This is Process is also a major MOE of the model.

5.1.2.3 RtePM Relationships

Relationships are linkages between two other conceptual components. They have rules that define them to give them proper context. Relationships are not a documented concept in RtePM and are implicit in the descriptions between and among Object and Process concepts. Identifying the Relationships in RtePM is accomplished through the breakdown of the other components where dependencies among conceptual components can be identified. Descriptions of the Objects and Processes in the previous two subsections identify the Relationships in RtePM and the other concepts that they link together. The Relationships and the other concepts that they link together are summarized in the Table 33.

TABLE 33

RtePM RELATIONSHIPS

Concept 1	Concept 1 Type	Concept 2	Concept 2 Type
Roads	Object	Vehicle	Object
Roads	Object	Shelter	Object
Roads	Object	Signal Phasing	Process
Evacuation Zone	Object	Vehicle	Object
Vehicle	Object	Signal Phasing	Process
Congestion	Process	Signal Phasing	Process
Congestion	Process	Roads	Object
Vehicle Entry	Process	Vehicle	Object
Vehicle Entry	Process	Population Block	Object Attribute
Vehicle Entry	Process	Roads	Object
Car Following Model	Process	Vehicle	Object
Car Following Model	Process	Freeflow Speed	Object Attribute
Path Determination	Process	Roads	Object
Path Determination	Process	Vehicles	Object
Path Determination	Process	Entry Point	Object Attribute
Path Determination	Process	Evacuation Point	Object Attribute

5.2 SUMO

5.2.1 Overview

The Simulation for Urban Mobility—or simply SUMO—is a "microscopic, open-source, multi-modal traffic simulation tool" [50]. The model is microscopic, explicitly incorporating each vehicle in the system and assigning a destination to every vehicle in the system. Given that it is a traffic simulation, it has different objectives and purposes than a tool such as RtePM for evacuation planning and emergency management. SUMO will calculate metrics related to traffic engineering, such as traffic light states, lane usage, queues (at junctions), air quality, and fuel consumption. With the higher level of detail available in SUMO, it consequently has a need for significant data inputs.

SUMO was developed by the Institute of Transportation Systems in Berlin, Germany for a variety of traffic engineering purposes, such as intersection performance, traffic forecasting, and vehicle routing. SUMO was chosen for this analysis in part due to its availability on the open web, and in part because of its flexibility in modeling vehicular networks at any arbitrary place in the world. It is the selected microscopic modeling tool for this dissertation.

SUMO is a free and open source modeling tool is well documented in its website with wiki-like entries, but also points directly to literature written by some of the developers that affords a consolidated listing of conceptual components in SUMO [51].

5.2.2 Conceptual Components of SUMO

5.2.2.1 SUMO Objects

The following conceptual Objects are identified in SUMO's documentation. SUMO is a suite of applications that support its traffic simulation capability. One of the underlying conceptual Objects is *roads*. Roads are importable and configurable from a variety of python-based scripts available in the SUMO suite.

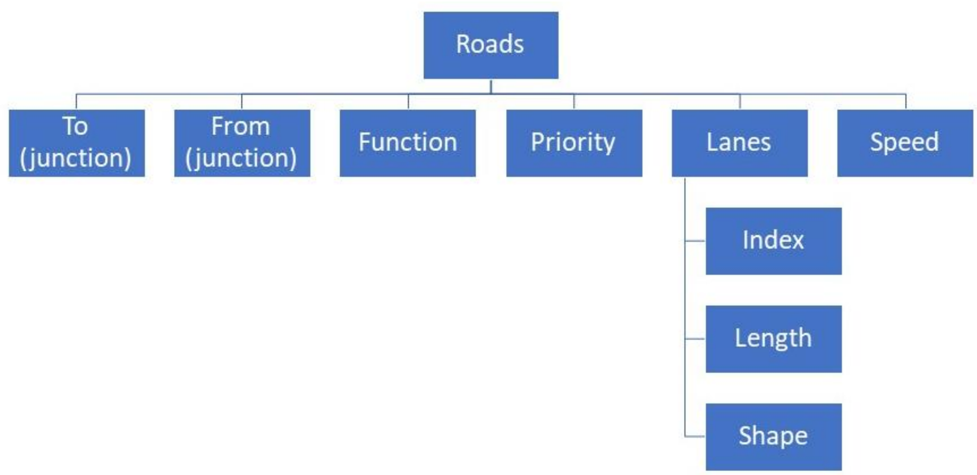


Fig. 22. SUMO roads object composition.

Fig. 22 depicts the qualities that define roads in SUMO. Every road Object in SUMO is a one-way edge with a collection of one or more lanes that define the road. Following is a brief description of the Attributes of this Object.

1) "To" Junction is representative of an intersection and denotes where the road segment ends and the direction of traffic flow.

2) “From” Junction represents the intersection from which the road segment originates and the source of traffic flow.
3) The function of the road segment denotes several options to classify the road. Road functions can be: A) ‘normal,’ meaning they are part of the road network connecting two points to one another. B) ‘connector,’ meaning that it is not representative of a real-world road segment and added by users to facilitate the represented network. C) ‘internal,’ meaning they are representative of connections within an intersection, D) ‘crossing,’ meaning it is unique for pedestrian traffic to cross a road, or E) ‘walking area,’ meaning that they are exclusive areas for pedestrian traffic.
4) Priority defines a road segment’s relative importance which allows for right-of-way decisions at intersections and pedestrian crossings.
5) Lanes are a large part of defining roads. They are further decomposed into A) an index, which is more than simply metadata, and defines the order of lanes within a road segment from right to left. B) speed, which defines the maximum speed permitted in the lane. C) length, which describes the length of the lane. D) shape, which is a position vector describing the curvature and height of the lane.
6) Speed is the maximum allowable speed on the road.

The next major conceptual Object in SUMO are junctions. Junctions are nodes within the road network and typically represent intersections where traffic flows cross one another. Fig. 23 depicts the breakdown of this Object into its defining qualities.

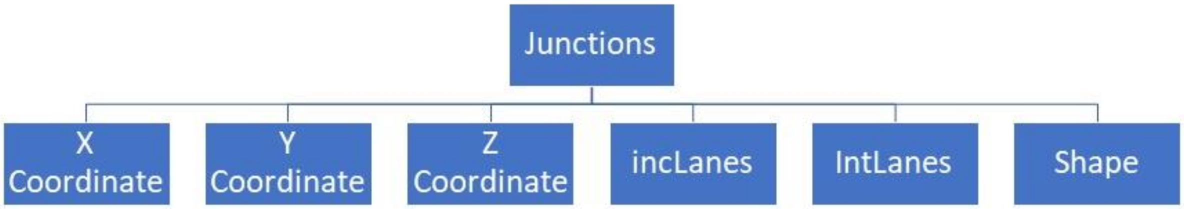


Fig. 23. SUMO junction object composition.

1) The "X Coordinate" is the real value depicting the left-right or East-West location of junction.
2) The "Y Coordinate" is the real value depicting the up-down or North-South location of the junction.
3) The "Z coordinate" is the elevation of the junction. While the value is optional, it is useful for interpolating the rise or slope of a segment of roadway.
4) "incLanes" is a list of lanes that end at the junction.
5) "intLanes" is a list of lanes *within* the junction, meaning they are responsible for connecting inbound and outbound lanes to one another.
6) "Shape" is a list of points defining a polygon shape that is the boundary of the intersection.

Connections are a major conceptual Object in SUMO. Connections are linkages describing the ingress and egress of lanes. Connections are similar to junctions, but there is not a necessity of a crossflow of traffic, and connections can depict points along a road where the numbers of lanes either increase or decrease. Fig. 24 depicts the breakdown of this Object into its defining qualities.

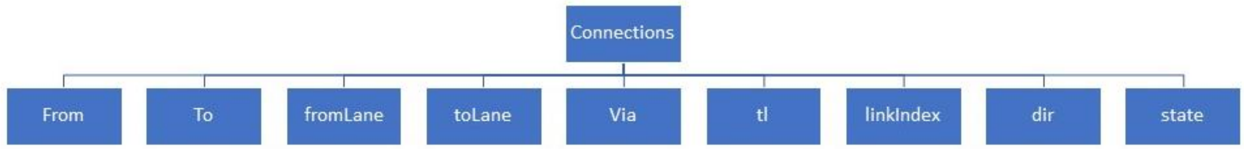


Fig. 24. SUMO connections object.

1) The “from” value is the ID of inbound road.
2) The “to” value is the ID of outbound road.
3) The “fromLane” value identifies which lane on the incoming edge where the connection begins.
4) The “toLane” value identifies which lane on the outbound edge where the connection ends.
5) “via” is the ID of the lane to govern the connection across a junction.
6) “tl” is the ID of a traffic light that controls the connection.
7) “linkIndex” is defines the traffic signal’s grouping, allowing for the synchronization of traffic lights across intersections.
8) “dir” defines the direction of a connection – either straight, left, or right.
9) “state” defines the state of connection with parameters available for the state of traffic control and the functionality of the linkage.

Traffic Lights is the next conceptual object in SUMO. Traffic lights are the governance mechanisms at intersections in the road network. They are decomposed into two major branches to describe the traffic light. The first component of the traffic light is its logic or “tlogic;” this defines the order of each color of light in each direction of the intersection. The second major component of the traffic light is its phase which defines the timing of each signal phase. The Object's decomposition is depicted below in Fig. 25 with descriptions of each Attribute following.

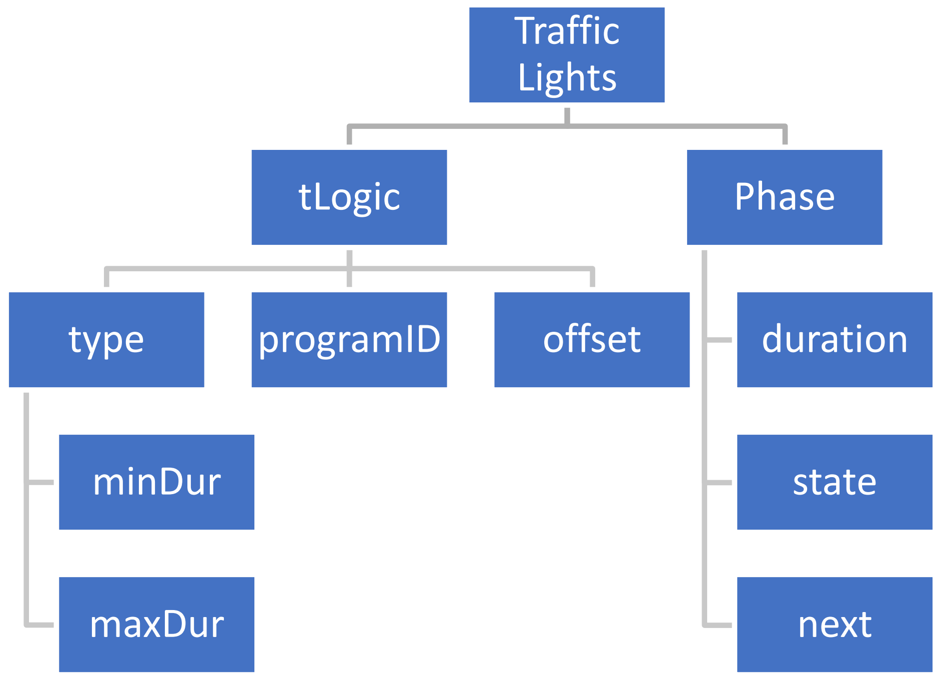


Fig. 25. SUMO traffic light object.

The Attributes of Traffic Lights are as follows:

1) "tLogic" is the governance mechanism to establish the traffic light's phasing and timing.
2) "Phase" is the description of the light's functionality, to include the duration of each light and the rules of the color. For example, right-turn-on-red or prioritizing certain vehicles.

3) "type" is an Attribute of "tLogic" that defines the class of the traffic signal: strictly timed, actuated by vehicle detectors, or a queue-based delay of vehicles waiting.
4) "programID" is Attribute of "tLogic" that describes phases at the traffic light.
5) "duration" is an Attribute of "Phase" that is simply the length of the phase in seconds.
6) "State" defines the state of the traffic light. Beyond simple "Red, Yellow, Green," parameters are available to define passage for certain vehicles (ex: emergency vehicles or mass transit), pedestrians, and right-turns-on-red.
7) "minDur" is the minimum length of time of a light's phase relevant for when the signal is actuated.
8) "maxDur" is the maximum length of time of a light's phase relevant for when the signal is actuated.
9) "next" describes which phase of the light follows the current state.

"Requests" is an abstract Object within SUMO that sets priorities for traffic flows that intersect each other. Fig. 26 depicts the Attributes that define the Requests Object. The Attributes that define the requests Object are defined below.

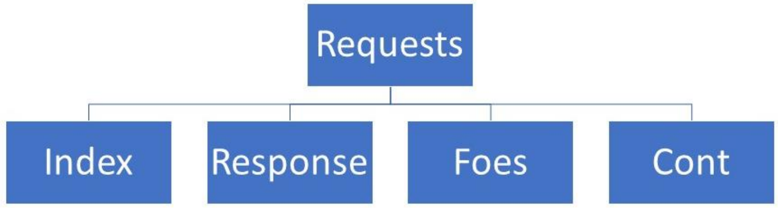


Fig. 26. SUMO requests object.

1) The “index” value is the connection index number within a right of way matrix.
2) The “response” value is applied to each connection and indicates whether vehicles may pass at speed or must decelerate to pass.
3) The “Foes” value identifies which lane on the incoming edge where the connection begins.
4) The “toLane” value identifies which lane on the outbound edge where the connection ends.
5) “Requests” is an abstract Object within SUMO that sets priorities for traffic flows that intersect each other. Fig. 26 depicts the Attributes that define the Requests Object.

Another Object identifiable from the SUMO documentation is the concept of *routes* which are the path by which vehicles transit the network of roads in the model. It is composed of one significant Attribute: edges. Fig. 27 below depicts the breakdown of the Object. The “id” value is simply a unique identifier of the route and does not necessarily provide value to defining a route. More importantly is the list of edges that defines the route. This is a non-empty set of connected road edges.

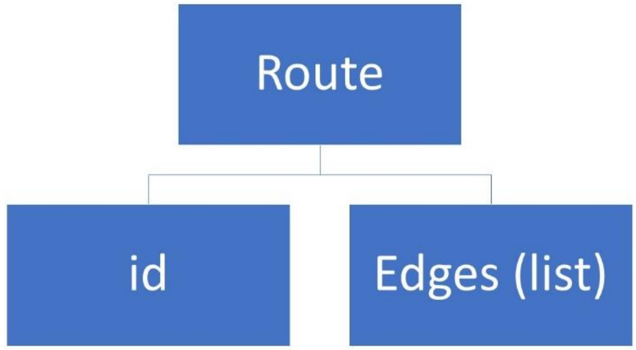


Fig. 27. SUMO routes object.

The next Object defined is V*ehicle Type.* As a microscopic traffic and transportation model, SUMO offers a significant detail on its vehicle Objects. In Fig. 28 below depicts the large number of Attributes that SUMO uses to define it vehicle type Object and appears to be the most complex Object in the model [50]. The Attributes that define the vehicles Object are listed below.

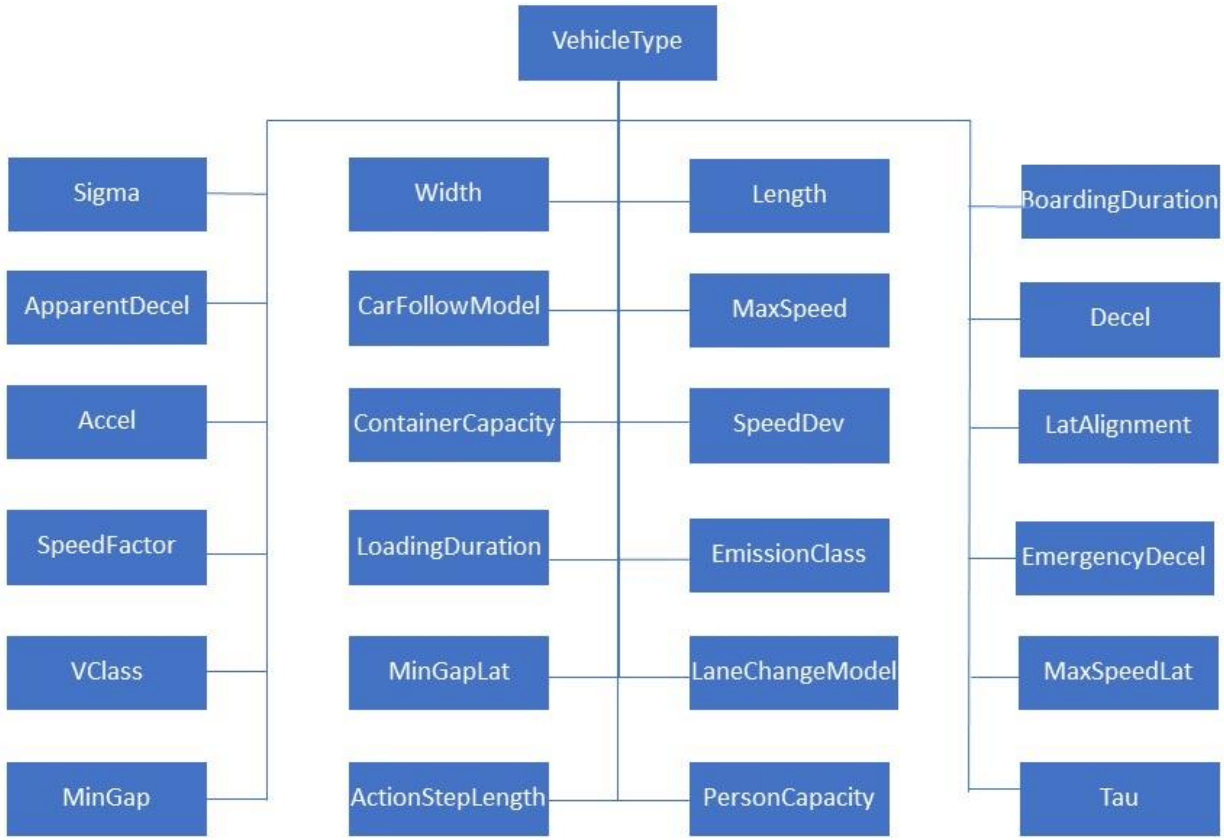


Fig. 28. SUMO vehicles object structure.

The Attributes that define Vehicle Type are:

1) “Accel” is the vehicle’s acceleration capability.

2) “Decel” is the vehicle’s deceleration capability.

3) “ApparentDecel” is the perception of the following vehicle of the deceleration of this vehicle.

4) “EmergencyDecel” is the maximum deceleration possible for this vehicle.

5) "Sigma" is a parameter for the car-following process, described in the next section.
6) "Tau" is a parameter for the car-following process, described in the next section.
7) "Length" is the vehicle's physical length.
8) "MinGap" is the vehicle's desired following minimum distance from the vehicle in front of it.
9) "MaxSpeed" is the fasted velocity possible by this vehicle.
10) "SpeedFactor" is the vehicle's ratio to the posted speed limit.
11) "SpeedDev" is the vehicle's variation from its own speed factor.
12) "VClass" is the class of vehicle. It can be useful for lane restrictions.
13) "EmissionClass" defines the exhaust outputs of the vehicle from a listing of different fuel types and efficiencies.
14) "Width" is the vehicle's physical width.
15) "LaneChangeModel" selects the model that governs this vehicle's willingness to change models and speed of doing so.
16) "CarFollowModel" selects the governing process for the vehicle's following distance and behavior
17) "PersonCapacity" defines how many people can be in this vehicle.
18) "ContainerCapacity" defines the number of containers this vehicle can transport. Typically for commercial vehicles, such as trucks.
19) "BoardingDuration" is the time that it takes for a person to board a vehicle.
20) "LoadingDuration" is the time required to load a container onto a vehicle.
21) "LatAlignment" is the preference of the vehicle regarding its orientation in a lane.

22) "MinGapLat" is the vehicle's desired space between vehicles to either its left or right.
23) "MaxSpeedLat" is the maximum lateral speed a vehicle will use.
24) "ActionStepLength" defines a time interval by which the vehicle will execute its logic.

Following the "Vehicle Types" Object is the "Vehicle" Object itself. The important difference between these Objects is that the Vehicle Types creates an abstract class from which Vehicles derive information. Vehicle type is an Object because of its persistent set of values that remain throughout the model. Individual Vehicles are more precise instantiation of Vehicle Types. Fig. 29 below depicts the attributes that define Vehicles, and are describe as follows:

1) "Type" is defined from the Vehicle Type Object before that sets many of the Attributes a Vehicle has throughout the simulation.
2) "Route" is a selected from a list of defined routes that are a collection of edges a vehicle will follow through the road network.
3) "Depart" is the time step when a vehicle begins its transit through the system, or enters the network.
4) "departLane" is the specific lane where the vehicle will begin its transit through the network.
5) "departPos" defines the position of the vehicle as enters the network.
6) "departSpeed" defines the speed of the vehicle when it enters the network. This Attribute shows that a speed of zero is not necessarily assumed, the vehicle may already be traveling at speed when it enters the simulated road network.
7) "ArrivalLane" is the lane where the vehicle exits the road network.

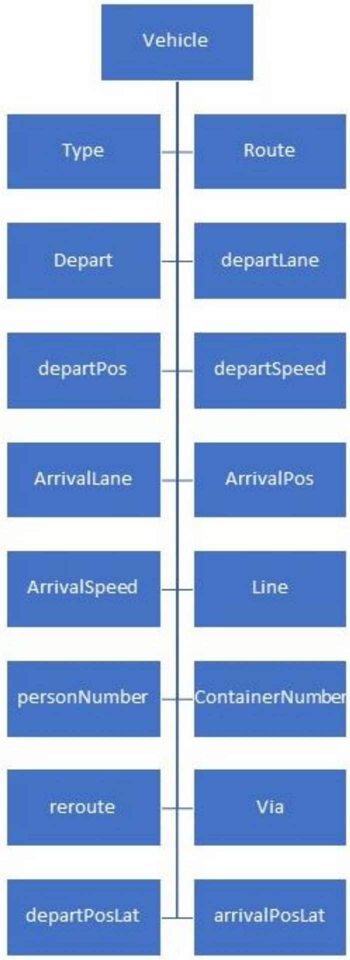


Fig. 29. SUMO vehicle object.

8) "ArrivalPos" is the position of the vehicle when it departs the network.
9) "ArrivalSpeed" is the vehicle's speed as it departs the network.
10) "Line" is unique to public transit vehicles and is a string that defines what transit line they are following.
11) "personNumber" is the number of people in the car. Less than or equal to the "personcapacity" Attribute of the Vehicle Types.

12) "ContainerNumber" is the number of containers on a commercial vehicle. Less than or equal to the "containercapacity" Attribute of the Vehicle Types.
13) "reroute" is a toggle that allows the vehicle to make dynamic routing decisions through the network.
14) "Via" is a list of edge ids selected when rerouting is toggled on.
15) "departPosLat" is the lateral position within a lane as a vehicle enters the road network.
16) "arrivalPosLat" is the lateral position within a lane as a vehicle exits the road network.

5.2.2.2 SUMO Processes

SUMO's documentation lists and explains several Processes in the model. They are oftentimes described as behaviors or alternative models to define behaviors. Nonetheless, the Processes captured below meet the definitional requirements of Process in that they mark changes in state of the overall model.

The first Process in SUMO is the *Repeated Flow* process that generates or created vehicles with identical Attributes (save for arrival and departure times). The Characteristics that comprise this Process are depicted in Fig. 30 below and described below.

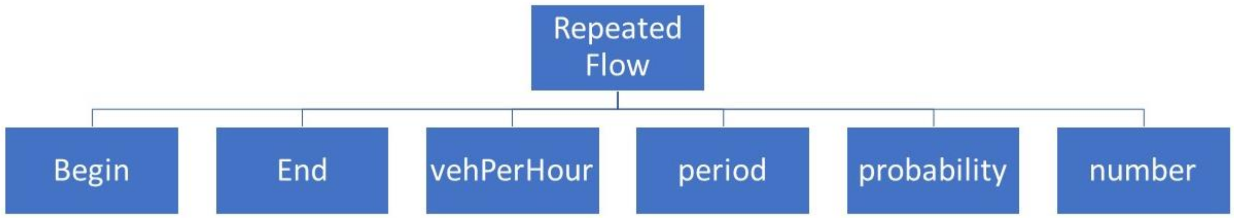


Fig. 30. SUMO repeated flow process.

1) “Begin” defines the simulation time that the first vehicle of this Process arrives in the road network.
2) “End” defines the simulation time of when this Process stops generating vehicles.
3) “vehPerHour” defines the number of vehicles generated by this Process in an hour. They are uniformly distributed. It cannot be used with either “period” or “probability” Characteristics.
4) “period” inserts equally spaced vehicles during the begin to end time period that. It cannot be used in conjunction with either “vehPerHour” or “probability” Characteristics.
5) “probability” defines the chance of a vehicle being generated at any given second. Cannot be used in conjunction with “vehPerHour” or “period” Characteristics
6) “number” is the total number of vehicles generated by this Process.

The next Process in SUMO is the lane changing model. This process determines which lane a given vehicle will chose when an edge has multiple lanes available. It also determines a vehicle’s speed when changing lanes [52]. Fig. 31 below depicts the characteristics that define this Process.

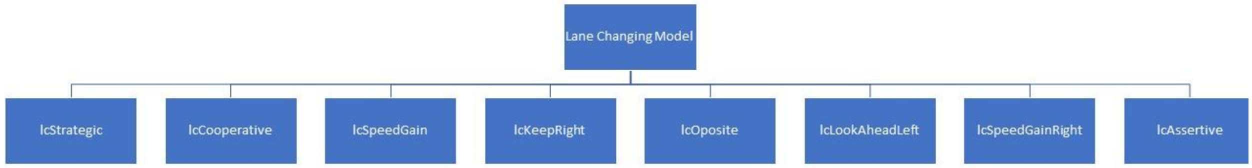


Fig. 31. SUMO lane changing process.

1) “lcStratgic” are a floating value that indicate the willingness of a vehicle to change lanes.
2) “lcCooperative” is a floating value that indicate how cooperative a vehicle is in allowing other vehicles to change lanes.
3) “lcSpeedGain” is a floating value that determines a vehicle’s willingness to change lanes in order to achieve higher speeds.
4) “lcKeepRight” is a vehicle’s desire to stay to the right-most lanes.
5) “lcOpposite” is a vehicle’s desire to pass other vehicles by changing lanes into the opposing direction of traffic.
6) “lcLookaheadLeft” is a vehicle’s decision factor for its strategic lane choice with regards to availability to change lanes in the left lane.
7) “lcSpeedGainRight” is a vehicle’s decision factor for the asymmetry of its lane-changing decision to go into either left or right lanes.
8) “lcAssertive” is a vehicle’s decision factor to accept smaller gaps between it and other vehicles when it changes lanes.

The Car Following Model is the next major Process in SUMO. This Process governs the behavior of a vehicle in the network, particularly when it is behind another vehicle. It’s responsible for changing speed attributes on vehicles in order to meet their preferences for following. The Process is decomposed into six Characteristics. The decomposition is depicted in Fig. 32 and the Characteristics are described below.

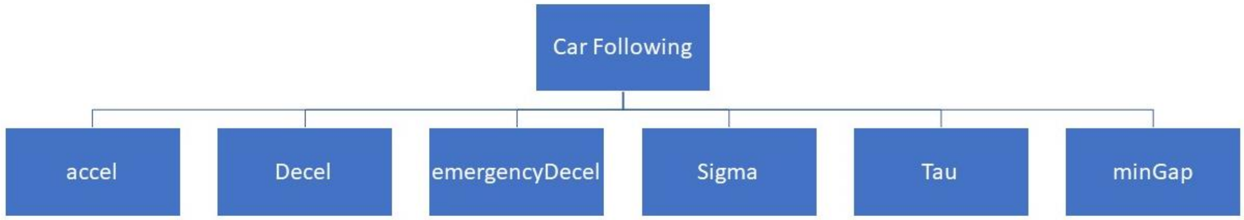


Fig. 32. SUMO car following model.

1) “accel” is the acceleration capability of the vehicle.
2) “decel” is the braking capability of the vehicle.
3) “emergencyDecel” braking capability of the vehicle in emergency situations.
4) “Sigma” is a scalar multiplier for individual drivers’ effects on their respective vehicles and ranges from 0 to 1.
5) “tau” is a scalar for individual drivers’ preference for times to stop.
6) minGap is the minimum gap in distance required in front of the vehicle.

SUMO contains a Process for user control over traffic light functionality. Identified in the traffic light Object as the programID, this traffic light control Process governs the duration of traffic signals. The traffic light Process is depicted and described below in Fig. 33. The logic presumes a user-defined actuated traffic signal; when one is not established, default timed intersections are used instead.

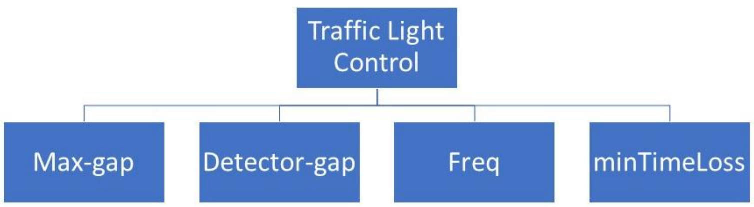


Fig. 33. SUMO traffic light control.

1) "Max-gap" is the Characteristic that the defines the length of time between vehicles passing that will allow the phase to be lengthened.
2) "Detector-gap" defines the time between the actual stopping position of the intersection and the location of a detector in any given lane.
3) "Freq" determines the interval in seconds that the program will evaluate traffic building at its junction.
4) "minTimeLoss" is the threshold for a given vehicle's loss of time in traversing the junction. It is calculated as the ratio of the vehicle's current speed to its possible max speed. When and if the delay exceeds minTimeLoss, a request for the signal to remain green is placed.

Routing in SUMO is the last major Process. This process assigns a route to a vehicle Object to allow it to traverse the road network. The breakdown of this Process and its supporting characteristics are depicted in Fig. 34 below and described as follows [52].

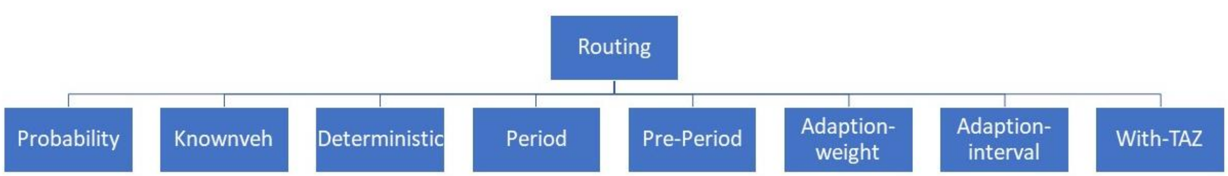


Fig. 34. SUMO routing process.

1) "Probability" is the chance that any vehicle will have a logical rerouting device associated with it.
2) "Knownveh" is a logical device that's assigned to any specific vehicle.
3) "Deterministic" is the portion of vehicles that are given a routing device.
4) "Period" is the time period that a vehicle will be routed by its device.
5) "Pre-period" is the time before a vehicle enters the network that it will consider routing options.
6) "Adaption-weight" is the weight the vehicle's prior edge.
7) "Adaption-interval" is the time interval for updating a vehicle's edge weights.
8) "With-TAZ" directs the vehicle to use a Traffic Analysis Zone (TAZ) as a routing end-point.

5.2.2.3 SUMO Relationships

As in RtePM, Relationships are not a documented concept in SUMO and are implicit in the descriptions between and among Object and Process concepts. Indeed, Relationships are likely to be only identifiable from proper model documentation and determining the dependencies from one or more Objects or Processes to other's Objects and Processes. Identifying the Relationships in SUMO is accomplished through the breakdown of the other conceptual components and noting dependencies among conceptual components can be identified. Descriptions of the Objects and Processes in the previous two subsections identify the concepts that are linked together via Relationships in SUMO. The Relationships and the other concepts that they link together are summarized in Table 34 below.

TABLE 34

SUMO RELATIONSHIPS

Concept 1	Concept 1 Type	Concept 2	Concept 2 Type
Roads	Object	Vehicle	Object
Roads	Object	Junction	Object
Connection	Object	Junction	Object
Connection / Lane	Object / Attribute	Road / Lane	Object / Attribute
Requests	Object	Junction	Object
Requests	Object	Vehicle	Object
Route	Object	Road / Lane	Object
Vehicle	Object	Car Following	Process
Vehicle Type	Object	Vehicle	Object
Vehicle Type	Object	Lane Changing	Process
Vehicle Type	Object	Person Loading	Process
Vehicle Type	Object	Container Loading	Process
Vehicle	Object	Route	Object
Vehicle	Object	Routing	Process
Repeated Flow	Process	Vehicle	Object
Repeated Flow	Process	Roads	Object
Routing	Process	Route	Object
Traffic Light Control	Process	Junction	Object
Traffic Light Control	Process	Vehicle	Object
Traffic Light Control	Process	Lane	Object/Attribute

5.3 Integration of the Models

Integrating these two models, as noted before requires a perspective of what information can conceivably be offered by SUMO's instantiation of the I-395 corridor and its surrounding roadways that is usable as input parameters into the RtePM instantiation of the larger

Washington, DC metro area. SUMO generates numerous metrics that may be of interest to RtePM modeling. An experienced domain expert in transportation analysis and engineering will recognize the utility of various metrics between the models. SUMO offers 31 different output files that are accessible by the model user, and can be broken up into the following categories: disaggregated vehicle-based information, simulated detectors, values for edges or lanes, aggregated vehicle-based information, network-based information, and traffic-lights-based information [50]. As RtePM is fundamentally concerned with the flow of vehicular traffic out of an area, metrics related to road capacities and speeds are of interest.

5.3.1 Conceptual Alignments

From sections 5.1.2 and 5.2.2, we have a listing and decomposition of concepts from both models. Applying the theory developed in section 4.2 to the concepts described will yield an assessment of alignments across the three axes of conceptual elements.

5.3.1.1 Objects' Conceptual Alignment

The first step is to calculate the alignment of individual Objects between the two models. This conceptual component is perhaps the most readily identifiable when examining models, as they are the most clearly defined in user documentation and evident in the usage of the model. There are some notable differences that can be seen simply by inspection and others that become apparent when delving deeper into model documentation. For example, the conceptual component of traffic lights exists in SUMO with some rather significant detail while there is no such similar Object in RtePM.

A side by side listing of Objects from each model are presented below in Table 35. Absent from this list are detailed descriptions of the Objects' Attributes. This straightforward inspection makes it plain that the models differ at least in scope in their Objects' alignments.

TABLE 35

CONCEPTUAL OBJECTS LISTING FOR SUMO AND RtePM

SUMO Objects	RtePM Objects
Roads	Roads
Junctions	Evacuation Zones
Connections	Shelters
Traffic Lights	Vehicles
Requests	
Routes	
Vehicle Type	
Vehicle	

By inspection, the number of Objects identified in SUMO is eight, whereas RtePM has four. The fact that there are more Objects in one model than the other is a fast indicator that there is at least a misalignment of model scope between these models. However, even though RtePM has four Objects, two of them are unique to the model: Shelters and Evacuation Zones. SUMO has six unique Objects: Junctions, Connections, Traffic Lights, Requests, Routes, and Vehicle Types. Only two similar Objects exist in each model: Vehicles and Roads. This difference in the models Objects' space further indicates that the models have a difference in scope.

Next, a comparison of the models' representations of these Objects is required. A side-by-side comparison of the Attributes that define each model's Roads Object is in Table 36 below.

TABLE 36

COMPARISON OF ROADS OBJECTS IN SUMO AND RtePM

SUMO's Roads	RtePM's Roads
To	Evacuation End Point
From	Modified Roads
Function	Lanes
Priority	FreeFlow Speed
Lanes: Index, Length, Shape	Functional Class
Speed	Speed Limit

There are noticeably different Attributes from one model to the other in their descriptions of Roads. The "To" Attribute in SUMO is similar in concept to RtePM's "Evacuation End Point," but is different in what it semantically describes. Specifically, SUMO describes another junction within the road network whereas RtePM is specifically describing a node on the outer perimeter of the road network. At first glance, "Lanes" would appear to be another similar Attribute in each Object. However, SUMO's Lanes Attribute describes the length and shape of the segment of a portion of a road and RtePM's lanes contain a simple count of the number of lanes in the Road. These two Attributes are not the same across the models, albeit similar. SUMO offers more resolution on Lanes than RtePM, meaning that there is a misalignment of

Resolution between these models. Likewise, the “Function” in SUMO describes the nature of the edge—part of an intersection, part of a road, or a specially designated pedestrian area. “Functional Class” within RtePM describes the class of only roadway: Highway, Major Artery, or Minor Artery.

The other nominally similar Object between the two models is “Vehicles.” Table 37 below lists the Attributes that describe each model’s Vehicles Object.

TABLE 37

VEHICLES OBJECT COMPARISON

SUMO’s Vehicles	RtePM’s Vehicles
Type	Population Block
Route	PercentUsingVehicles
Depart	PercentUsingTransit
departPos	VehicleSize
ArrivalLane	VehicleDesiredSpeed
ArrivalPos	Vehicle Acceleration Rate
ArrivalSpeed	
Line	
personNumber	
ContainerNumber	
Reroute	
departPosLat	
arrivalPosLat	

None of the Attributes in either model’s description of Vehicles aligns with Attributes of the other model’s description of Vehicles, presenting a misaligned scope of these two Objects in particular. Attributes defining quantities of Vehicles are present in RtePM, but not in SUMO. Other Attributes in RtePM describing Vehicles are present in other Objects in SUMO, such as Vehicle Size, Vehicle Acceleration Rate, and Vehicle Desired Speed being related to SUMO's VehicleType Object.

RtePM's Roads contain Attributes regarding speed—both Freeflow Speed and Speed Limit—which are not present in SUMO's description of Roads but are present in SUMO's descriptions of Vehicle Types. Therefore, there is also a misalignment of structure between these two models' Objects as well. Having previously identified a misalignment due to Scope and Resolution, there is now a case of the combinatorically complex Scope, Resolution, and Structure misalignment scenario.

The comparison of Attributes in both Roads and vehicles between the two models is not a binary comparison, either. As had been noted in reference [15], described in section 2.1.3, semantic differences between the models need to be assessed as well using a structured value hierarchy. The differences of the Attributes of the similar Objects needs to be assessed. The assessments yield several steps of numeric alignment values for each conceptual element.

Table 38 assesses the alignment of the Roads Object from RtePM to SUMO by mapping Attributes of RtePM's Roads to Attributes of SUMO's Roads.

These assessments show that not only are there several conceptual Attributes in RtePM that are absent in SUMO, but that even those nominally shared concepts have notable semantic differences as well. Using Wartik et al.'s method, the overall alignment of Roads as calculated from RtePM to SUMO is:

TABLE 38

ASSESSMENT OF ROADS OBJECT FROM RtePM TO SUMO

RtePM Attributes	SUMO Attributes	Alignment Assessment	Standard Phrase	Rationale
Evacuation End Point	To	25%	Low Alignment	There are limited occasions where RtePM’s end points may coincide with the defining "to" attribute of SUMO. This is coincidental and the elements do not conceptually match completely
Modified Roads: Shoulder	NA	0%	No alignment	There is no corollary in SUMO to RtePM’s modified roads, shoulder.
Modified Roads: Close	NA	0%	No alignment	There is no corollary in SUMO to RtePM’s modified roads, close(d).
Modified Roads: ContraFlow	NA	0%	No alignment	There is no corollary in SUMO to RtePM’s modified roads, contraflow.
Lanes	Lanes: index	100%	Perfect Alignment	There is an unambiguous mapping from RtePM’s Lanes to SUMO’s Lanes index
Freeflow Speed	Lanes: speed	75%	High Alignment	The models’ elements can be made to align with some effort, but RtePM’s Freeflow speed is not the same as a speed limit
Functional Class	Function	25%	Low Alignment	Any overlap is coincidental, despite the naming conventions. RtePM refers to class of the roadway, whereas SUMO only has one class of roadway, but also offers many other classes of transit
Speed Limit	Lanes: speed	100%	Perfect Alignment	There is an unambiguous mapping from RtePM’s SpeedLimit to SUMO’s Lanes Speed

$$\frac{0.25 + 0 + 0 + 0 + \ 1.0 + 0.75 + \ 0.25 + 1.0}{8} = 0.40625$$

This indicates that just over 40% of the Roads definition of RtePM is shared with SUMO. While a novice user of these models might presume that Roads are effectively the same concept, they are definitively quite misaligned concepts.

The inverse relationship, mapping the Roads Object from SUMO to RtePM yields a different alignment calculation. Table 39 below shows the alignment assessment of Roads from SUMO to RtePM.

Applying Wartik et al.'s method [15], the overall alignment of Roads as calculated from RtePM to SUMO is:

$$\frac{0.1 + 0.1 + 0.25 + 0 + \ 0.75 + 1.0 + \ 0 + 1}{8} = 0.2750$$

This indicates that only 27.5% of SUMO's semantic concept of Roads is shared in RtePM. Again, where a novice user may see these Roads concepts as similar, they are substantively different. Furthermore, SUMO has significantly more information contained in its definitions of roads, as only 27.5% is included in the RtePM definition of roads. Of important note is that the calculated assessments of this particular Object's alignment between models yields different results depending on the direction of the transaction. That is, 40.625% of RtePM's *roads* are shared with SUMO, meaning that nearly 60% of its concept of roads is unique to RtePM. At the same time, 27.5% of SUMO's concepts of roads is contained in RtePM, meaning that 72.5% of its road definition is unique to SUMO. The seemingly similar concepts of roads have important semantic differences between the models and the degree of alignment is dependent upon which model is being integrated into the other model.

The second similar Object between SUMO and RtePM is the Vehicles Object. Table 40 below depicts an assessment of Vehicles mapped from RtePM to SUMO.

TABLE 39

ROADS OBJECT ALIGNMENT ASSESSMENT FROM SUMO TO RtePM

SUMO Attributes	RtePM Attributes	Alignment Assessment	Standard Phrase	Rationale
To	Evacuation End Points	10%	Very Low Alignment	Extrapolated Value. There are limited instances where a SUMO "To" point equates to a RtePM "Evacuation End Point."
From	Evacuation Zone: Population Blocks	10%	Very Low Alignment	Extrapolated Value. The related value for RtePM is found in another Object, defining the source of vehicular traffic. This value will only align in a small set of instances where the "from" point in SUMO marks the beginning of an egress route
Function	Functional Class	25%	Low Alignment	Overlap here is coincidental. SUMO’s function allows for non-vehicular pathways (such as trails, rail, or waterway) whereas RtePM’s functional class offers greater detail on roadway types
Priority	NA	0%	No alignment	There is no corollary in RtePM for this Attribute in SUMO
Lanes: index	Lanes	75%	High Alignment	There is a high degree of overlap in the models’ Attributes, but more semantic information is included in SUMO’s representation of lanes due to its explicit ordering of lanes from left to right
Lanes: speed	Speed Limit	100%	Very High Alignment	These attributes align to represent the same concept
Lanes: length	NA	0%	No alignment	There is no corollary in RtePM for this Attribute in SUMO
Lanes: shape	NA	0%	No alignment	There is no corollary in RtePM for this Attribute in SUMO

TABLE 40

RtePM VEHICLES TO SUMO VEHICLES

RtePM Attributes	SUMO Attributes	Alignment Assessment	Standard Phrase	Rationale
Population Block	NA	0%	No alignment	There is no corollary Attribute in SUMO for the Population Block Attribute in RtePM
Population Block: Percentage Using Vehicles	NA	0%	No alignment	There is no corollary Attribute in SUMO for the Population Block Attribute in RtePM
Population Block: Percentage Using Vehicles: Percent Towing	NA	0%	No alignment	There is no corollary Attribute in SUMO for the Population Block Attribute in RtePM
Population Block: Percentage Using Public Transit	NA	0%	No alignment	There is no corollary Attribute in SUMO for the Population Block Attribute in RtePM
People per Vehicle	personNumber	75%	High alignment	These attributes are very similar in nature and can be made to align with some minor effort
Vehicle Size	NA	0%	No alignment	There is no corollary Attribute in SUMO's Vehicle Object for the Vehicle Size Attribute in RtePM
Vehicle Desired Speed	NA	0%	No alignment	There is no corollary Attribute in SUMO's Vehicle Object for the Vehicle Desired Speed Attribute in RtePM
Vehicle Acceleration Rate	NA	0%	No alignment	There is no corollary Attribute in SUMO's Vehicle Object for the Vehicle Acceleration Rate Attribute in RtePM

When the weighted average is applied, the overall alignment of Vehicles as calculated from RtePM to SUMO is:

$$\frac{0.0 + 0.0 + 0.0 + 0.0 + \ 0.75 + 0.0 + \ 0.0 + 0.0}{8} = 0.09375$$

This means that only 9.375% of RtePM's Vehicles definition is found in SUMO's definitions of vehicles. However, in this situation, there is overlap between RtePM's Vehicles and SUMO's Vehicle Type. The assessment of RtePM's Vehicles to SUMO's Vehicle Type is presented in Table 41.

RtePM's Vehicles Object then have a partial alignment with SUMO's Vehicle Type Object. Its alignment is calculated as:

$$\frac{0.0 + 0.0 + 0.0 + 0.0 + \ 0.0 + 1.0 + \ 1.0 + 1.0}{8} = 0.375$$

This indicates that RtePM's Vehicles are 37.5% aligned with SUMO's Vehicle Types. Coupled with the 9.375% overlap with SUMO Vehicles, RtePM's still have 53.125% of their definition unique to RtePM.

Table 42 assesses the relationship from SUMO's Vehicle Type to RtePM's Vehicles. SUMO having significantly more Attributes to define Vehicles than RtePM, the comparison is significantly longer.

TABLE 41

RtePM VEHICLES TO SUMO VEHICLE TYPE

RtePM Attributes	SUMO Vehicle Type Attributes	Alignment Assessment	Standard Phrase	Rationale
Population Block	NA	0%	No alignment	There is no corollary Attribute in SUMO for the Population Block Attribute in RtePM
Population Block: Percentage Using Vehicles	NA	0%	No alignment	There is no corollary Attribute in SUMO for the Population Block Attribute in RtePM
Population Block: Percentage Using Vehicles: Percent Towing	NA	0%	No alignment	There is no corollary Attribute in SUMO for the Population Block Attribute in RtePM
Population Block: Percentage Using Public Transit	NA	0%	No alignment	There is no corollary Attribute in SUMO for the Population Block Attribute in RtePM
People per Vehicle	NA	0%	No alignment	These attributes are very similar in nature and can be made to align with some minor effort
Vehicle Size	Length	100%	Perfect Alignment	There is an exact, unambiguous mapping between the models.
Vehicle Desired Speed	Desired Speed	100%	Perfect Alignment	There is an exact, unambiguous mapping between the models.
Vehicle Acceleration Rate	Accel	100%	Perfect Alignment	There is an exact, unambiguous mapping between the models.

TABLE 42

SUMO VEHICLE TYPE TO RtePM VEHICLES

SUMO Attributes	RtePM Attributes	Alignment Assessment	Standard Phrase	Rationale
Type	NA	0%	No alignment	There is no corollary Attribute in RtePM for the Vehicle Type in SUMO
Route	NA	0%	No alignment	There is no corollary Attribute in RtePM for the Route in SUMO
Depart	NA	0%	No alignment	There is no corollary Attribute in RtePM for Depart in SUMO
departLane	NA	0%	No alignment	There is no corollary Attribute in RtePM for departLane in SUMO
departPos	NA	0%	No alignment	There is no corollary Attribute in RtePM for departPos in SUMO
departSpeed	NA	0%	No alignment	There is no corollary Attribute in RtePM for departSpeed in SUMO
ArrivalLane	NA	0%	No alignment	There is no corollary Attribute in RtePM for ArrivalLane in SUMO
ArrivalPos	NA	0%	No alignment	There is no corollary Attribute in RtePM for ArrivalPos in SUMO
ArrivalSpeed	NA	0%	No alignment	There is no corollary Attribute in RtePM for ArrivalSpeed in SUMO
Line	NA	0%	No alignment	There is no corollary Attribute in RtePM for Line in SUMO
personNumber	People per Vehicle	75%	High alignment	These attributes are very similar in nature and can be made to align with some minor effort
ContainerNumber	NA	0%	No alignment	There is no corollary Attribute in RtePM for ContainerNumber in SUMO

SUMO Attributes	RtePM Attributes	Alignment Assessment	Standard Phrase	Rationale
reroute	NA	0%	No alignment	There is no corollary Attribute in RtePM for reroute in SUMO
Via	NA	0%	No alignment	There is no corollary Attribute in RtePM for Via in SUMO
departPosLat	NA	0%	No alignment	There is no corollary Attribute in RtePM for departPosLat in SUMO
arrivalPosLat	NA	0%	No alignment	There is no corollary Attribute in RtePM for arrivalPosLat in SUMO

When the weighted average is applied, the overall alignment of Vehicles as calculated from SUMO to RtePM is:

$$\frac{0.75}{16} = 0.046875$$

This means that less than 5% of SUMO's Vehicles definition is found in RtePM's definitions of vehicles, for a rather substantial difference between the models' representation of Vehicles. Most of the Attributes in SUMO's Vehicles have no corollary to RtePM's definition of Vehicles.

SUMO's Vehicle Type Object must also be compared to RtePM's Vehicle because of their shared attributes. Table 43 below assess the alignment between these model Objects.

TABLE 43

SUMO'S VEHICLE TYPE OBJECT COMPARED TO RtePM'S VEHICLE OBJECT

SUMO Attributes	RtePM Attributes	Alignment Assessment	Standard Phrase	Rationale
Accel	Vehicle Acceleration Rate	100%	Perfect Alignment	There is an exact, unambiguous mapping between the models.
Decel	NA	0%	No alignment	There is no corollary Attribute in SUMO for the Decel Attribute in RtePM
ApparentDecel	NA	0%	No alignment	There is no corollary Attribute in SUMO for the ApparentDecel Attribute in RtePM
EmergencyDecel	NA	0%	No alignment	There is no corollary Attribute in SUMO for the EmergencyDecel Attribute in RtePM
Sigma	NA	0%	No alignment	There is no corollary Attribute in SUMO for the Sigma Attribute in RtePM
Tau	NA	0%	No alignment	There is no corollary Attribute in SUMO for the Tau Attribute in RtePM
Length	Vehicle Size	100%	Perfect Alignment	There is an exact, unambiguous mapping between the models.
MinGap	NA	0%	No alignment	There is no corollary Attribute in SUMO for the MinGap Attribute in RtePM
MaxSpeed		0%	No alignment	There is no corollary Attribute in SUMO for the MaxSpeed Attribute in RtePM
SpeedFactor	Vehicle Desired Speed	100%	Perfect Alignment	There is an exact, unambiguous mapping between the models.
SpeedDev	Vehicle Desired Speed	25%	Low Degree of Alignment	There is coincidental overlap due to the additional resolution on speed variance offered in SUMO

SUMO Attributes	RtePM Attributes	Alignment Assessment	Standard Phrase	Rationale
VClass	NA	0%	No alignment	There is no corollary Attribute in SUMO for the VClass Attribute in RtePM
EmissionClass	NA	0%	No alignment	There is no corollary Attribute in SUMO for the EmissionClass Attribute in RtePM
Width	NA	0%	No alignment	There is no corollary Attribute in SUMO for the Width Attribute in RtePM
LaneChangeMode l	NA	0%	No alignment	There is no corollary Attribute in SUMO for the LaneChangeModel Attribute in RtePM
CarFollowModel	NA	0%	No alignment	There is no corollary Attribute in SUMO for the CarFollowModel Attribute in RtePM
PersonCapacity	NA	0%	No alignment	There is no corollary Attribute in SUMO for the PersonCapacity Attribute in RtePM
ContainerCapacity	NA	0%	No alignment	There is no corollary Attribute in SUMO for the ContainerCapacity Attribute in RtePM
BoardingDuration	NA	0%	No alignment	There is no corollary Attribute in SUMO for the BoardingDuration Attribute in RtePM
LoadingDuration	NA	0%	No alignment	There is no corollary Attribute in SUMO for the LoadingDuration Attribute in RtePM
LatAlignment	NA	0%	No alignment	There is no corollary Attribute in SUMO for the LatAlignment Attribute in RtePM
MinGapLat	NA	0%	No alignment	There is no corollary Attribute in SUMO for the MinGapLat Attribute in RtePM

SUMO Attributes	RtePM Attributes	Alignment Assessment	Standard Phrase	Rationale
MaxSpeedLat	NA	0%	No alignment	There is no corollary Attribute in SUMO for the MaxSpeedLat Attribute in RtePM
ActionStepLength	NA	0%	No alignment	There is no corollary Attribute in SUMO for the ActionStepLength Attribute in RtePM

The overall alignment of SUMO's Vehicle Types to RtePM's Vehicles is:

$$\frac{3.25}{24} = 0.135$$

This means that just over 13.5% of the definition of Vehicle Types of SUMO is found in some fashion in RtePM's definition of Vehicles.

These assessments may be useful in their own right, but to calculate the entire alignment difference of Objects between these models, all Objects need to be accounted for. Knowing that these models' Objects have been shown to be misaligned in Scope, Structure, and Resolution, the calculation of alignment recognizes that there are common Attributes outside of the shared space of model alignment. Specifically, "Acceleration" is included in SUMO's "VehicleType" and on RtePM's "Vehicles," which outside the shared scope-aligned space, which includes only Roads and vehicles. From section 4.2.1.7, the general form of the Objects' alignment is

$$Scope\ Alignment\ Term\ +\ Structure\ Alignment\ Term\ +\ Resolution\ Alignment\ Term$$

Scope Alignment Term:

$$((Roads\ Alignment\ +\ Vehicles\ Alignment)\ /\ All\ RtePM)\ \times\ ((Roads\ +\ Vehicles)\ /\ All\ SUMO)\ =\ ((0.40625\ +\ 0.09375) \div 4)\ \times\ (2 \div 8)\ =\ 0.03125$$

Structure Alignment Term:

$$(RtePM\ Vehicle\ |\ SUMO\ Vehicle\ Type)\ \times\ (Scope\ Alignment\ Term)\ =\ 0.375\ \times\ 0.03125\ =\ 0.11719$$

Misaligned Resolution Term:

$$(Scope\ Alignment\ Term)\ \times\ (SUMO\ Resolution\ Ratios)\ =\ 0.03125\ \times\ 0.3333\ =\ 0.010417$$

$$Total\ Object\ Alignment\ from\ RtePM\ to\ SUMO\ =\ 0.03125\ +\ 0.011719\ +\ 0.010417\ =\ 0.053385$$

From these calculations, the alignment of these two models' Objects is calculated to be 5.34%, meaning that their misalignment is 1 – 0.053385, or 0.946615, or approximately 94.66%.

The alignment metric can be calculated from the perspective of SUMO to RtePM, as well. This follows the same general form of:

$$Scope\ Alignment\ Term\ +\ Structure\ Alignment\ Term\ +\ Resolution\ Alignment\ Term$$

Scope Alignment Term:

$$((Roads\ Alignment\ +\ Vehicles\ Alignment)\ /\ All\ SUMO)\ \times\ ((Roads\ +\ Vehicles)\ /\ All\ RtePM)\ =\ ((0.275\ +\ 0.046875)/8)\ \times\ (2/4)\ =\ 0.020117$$

Structure Alignment Term:

$$(SUMO\ Vehicle\ Type \mid RtePM\ Vehicle) \times (Scope\ Alignment\ Term)$$

$$= 0.135417 \times 0.020117 = 0.002724$$

Resolution Alignment Term:

$$(Scope\ Alignment\ Term) \times (SUM\ Resolution\ Ratios) = 0.020117 \times 0.3333$$

$$= 0.000908$$

Total Object alignment, from SUMO to RtePM is $0.020117 + 0.002724 + 0.000908 = 0.023749$, or just over 2.37% aligned. Their misalignment then is $1 - 0.023749 = .976251$, or 95.63%.

5.3.1.2 Processes' Conceptual Alignment

The next major step of the integration risk tuple is to assess the alignment of the two models' Processes. A side by side comparison of the two sets of Processes in SUMO and RtePM makes several differences between the two models readily apparent.

Table 44 below lists the Processes previously identified in both models. A preliminary inspection shows that each RtePM and SUMO contain contains five Processes. Each model contains a Process that is unique to it: RtePM has Congestion Modeling and SUMO has Lane Changing Model. This uniqueness indicates that Processes are at least misaligned in scope.

TABLE 44

MODELS' PROCESSES LIST

RtePM Processes	SUMO Processes
Congestion Modeling	Lane Changing Model
Traffic Signal Phasing	Traffic Light Control
Car Following	Car Following
Vehicle Entry	Repeated Flow
Path Determination	Routing

The four semantically similar Processes are compared in turn below.

First, RtePM's Traffic Signal Phasing would seem to map well to the traffic Light Control Process in SUMO. However, the exploration of each model's Characteristics of these Processes reveal that they are not aligned at all. RtePM has two identified Characteristics in its Traffic Signal Phasing Process while SUMO contains four. These two Processes would be misaligned by scope at least, but an assessment of the alignment from RtePM to SUMO shows that neither of its Characteristics are found in SUMO. This alignment assessment is presented in Table 45 below.

TABLE 45

RtePM TRAFFIC SIGNAL PHASING TO SUMO TRAFFIC SIGNAL CONTROL

RtePM Characteristics	SUMO Characteristics	Alignment Assessment	Standard Phrase	Rationale
Sets of Vehicular Direction	NA	0%	No Alignment	There is not a similar Characteristic in SUMO
Proportion of inbound traffic flow	NA	0%	No Alignment	There is not a similar Characteristic in SUMO

Similarly, Table 46 below depicts an alignment assessment from SUMO to RtePM and it can be seen that the four Characteristics of SUMO's traffic light control do not map to RtePM.

Comparisons of alignment of traffic signal controls or phasing from one model to the other, regardless of perspective yields zero; these Processes are not aligned despite the seemingly common naming convention.

The next pair of Processes across the two models to map to one another are the Car Following Models of each. These two Processes are at a minimum different in scope due to the unique Characteristics in each model's Process. The Characteristics of each have been previously identified in sections 5.1.2.2 and 5.2.2.2.

Comparing the Characteristics of RtePM's Car Following Model's four Characteristics to SUMO's Car Following model is depicted in Table 47 below.

TABLE 46

SUMO TRAFFIC LIGHT CONTROL TO RtePM TRAFFIC LIGHT PHASING

SUMO Characteristics	RtePM Characteristics	Alignment Assessment	Standard Phrase	Rationale
Max-gap	NA	0%	No Alignment	There is not a similar Characteristic in RtePM
Detector-gap	NA	0%	No Alignment	There is not a similar Characteristic in RtePM
Freq	NA	0%	No Alignment	There is not a similar Characteristic in RtePM
minTimeLoss	NA	0%	No Alignment	There is not a similar Characteristic in RtePM

TABLE 47

RtePM CAR FOLLOWING PROCESS TO SUMO CAR FOLLOWING PROCESS

RtePM Characteristics	SUMO Characteristics	Alignment Assessment	Standard Phrase	Rationale
Max Allowed Speed	NA	0%	No Alignment	There is no similar characteristic in SUMO
Max Allowed Speed: Freeflow Speed	NA	0%	No Alignment	There is no similar characteristic in SUMO
Max Allowed Speed: Acceleration Rate	accel	100%	Very high alignment	Minimal interpretation is needed for these Characteristics
Fixed Following Distance	minGap	100%	Very high alignment	Minimal interpretation is needed for these Characteristics

Using the value hierarchy discussed previously, the overall alignment of the car following model from RtePM to SUMO can be calculated as follows.

$$\frac{0.0 + 0.0 + 1.0 + 1.0}{4} = 0.5$$

The alignment of the car following process from RtePM to SUMO is only 50%, despite a common naming convention.

Inversely, the assessment of the alignment of the car following model from SUMO to RtePM is presented in Table 48 below.

TABLE 48

CAR FOLLOWING MODEL FROM SUMO TO RtePM

SUMO Characteristics	RtePM Characteristics	Alignment Assessment	Standard Phrase	Rationale
accel	Max Allowed Speed: Acceleration Rate	100%	Very high alignment	Minimal interpretation is needed for these Characteristics
decel	NA	0%	No Alignment	There is no similar characteristic in RtePM
emergencyDecel	NA	0%	No Alignment	There is no similar characteristic in RtePM
Sigma	NA	0%	No Alignment	There is no similar characteristic in RtePM
Tau	NA	0%	No Alignment	There is no similar characteristic in RtePM
minGap	Fixed Following Distance	100%	Very high alignment	Minimal interpretation is needed for these Characteristics

Assessing the overall alignment of these car following Process from SUMO to RtePM,

$$\frac{1.0 + 0.0 + 0.0 + 0.0 + 0.0 + 1.0}{6} = 0.333$$

Table 49 below compares the next pair of Processes, RtePM's Vehicle Entry Process with SUMO's Repeated Flow Process. These two Processes differ in their scope, simply by tabulating the number of Characteristics in each. The difference in scope is bigger, however, because both Processes have several Characteristics that are unique to their respective models. Comparing the Characteristics from RtePM's Vehicle Entry to SUMO's Repeated Flow Process yields the assessment below. RtePM's Process assigns vehicles to the system from their entry point to the road network for the duration of the system, whereas SUMO allows for more granular control, allowing the repeated flows to be controlled for discrete periods of time and through alternative statistical methods.

Misalignments of resolution and structure also become apparent when examining RtePM's Departure Rate, Path Allocation, and Path Allocation: Vehicles loaded Characteristics; they map to mutually exclusive concepts within SUMO's process. Assessing the overall alignment of these Process from RtePM to SUMO,

$$\frac{0.25 + 0.25 + 0.25 + 0.0 + 0.0 + 0.0}{6} = 0.125$$

Only 12.5% of RtePM's Vehicle Entry Process is captured in SUMO's repeated flow Process. Once again, the inverse of this relationship is presented. Table 50 below depicts the assessment of Characteristics from SUMO's Repeated Flow to RtePM's Vehicle Entry Process. A misalignment of resolution is strongly apparent. All of SUMO's Characteristics are partially aligned to only RtePM's Departure Rate Characteristic.

TABLE 49

RtePM VEHICLE ENTRY PROCESS AND SUMO REPEATED FLOW PROCESS

RtePM Characteristics	SUMO Characteristics	Alignment Assessment	Standard Phrase	Rationale
Departure Rate	vehPerHour	25%	Low Degree of Alignment	Attributes in SUMO partially fulfill the Departure Rate in RtePM, but do not meet all of RtePM's departure rate. These three characteristics are mutually exclusive within SUMO define departure mechanisms from given points within the system for fixed periods of time.
Path Allocation	probability	25%	Low Degree of Alignment	
Path Allocation: Vehicles Loaded	number	25%	Low Degree of Alignment	
Path Allocation	NA	0%	No Alignment	SUMO does not set the Path for vehicles here; it is a separate process (Routing).
Path Allocation: Vehicles Loaded	NA	0%	No Alignment	There is no corollary Characteristic in SUMO to match RtePM's Path Allocation Characteristic of vehicles loaded.
Path Allocation: Shortest Path	NA	0%	No Alignment	There is no corollary Characteristic in SUMO to match RtePM's Path Allocation Characteristic of shortest path.

TABLE 50

SUMO REPEATED FLOW TO RtePM VEHICLE ENTRY PROCESSES

SUMO Characteristics	RtePM Characteristics	Alignment Assessment	Standard Phrase	Rationale
begin	Departure Rate	25%	Low Degree of Alignment	SUMO uses stochastic methods to determine departures. It is best compared to RtePM's departure rate, but SUMO is fundamentally more detailed, allowing for alternative denture methods and to control beginning and ending times of the process.
end	Departure Rate	25%	Low Degree of Alignment	
vehPerHour	Departure Rate	25%	Low Degree of Alignment	
period	Departure Rate	25%	Low Degree of Alignment	
probability	Departure Rate	25%	Low Degree of Alignment	

The overall alignment in this direction of the comparison is 25%.

$$\frac{0.25 + \ 0.25 + 0.25 + 0.25 + 0.25}{5} = 0.25$$

The fourth Process pairing to be assessed is RtePM's Path Determination to SUMO's Routing Processes. Table 51 below depicts the alignment assessment of RtePM's Path Determination Characteristics to SUMO's Routing. These two Processes are misaligned given their scope, which is perhaps unsurprising given that RtePM establishes a Path Determination when a vehicle enters the system and SUMO has dynamic routing that can be applied to subsets of vehicles at different points in the simulation.

TABLE 51

RtePM PATH DETERMINATION TO SUMO ROUTING

RtePM Characteristics	SUMO Characteristics	Alignment Assessment	Standard Phrase	Rationale
Access Points	NA	0%	No alignment	there is no similar characteristic in SUMO
End Points	with-TAZ	50%	Medium degree of alignment	There is a significant difference between the models, but each of these Characteristics signifies destinations of their respective algorithms
A* Algorithm	NA	0%	No alignment	there is no similar characteristic in SUMO
A* Algorithm: Ratio: Length of path / Veh speed	adaption-weight	50%	Medium degree of alignment	There is a significant difference between the models, but each of these Characteristics weights the road/road segments the vehicles traverse
A* Algorithm: Capacity of segment	NA	0%	No alignment	there is no similar characteristic in SUMO

The overall alignment in this direction of the comparison is 20%.

$$\frac{0.0 + \ 0.5 + 0.0 + 0.5 + 0.0}{5} = 0.2$$

The inverse comparison when SUMO’s Routing is compared to RtePM’s Vehicle Path Determination is presented in Table 52 below. As evidenced by several unique Characteristics, the two Process differ in their scope.

TABLE 52

SUMO ROUTING TO RtePM PATH DETERMINATION

SUMO Characteristics	RtePM Characteristics	Alignment Assessment	Standard Phrase	Rationale
probability	NA	0%	No alignment	There is no similar characteristic in RtePM
knowveh	NA	0%	No alignment	There is no similar characteristic in RtePM
deterministic	NA	0%	No alignment	There is no similar characteristic in RtePM
period	NA	0%	No alignment	There is no similar characteristic in RtePM
pre-period	NA	0%	No alignment	There is no similar characteristic in RtePM
adaption-weight	A* Algorithm: Ratio: Length of path / Veh speed	50%	Medium degree of alignment	There is a significant difference between the models, but each of these Characteristics weights the road/road segments the vehicles traverse
adaption-interval	NA	0%	No alignment	There is no similar characteristic in RtePM
with-TAZ	End Points	50%	Medium degree of alignment	There is a significant difference between the models, but each of these Characteristics signifies destinations of their respective algorithms

The overall alignment in this direction of the comparison is 12.5%.

$$\frac{0.0 + \ 0.0 + 0.0 + 0.0 + 0.0 + 0.5 + 0.0 + 0.5}{8} = 0.125$$

As noted earlier, there is also a misaligned structure between these models. Specifically, RtePM's Vehicle Entry Process has Characteristics that can be compared and assessed against another Process in SUMO: Routing. Similar to Table 52 above, which compares RtePM's Vehicle Entry to SUMO's Repeated Flow, Table 53 below compares these RtePM's Vehicle Entry to SUMO's Routing.

The overall alignment in this between these Processes is 25%.

$$\frac{0.0 + \ 0.5 + 0.0 + 0.0 + 0.75}{5} = 0.25$$

The inverse relationship of SUMO's Routing to RtePM's Vehicle Entry is assessed below in Table 54.

The overall alignment in this between these Processes is 6.25%.

$$\frac{0.0 + \ 0.0 + 0.0 + 0.0 + 0.0 + 0.5 + 0.0}{8} = 0.0625$$

TABLE 53

RtePM VEHICLE ENTRY TO SUMO ROUTING

RtePM Attributes	SUMO Attributes	Alignment Assessment	Standard Phrase	Rationale
Departure Rate	NA	0%	No Alignment	
Path Allocation	NA	50%	Medium Degree of Alignment	SUMO's routing algorithm uses weighted shortest path methods and offers other controls on the routing algorithm
Path Allocation: Vehicles Loaded	NA	0%	No Alignment	There is no corollary Characteristic in SUMO to match RtePM's Path Allocation Characteristic of vehicles loaded
Path Allocation: Shortest Path	NA	0%	No Alignment	There is no corollary Characteristic in SUMO to match RtePM's Path Allocation Characteristic of shortest path
Path Allocation: Weighted Shortest Path	adaption-weight	75%	High Degree of Alignment	SUMO uses a similar concept for routing, but is useable within the bounds of the simulation rather than solely as an exit mechanism

TABLE 54

SUMO ROUTING TO RtePM VEHICLE ENTRY

SUMO Characteristics	RtePM Characteristics	Alignment Assessment	Standard Phrase	Rationale
probability	NA	0%	No alignment	There is no similar characteristic in RtePM
knowveh	NA	0%	No alignment	There is no similar characteristic in RtePM
deterministic	NA	0%	No alignment	There is no similar characteristic in RtePM
period	NA	0%	No alignment	There is no similar characteristic in RtePM
pre-period	NA	0%	No alignment	There is no similar characteristic in RtePM
adaption-weight	Path Allocation: Weighted Shortest Path	50%	Medium degree of alignment	RtePM also uses an A* algorithm to determine weightings for shortest paths, but it is only applied when the vehicle enters the network.
adaption-interval	NA	0%	No alignment	There is no similar characteristic in RtePM
with-TAZ	NA	0%	No alignment	There is no similar characteristic in RtePM

Again, these Process passements might have utility in understanding the models' differences on their own, but as in the case of Objects, an overall alignment assessment can be calculated from these individual alignments. These assessments may be useful in their own right, but to calculate the entire alignment difference of Objects between these models, all Objects need to be accounted for. These Processes have been misaligned in Scope, Structure, and Resolution. From section 4.2.2.7, the general form of the Process' alignment is:

$$Scope\ Alignment\ Term + Structure\ Alignment\ term + Resolution\ Alignment\ Term$$

Scope Alignment Term:

$$((Traffic\ Lights + Car\ Following + Vehicle\ Entry + Path\ Determination)\ /\ All\ RtePM\ Processes) \times ((Traffic\ Lights + Car\ Following + Repeated\ Flow + Routing)\ /\ All\ SUMO\ Processes)$$

$$= ((0.0 + 0.5 + 0.125 + 0.2) \div 5) \times (4/5) = 0.132$$

Structure Alignment Term:

$$(RtePM\ Vehicle\ Entry \mid SUMO\ Routing) \times (Scope\ Alignment\ Term) = 0.25 \times 0.132 = 0.0264$$

Misaligned Resolution Term:

$$(Scope\ Alignment\ Term) \times (SUMO\ Resolution\ Ratios) \times (Vehicle\ Entry\ to\ Repeated\ Flow\ alignment)$$

$$= 0.132 \times 0.3333 \times 0.25 = 0.010999$$

Total Process Alignment from RtePM to SUMO = $0.132 + 0.0264 + 0.010999 = 0.169339$.

From these calculations, the alignment of these two models is calculated to be 16.93%, meaning that their misalignment is 1 – 0.169339, or 0.830601, approximately 83.06%.

The alignment metric can be calculated from the perspective of SUMO to RtePM, as well. This follows the same general form of:

$$Scope\ Alignment\ Term\ +\ Structure\ Alignment\ Term\ +\ Resolution\ Alignment\ Term$$

Scope Alignment Term:

$$((Traffic\ Lights\ +\ Car\ Following\ +\ Repeated\ Flow\ +\ Routing) \div All\ SUMO\ Processes) \times ((Traffic\ Lights\ +\ Car\ Following + Vehicle\ Entry\ +\ Path\ Determination) \div All\ RtePM\ Processes)$$

$$= ((0.0\ +\ 0.333\ +\ 0.25\ +\ 0.125) \div 5) \times (4 \div 5)\ =\ 0.11333$$

Structure Alignment Term:

$$(SUMO\ Routing\ |\ RtePM\ Vehicle\ Entry) \times (Scope\ Alignment\ Term)$$

$$=\ 0.0625\ \times\ 0.11333\ =\ 0.002724$$

Resolution Alignment Term:

$$(Scope\ Alignment\ Term) \times (SUMO\ Resolution\ Ratios) \times (Repeated\ Flow\ to\ Vehicle\ Entry\ Alignment)$$

$$=\ 0.11333 \times\ (0.2)\ \times\ (0.25)\ =\ 0.005667.$$

Total Process alignment, from SUMO to RtePM is $0.11333\ +\ 0.002724\ + 0.005667\ =\ 0.121721$, or 12.17% aligned. Their misalignment then is $1 -\ 0.121721\ = 0.878279$, or 87.83%.

5.3.1.3 Relationships' Conceptual Alignment

The last element of the risk tuple is the assessment of the Relationships alignment. Relationships are notably different than either the Objects or the Processes concepts of the models' structures. Relationships are dependent on the existence of other concepts by their very nature. Relationships are defined as linkages between two other conceptual components. Relationships' alignments are therefore treated as binary; either they link together the same two similar concepts or do they do not. Having previously identified the conceptual components of each model, to include the Relationships, a comparison of the relationships is straightforward. Table 55 below depicts a comparison of the Relationships from RtePM to SUMO, identifying those Relationships in SUMO that link similar concepts there.

TABLE 55

RtePM TO SUMO RELATIONSHIPS

RtePM Relationships		SUMO Relationships		Alignment Assessment
Concept 1	Concept 2	Concept 1	Concept 2	
Roads	Vehicle	Roads	Vehicle	1.0
Roads	Shelter	#N/A	#N/A	0.0
Roads	Signal Phasing	Traffic Light Control	Lane	1.0
Evacuation Zone	Vehicle	#N/A	#N/A	0.0
Vehicle	Signal Phasing	Traffic Light Control	Vehicle	1.0
Congestion	Signal Phasing	#N/A	#N/A	0.0
Congestion	Roads	#N/A	#N/A	0.0
Vehicle Entry	Vehicle	Repeated Flow	Vehicle	1.0
Vehicle Entry	Population Block	#N/A	#N/A	0.0
Vehicle Entry	Roads	Repeated Flow	Roads	1.0
Car Following Model	Vehicle	Traffic Light Control	Junction	1.0
Car Following Model	Freeflow Speed	#N/A	#N/A	0.0
Path Determination	Roads	#N/A	#N/A	0.0
Path Determination	Vehicles	Vehicle	Routing	1.0
Path Determination	Entry Point	#N/A	#N/A	0.0
Path Determination	Evacuation Point	#N/A	#N/A	0.0

Applying the same weighting as in the other conceptual alignments before, with the notable difference that there are no partial alignments.

$$\frac{1.0 + 0.0 + 1.0 + 0.0 + 1.0 + 0.0 + 0.0 + 1.0 + 0.0 + 1.0 + 1.0 + 0.0 + 0.0 + 1.0 + 0.0 + 0.0}{16}$$

$$= 0.4375$$

The alignment value of the Relationships between the two models from the starting point of RtePM is 43.75% and the misalignment is 56.25%. The inverse alignment from SUMO to RtePM is listed in Table 56 below.

Averaging the total value of these assessments yields an overall alignment of 35% from SUMO to RtePM or a 65% misalignment.

TABLE 56

RELATIONSHIPS ALIGNMENTS FROM SUMO TO RtePM

SUMO Relationships		RtePM Relationships		Alignment Assessment
Concept 1	Concept 2	Concept 1	Concept 2	
Roads	Vehicle	Roads	Vehicle	1.00
Roads	Junction	#N/A	#N/A	0.00
Connection	Junction	#N/A	#N/A	0.00
Connection / Lane	Road / Lane	#N/A	#N/A	0.00
Requests	Junction	#N/A	#N/A	0.00
Requests	Vehicle	#N/A	#N/A	0.00
Route	Road / Lane	#N/A	#N/A	0.00
Vehicle	Car Following	Roads	Vehicle	1.00
Vehicle Type	Vehicle	#N/A	#N/A	0.00
Vehicle Type	Lane Changing	#N/A	#N/A	0.00
Vehicle Type	Person Loading	#N/A	#N/A	0.00
Vehicle Type	Container Loading	#N/A	#N/A	0.00
Vehicle	Route	#N/A	#N/A	0.00
Vehicle	Routing	Roads	Vehicle	1.00
Repeated Flow	Vehicle	Vehicle Entry	Vehicle	1.00
Repeated Flow	Roads	Roads	Vehicle	1.00
Routing	Route	#N/A	#N/A	0.00
Traffic Light Control	Junction	#N/A	#N/A	0.00
Traffic Light Control	Vehicle	Congestion	Roads	1.00
Traffic Light Control	Lane	Roads	Signal Phasing	1.00

5.4 Risk Assessment

With these alignment values between both SUMO and RtePM and again from RtePM to SUMO, an overall conceptual alignment score between the two models is readily calculatable. The alignment is calculated as:

$$
\begin{aligned}
P(alignment) &= p(alignment\ of\ Objects) \cup p(alignment\ of\ Processes) \\
&\cup\ p(alignment\ of\ Relationships) \\
&= p(alignment\ of\ Objects) + p(alignment\ of\ Processes) \\
&+ p(alignment\ of\ Relationships) \\
&- p(alignment\ of\ Objects) \times p(alignment\ of\ Processes) \\
&- p(alignment\ of\ Objects) \times p(alignment\ of\ Relationships) \\
&- p(alignment\ of\ Processes) \times p(alignment\ of\ Relationships) \\
&+ p(alignment\ of\ Objects) \times p(alignment\ of\ Processes) \\
&\times p(alignment\ of\ Relationships).
\end{aligned}
$$

This formula yields a conceptual alignment value between the two models as shown in Table 57.

TABLE 57

CONCEPTUAL ALIGNMENTS

	O	P	R	Total
SUMO -> RtePM	0.0237	0.1217	0.3500	0.4427
RtePM -> SUMO	0.0534	0.1694	0.4375	0.5577

This means that 44.27% of SUMO's concepts are included somewhere in RtePM and that 55.77% of RtePM's concepts are included somewhere in SUMO.

The values Objects, Processes, and Relationships can be placed into the Risk Tuple to quantify the risk to consistent and valid model results. The general form of the risk function from section 4.5 is:

$$R = \langle S, D(Obj), D(Proc), D(Rel), D(MOE), D(MOP) \rangle \quad (16)$$

A modeling and simulation study can be informed by any number of measures of effectiveness or measures of performance that will be collectively referred to as metrics. Model metrics are expected to change upon the integration of a model. The germane metrics will be particular to individual instantiations of modeling and simulation studies. For the purposes of a risk assessment, it is important to note that those metrics can be influenced by the integration of a model. To assess the risk, permutations of metric changes were examined for an increasing number of output metrics. Sets of two, three, four, five, six, and seven output metrics were used. Using the value hierarchy table presented in section 4.3—and repeated here in Table 58 for convenience—each metric was assigned an impact and used to calculate the overall integration risk. There are nine total levels of impact for each metric. In instances with two, three, four, and five output metrics, a full factorial design of experiments was possible to assess each permutation of metrics' values. In the instances with six and seven metrics, the number of permutations becomes infeasible to manage, and random values are assigned. This section presents each set of output metrics in turn.

TABLE 58

VALUE HIERARCHY FOR MODEL METRICS

Index	Change in Metrics' Values	New Attributes in the Metrics	Conflicting Attributes	Preference Order	Upper bound of level	Centroid Weighting
1	Minor	No	NA	1	0.111	0.056
2	Minor	Yes	No	2	0.222	0.167
3	Minor	Yes	Yes	3	0.333	0.278
4	Moderate	No	NA	4	0.444	0.389
5	Moderate	Yes	No	5	0.556	0.500
6	Moderate	Yes	Yes	6	0.667	0.611
7	Significant	No	NA	7	0.778	0.722
8	Significant	Yes	No	8	0.889	0.833
9	Significant	Yes	Yes	9	1.000	0.944

In the case of two output metrics, there are a total of 81 possible permutations, where each of the two metrics was assigned each of the possible weightings. The combinatoric effect of the output metrics was calculated using the general formula:

$$P(metrics) = p(change\ in\ MOE_1) \cup p(change\ in\ MOE_2) \cup \ldots \cup p(change\ in\ MOE_2)$$
$$\cup\ p(change\ in\ MOE_1) \cup p(change\ in\ MOE_2) \cup \ldots \cup p(change\ in\ MOE_n)$$

A brief sample of combinations is shown in Table 59.

TABLE 59

TWO METRICS FACTORIAL TABLE, SAMPLE

Metric A Level	Metric B Level	Metric A Value	Metric B Value	Total
1	5	0.056	0.500	0.528
1	6	0.056	0.611	0.633
1	7	0.056	0.722	0.738
1	8	0.056	0.833	0.843
1	9	0.056	0.944	0.948
2	1	0.167	0.056	0.213
2	2	0.167	0.167	0.306
…	…	…	…	…
4	8	0.389	0.833	0.898
4	9	0.389	0.944	0.966
5	1	0.500	0.056	0.528

With the calculated alignments of the models and with permutations of metrics' changes, an overall risk curve is calculated with the formula $Integration\ Risk = p(misalignment) \times p(metrics) \times [1 - p(misalignment) + p(misalignment) \times p(metrics)]$ using each combinatoric value of metrics. The curve is depicted in Fig. 35. Shaded regions of this graph correspond with the shaded regions of the three-dimensional graph shown in section 4.5. Lower integration risk is in the green region, with an integration score less than 20%, moderately low risk is shown in the blue region between 20 and 40%, and moderate risk is shown in the yellow region between 40 and 60%. In addition, 21% of possible risk value reside in the green low-risk region, 52% in the blue moderate-low region, and 27% in the yellow moderate region.

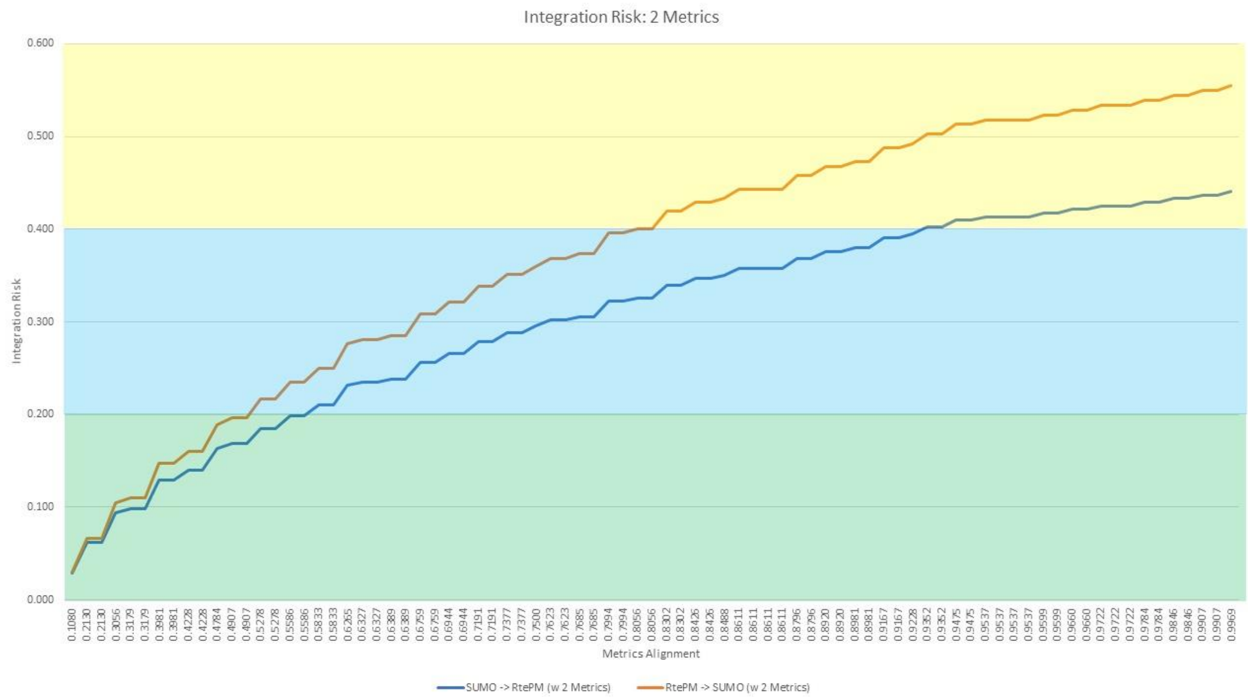


Fig. 35. Risk curves with two metrics.

In the case of three output metrics there are 9^3 permutations of metrics levels, or 729. This was manageable in a full design of experiments. A small sample of this DOE is shown below in Table 60.

Using the total metrics value, the integration risk was plotted against all 729 possibilities and depicted in Fig. 36 below. The same shaded regions depict low, low-moderate, and moderate integration risks. Approximately 5% of the values reside in the green low risk region, 45% in the blue low-moderate region, and 50% in the moderate region.

TABLE 60

THREE METRICS FACTORIAL TABLE, SAMPLE

Metric A Level	Metric B Level	Metric C Level	Metric A Value	Metric B Value	Metric C Value	Total
1	4	8	0.056	0.389	0.833	0.904
1	4	9	0.056	0.389	0.944	0.968
1	5	1	0.056	0.500	0.056	0.554
1	5	2	0.056	0.500	0.167	0.606
…	…	…	…	…	…	…
4	9	8	0.389	0.944	0.833	0.994
4	9	9	0.389	0.944	0.944	0.998
5	1	1	0.500	0.056	0.056	0.554
5	1	2	0.500	0.056	0.167	0.606
5	1	3	0.500	0.056	0.278	0.659
5	1	4	0.500	0.056	0.389	0.711

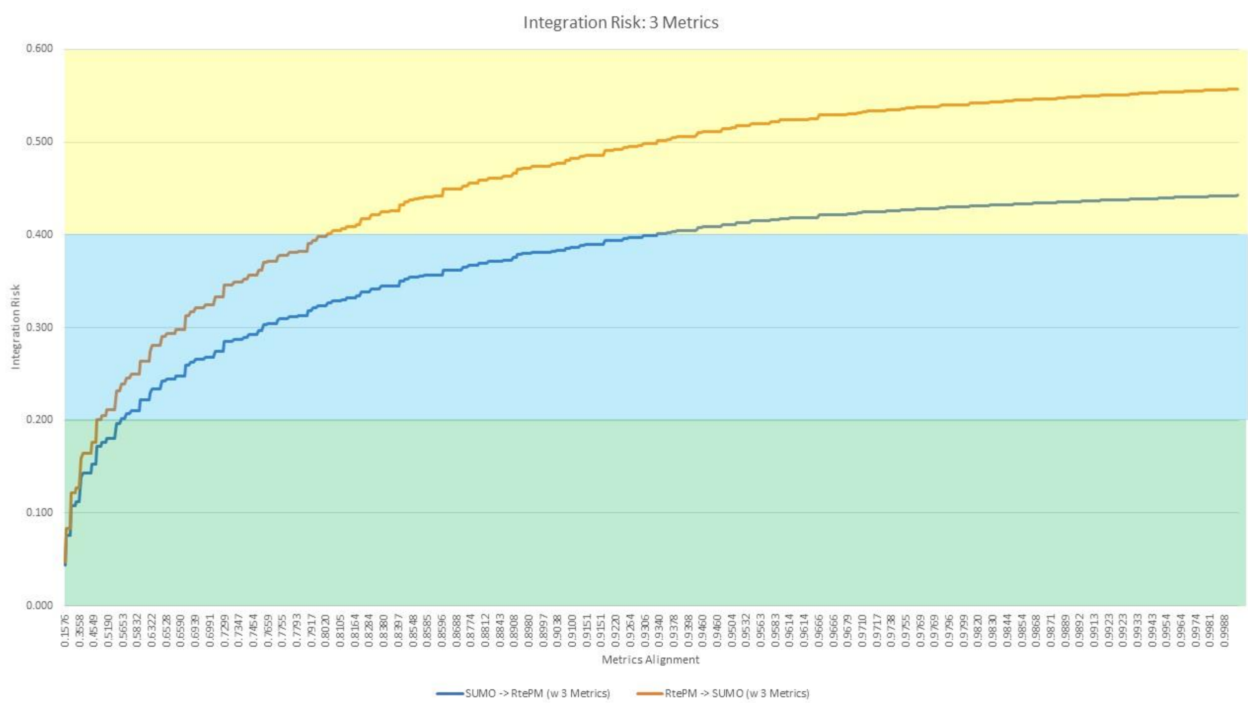


Fig. 36. Three metrics risk curves.

In the case of four output metrics there are 9^4 permutations of metrics levels, or 6561. This was manageable in a full design of experiments. A small sample of this DOE is shown below in Table 61 with just the values from the hierarchy table.

Using the total metrics values, the integration risk was plotted against all 6561 possibilities and depicted in Fig. 37 below. The same shaded regions depict low, low-moderate, and moderate integration risks. Approximately 1% of the values reside in the green low risk region, 28% in the blue low-moderate region, and 72% in the moderate yellow region.

Five output metrics have 9^5 permutations of metrics levels, or 59,049. While a large set, all permutations of metrics levels can be calculated. A small sample of this factorial table is shown below in Table 62.

All 59,049 permutations were plotted to generate the curves show in Fig. 38 below. Again, using the same shading, 0.2% of the possible risk assessments of SUMO to RtePM exist in the green low risk region, 13.4% in the blue low-moderate risk region, and 86.4% in the yellow moderate region.

With six output metrics, there are 9^6 permutations of metrics levels, or 531,441. This set became impractical to calculate all permutations, so a random sample of 50,000 permutations was generated by selecting uniformly randomly a level of 1 to 9 for each metric and looking up that level's value. A brief sample set is show below in Table 63.

All 50,000 permutations were plotted to generate the curves show in Fig. 39 below. With the same shading pattern, an extremely small portion of the sample set—0.02%—exists in the green low risk region, 5.5% exists in the low-moderate blue region, and 94.5% exists in the yellow moderate region.

TABLE 61

FOUR METRICS FACTORIAL VALUES, SAMPLE

Metric A Value	Metric B Value	Metric C Value	Metric D Value	Total
0.389	0.500	0.278	0.833	0.963
0.389	0.500	0.278	0.944	0.988
0.389	0.500	0.389	0.056	0.824
0.389	0.500	0.389	0.167	0.844
0.389	0.500	0.389	0.278	0.865

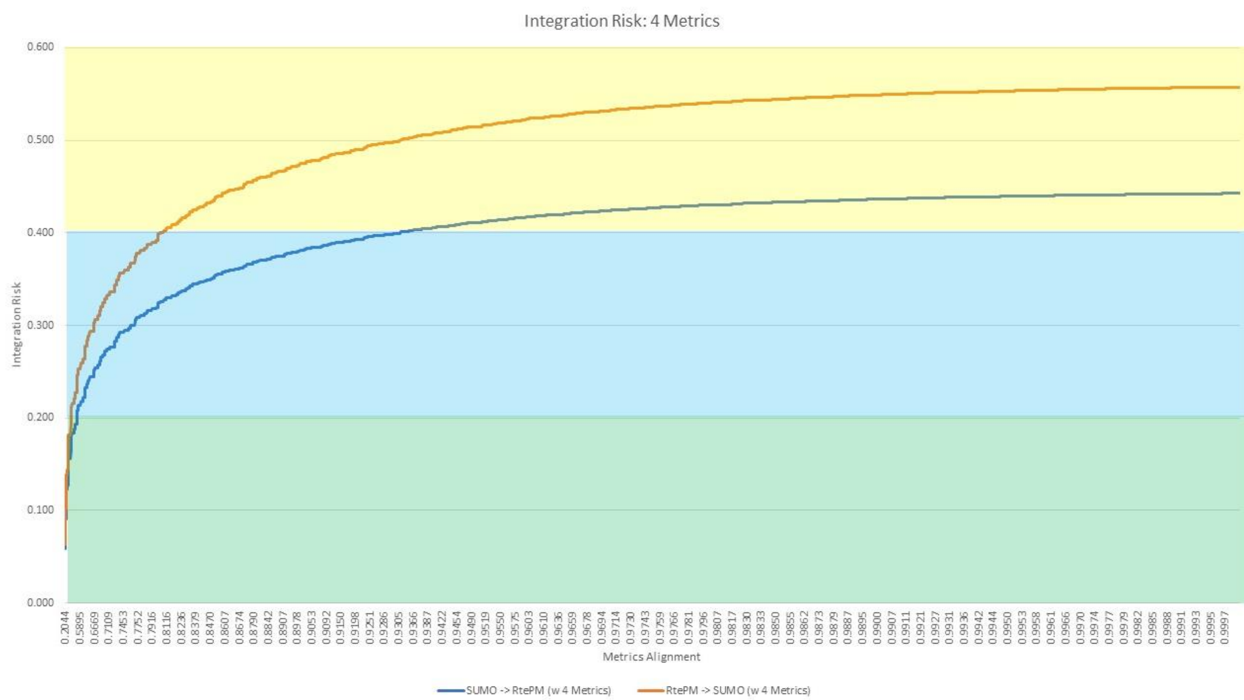


Fig. 37. Four metrics risk curves.

TABLE 62

FIVE METRICS FACTORIAL TABLE, SAMPLE

Metric A Value	Metric B Value	Metric C Value	Metric D Value	Metric E Value	Total
0.278	0.944	0.056	0.944	0.833	1.000
0.278	0.944	0.056	0.944	0.944	1.000
0.278	0.944	0.167	0.056	0.056	0.970
0.278	0.944	0.167	0.056	0.167	0.974
0.278	0.944	0.167	0.056	0.278	0.977
0.278	0.944	0.167	0.056	0.389	0.981

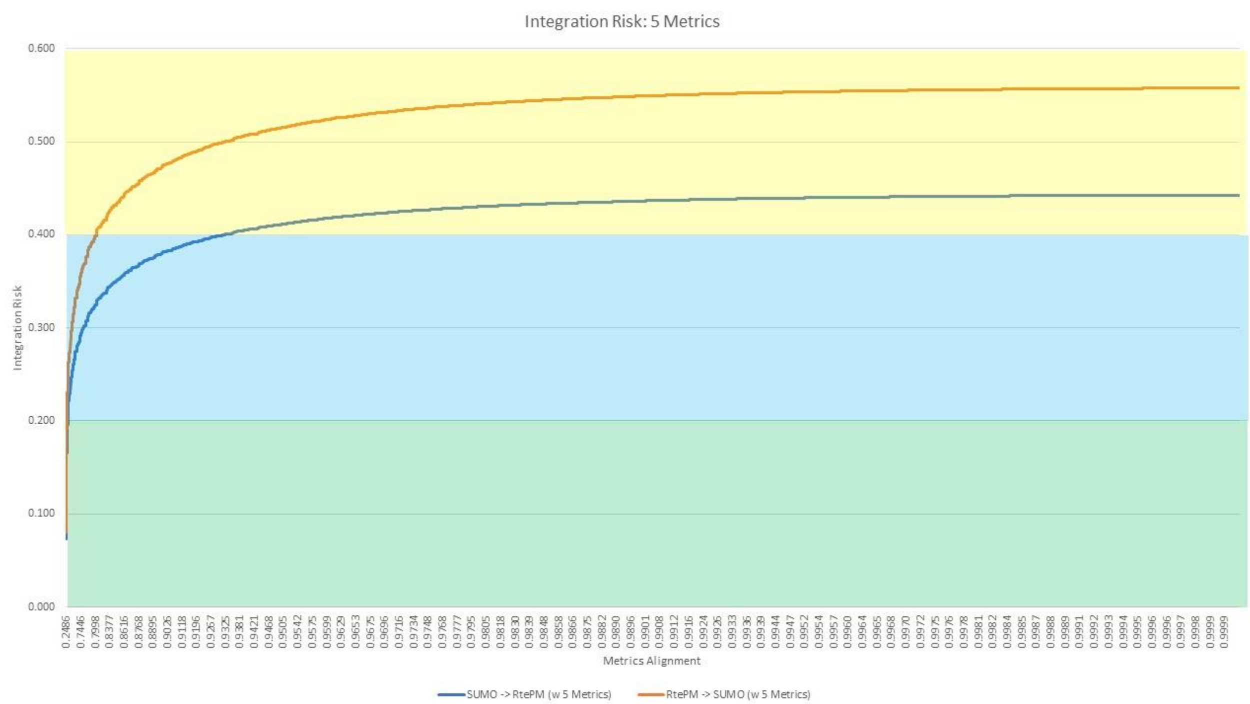


Fig. 38. Five metrics risk curves.

TABLE 63

SIX METRICS FACTORIAL TABLE

Metric A Value	Metric B Value	Metric C Value	Metric D Value	Metric E Value	Metric F Value	Total
0.056	0.056	0.389	0.944	0.833	0.389	0.997
0.611	0.833	0.833	0.611	0.056	0.278	0.997
0.167	0.944	0.833	0.389	0.833	0.833	1.000
0.500	0.500	0.278	0.278	0.611	0.167	0.958
0.944	0.278	0.278	0.611	0.167	0.833	0.998
0.500	0.500	0.611	0.056	0.833	0.278	0.989
0.944	0.056	0.056	0.389	0.944	0.278	0.999
0.722	0.833	0.611	0.833	0.278	0.833	1.000

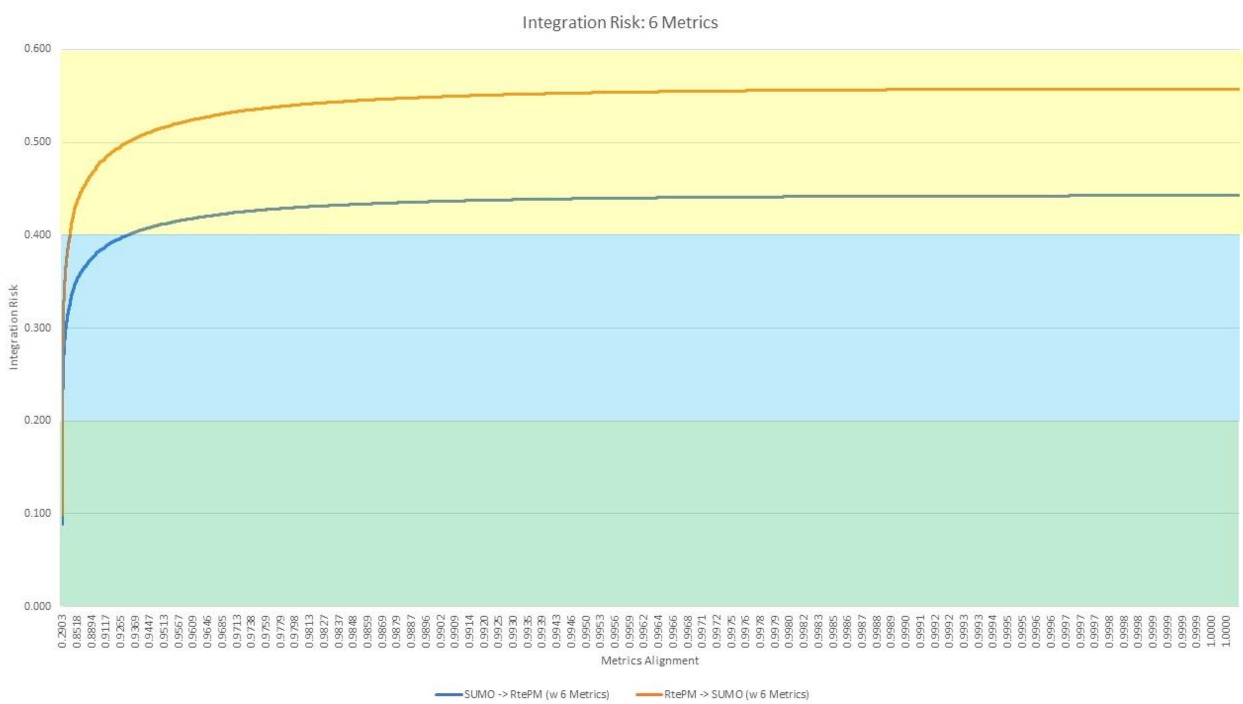


Fig. 39. Six metrics risk curves.

With seven output metrics, there are 9^7 permutations of metrics levels, or nearly 4.8 million. This set was impractical to calculate all permutations, so a random sample of 50,000 permutations was generated by selecting uniformly randomly a level of 1 to 9 for each metric and looking up that level's value. A brief sample set is shown below in Table 64.

Plotting all 50,000 random permutations with the model alignments generated the risk curve depicted below in Fig. 40. The same shading is used as before, and less than 0.001% of the permutations generate a risk in the low region, only 0.16% are in the blue low-moderate region, and the remaining 99.84% in the moderate yellow risk region.

Risk curves could further be generated with more potential output metrics, but a general trend is apparent at this point; when there are more output metrics influenced by the integration, the risk rapidly approaches a limit established by the conceptual misalignment.

TABLE 64

SEVEN METRICS FACTORIAL TABLE

Metric A Value	Metric B Value	Metric C Value	Metric D Value	Metric E Value	Metric F Value	Metric G Value	Total
0.056	0.056	0.389	0.944	0.833	0.389	0.997	0.056
0.611	0.833	0.833	0.611	0.056	0.278	0.997	0.611
0.167	0.944	0.833	0.389	0.833	0.833	1.000	0.167
0.500	0.500	0.278	0.278	0.611	0.167	0.958	0.500
0.944	0.278	0.278	0.611	0.167	0.833	0.998	0.944
0.500	0.500	0.611	0.056	0.833	0.278	0.989	0.500
0.944	0.056	0.056	0.389	0.944	0.278	0.999	0.944
0.722	0.833	0.611	0.833	0.278	0.833	1.000	0.722

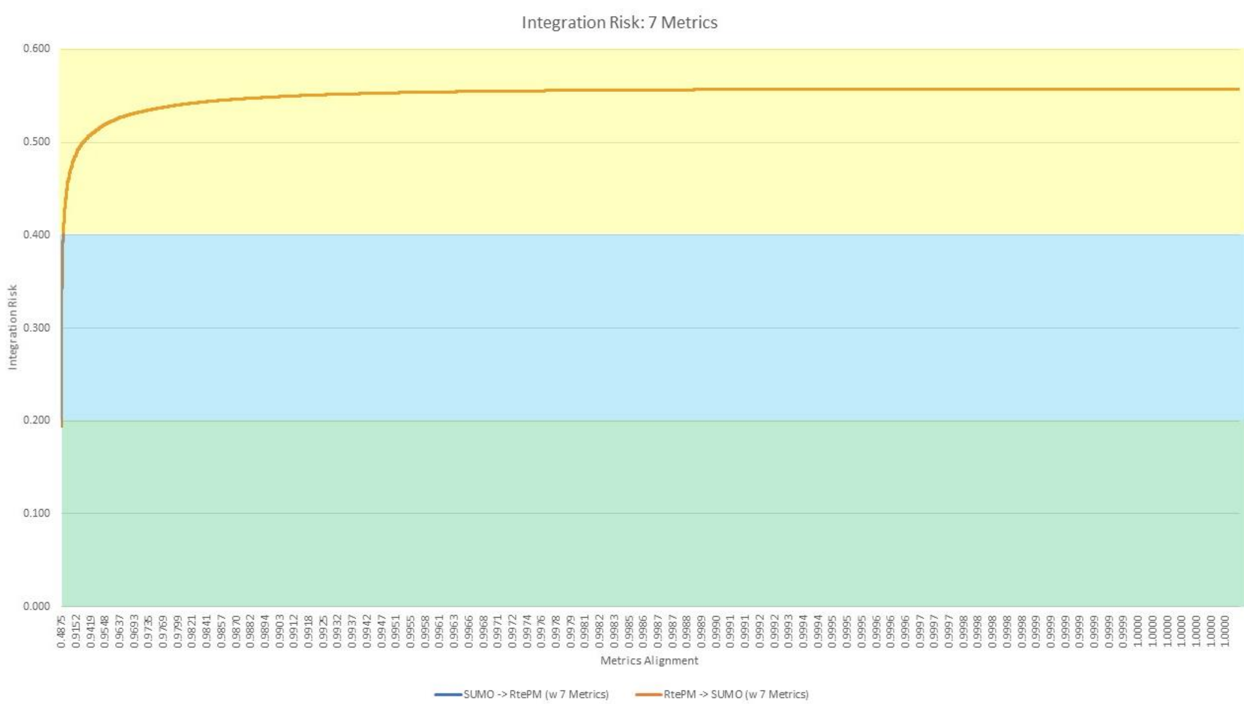


Fig. 40. Seven metrics risk curves.

CHAPTER 6

GENERALIZATION AND CONCLUSIONS

6.1 Integration of SUMO and RtePM

In the alignment example of RtePM and SUMO, the conceptual misalignment was moderately high. While these two models were chosen because of their similarity in a single domain, their representations of the underlying systems and behaviors have substantive differences. These differences are important to the models as independent analytic tools in their own right but create a sensitivity in the outputs of the integrated models. These two models could be integrated in either direction; that is SUMO could be integrated into RtePM or RtePM into SUMO. As examples, SUMO could be used to establish parameters for RtePM, or RtePM could establish a study for traffic control in SUMO. What is notable from this risk assessment is that these example scenarios will be established with concepts the other model cannot handle and will now be sensitive to information beyond its conceptual model.

As an example, if SUMO were to provide intersection-level data to RtePM for evacuation planning, a variety of concepts are not compatible. As specifically noted earlier, traffic lights are completely incompatible between the models. Where SUMO is calculating vehicular queues and wait times, and is sensitive to intersection actuators, RtePM looks at traffic lights as relative flows between cross flows of traffic. Those relative cross flows in SUMO are not the same as the calculated vehicular traffic in SUMO with explicit traffic signals and timings. The throughput traffic at an intersection—which may be critical in an evacuation—is now sensitive to these traffic light timings that are not manageable within RtePM. If SUMO were to provide multiple intersections of traffic throughput, the problem is multiplicative.

Similarly, RtePM could find that certain highway on-ramps are congested and look for ways to mitigate that by establishing a traffic study in SUMO. The assumptions that RtePM has about roads, lanes, and vehicles are not the same as SUMO's. The interaction of traffic at this hypothetical interchange is fundamentally different, and SUMO may not be able to replicate the issue.

Risk increases rapidly when more output metrics are included. This is perhaps unsurprising given that they are combinatoric in nature. However, in these two examples of integration, it's not unreasonable to expect that there are several there are several metrics.

In the example where SUMO creates intersection-level data for RtePM, the overall MOE of RtePM will still remain overall evacuation time and is likely not going to change very much due to one or even a few intersections in a region. However, RtePM uses congestion modeling as an MOP, which will undoubtably be sensitive to intersection level behaviors. In the example where RtePM directs a highway ramp mitigation question, the metrics might include through traffic volume, merging traffic volume, vehicular speeds, and delays. At a minimum, these examples would have two or three metrics that would potentially change as a result of the integration. Depending on the degree to which output metrics change, these are likely to present low integration risks.

6.2 Summary and Generalization

The objective of this research has been to develop a method to evaluate the risk presented to overall validity when models are integrated with one another. It has done so by extending research done in modeling and simulation theory into the risk analysis domain. Specifically, this research has used the OPR notation of models used these sets to compare models' conceptual alignment. The OPR notation allows for the categorization of model concepts. Next, this dissertation used value hierarchies to evaluate individual element's alignments from each model for use in the OPR notation. Lastly, the dissertation used models' alignments in an overarching risk assessment. Risk analysis in general holds that risk is a function of both probability and consequence. The conceptual alignment of models acts as a probability, indicating the portion of models that are common with one another. Consequences are evaluated with value hierarchies, leaving the interpretation of small or large impacts to decision makers and experts.

This theory was then applied to two models in the transportation domain as an example. The models chosen were SUMO and RtePM. SUMO is a free and open-source traffic engineering modeling tool and RtePM is an evacuation tool managed by Old Dominion University's Virginia Modeling Analysis and Simulation Center. SUMO is a microscopic simulation, offering high levels of detail to describe traffic systems whereas RtePM is a macroscopic simulation that calculates the time necessary to evacuate a wide geographic region in the event of an emergency. Both of these models are in the transportation domain and deal with the flow of vehicular traffic through a network of roadways. The assessments reveal elements in each model that does not align well or at all with the other model. The combination of these models produces sensitivities to the components that do not exist in both models, potentially breaking the underlying assumptions of the models' individual constructions and creating a risk to the composition's validity.

When the method developed in this dissertation to assess risk was applied, the conceptual misalignment between the models was found to be quite substantial, well above 99%. Combining this significant conceptual misalignment with even modest changes in model outputs would suggest to experts and decision makers to exercise caution in any potential integration of these two models.

6.3 Contributions Made by this Research

This dissertation has developed a method to assess risks to models' validity when they are integrated with one another. It has provided a mechanism to evaluate models' alignment and their overall appropriateness of use with one another. This research has developed a numerical method to trace models' underlying composition to an overall risk assessment to metamodel validity. There had been no readily available method to assess potential negative outcomes of model integration before this research. This dissertation is certainly related to and applicable to modeling and simulation research in model composability, interoperability, and reuse. These research topics within modeling and simulation assume certain degrees of similarity between models; this risk methodology can aide in understanding the appropriateness of that similarity. As has been shown in this transportation practicum, model alignment warrants a careful examination to assess potential risks.

6.4 Application

Where this research question had been inspired by the state of practice in analytic models, notably in the military domain, this work has application in other modeling and simulation applications and domains. Where this research had stressed that models taken as inputs to this risk assessment had been developed independently as might be found in analytic domains, this risk assessment can just as easily be applied in other applications of models and simulations. Model-based systems engineering, which is focused on the design and development of systems can benefit from understanding the risks inherent in underlying systems' representations and assumptions. Training and experimentation simulations also will use models in concert with one another to test particular components of systems and can benefit from understanding the risks inherent to understanding the entirety of their portfolio.

The example domain used in this research was the transportation domain. The risk assessment presented here has applicability in any domain where models are used in combination with one another. The research was inspired by the military analytic domain where models can and are used to inform one another—that is a form of parameter discovery or excursions of concepts. Military applications would benefit from the assessment of risk among their models for strategy and investment decisions. Logistics and supply chain models also would benefit from assessing the risk of underlying portfolios of models. Any complex decision space that relies on use of multiple modeling and simulation tools for decisions can benefit from understanding the risks inherent in their models' integration.

6.5 Future Research

Where this research presented a singular means to assess risk among models, there are likely alternative means to do so. This dissertation has laid the foundation for future research into model integration and multi-modeling risks. The calculations conducted revealed a moderately high misalignment of modeling concepts driving a high overall risk score. There are questions suitable for future research based on these surprising results, some of which are listed below, but certainly not exhaustive.

In this dissertation, the OPR notation was selected because of its novelty in distilling models, independent of any domain-based or specific ontology. Calculating models' conceptual alignment as a shared area of a Venn diagram assumes that all conceptual elements are weighted equally in their impact on the models' integration. Considerations perhaps could be made to parameter sensitivities that would make some misalignments more critical and others less critical. The integration of models creates new sensitivities that were not included in a single model before integration, but this analysis did not examine the magnitude of those sensitivities, rather, the changes in model outputs were considered as the impacts section in the risk analysis. Weighing misalignments by the relative sensitivities those elements have upon their individual models and on the overall integration is an open challenge.

As part of the risk analysis, the consequences of model integration are captured as changes in models' outputs. This inherently assumes that one of the models in the integration is a "principle" model and derives value from the "lesser" model. It also assumes that the principle model has been run in isolation without integration of the second model such that changes in the model's outputs may be collected. The value hierarchies of changes in model metrics are only one way to estimate the impact but have the benefit of incorporating domain experts' judgement on the impact of the metric changes. The value hierarchy presented in this dissertation is based on the principle maximum entropy in information where all uncertainty of the values is distributed across the spectrum of possible outcomes. In this case, a weighted centroid was calculated for 16 bins of values. Additional research may discover different weighting schemas for changes in model outputs. As in the case of model sensitivities to different elements in either or both underlying models, each metric that is output by the model is considered to be weighted equally with all other metrics. Designating metrics that are more or less germane to a specific use case, study, or analysis being performed by the models can lead to a weighing of model outputs.

Risk contains not only probability and consequence, but also a risk scenario. The risk scenarios in this dissertation were based on the combination of different types of model misalignments—scope, resolution, and structure. Other authors have noted different rationale for why models may be combined and integrated with one another such as concatenation, amplification, parameter discovery, model construction, and model merging [45]. These different integration purposes may serve as alternative risk scenarios and perhaps address some of the underlying assumptions about sensitivities and weightings of concepts and of metrics. Further analysis and research within the contexts of these different cases may reveal different forms of risk analysis in terms of models' ability to integrate with one another.

This dissertation did not fully instantiate simulations of these models for the evaluation. There is possibly more risk that is generated by specific instantiations of a simulation beyond that calculatable from models' conceptual alignment.

One of the risks not examined in this dissertation was the concept of confirmation bias. Models are occasionally used validate one another, and this is a generally accepted means to do so. However, what is not known is if a second model is generating insights and metrics that are similar to a first model, yet still wrong. In this case, decision makers may be led to believe there is more evidence to support an investment, course of action, or other decision when in fact they are being misled.

The concept of risk in model integration is a novel application of both risk analysis and model theory, and there are ample opportunities to evaluate alternative risk calculi and develop mitigation strategies to ensure consistency in model integration.

REFERENCES

[1] C. D. Turnitsa, "Exploring the components of dynamic modeling techniques," Ph.D. dissertation, Modeling, Simulation and Visualization Engineering Department Old Dominion University, Suffolk, VA, 2012.

[2] J. Lathrop and B. Ezell, "A systems approach to risk analysis validation for risk management," *Safety Science*, pp. 187-195, 2017.

[3] O. Balci, "Validation, verification, and testing techniques throughout the life cycle of a simulation study," *Annals of Operations Research*, vol. 53, pp. 121-173, 1994.

[4] J.-K. Lee, Y.-H. Lim, and S.-D. Chi, "Hierarchical modeling and simulation environment for intelligent transportation systems," *Simulation*, vol. 80, no. 2, pp. 61-76, 2004.

[5] P. K. Davis, "An introduction to variable-resolution modeling," in *Warfare Modeling*, J. Bracken, M. Kress, and R. E. Rosnthal Eds. Danvers, MA: John Wiley & Sons, 1995, pp. 5-36.

[6] A. Tolk, "Interoperability and composability," in *Introduction to Modeling and Simulation*, J. A. Sokolowski and C. M. Banks Eds. Hoboken, NJ: John Wiley & Sons, 2010, pp. 403-433.

[7] E. W. Weisel, M. D. Petty, and R. R. Mielke, "Validity of models and classes of models in semantic composability," in *Fall Simulation Interoperability Workshop 2003*, Orlando, FL, Sept. 14-19, 2003, vol. 2, pp. 569-579.

[8] M. A. Gallagher, D. J. Caswell, B. Hanlon, and J. M. Hill, "Rethinking the hierarchy of analytic models and simulations for conflicts," *Military Operations Research*, vol. 19, no. 4, pp. 15-24, 2014.

[9] D. L. Neyland, *Virtual Combat*. Mechanicsburg, PA: Stackpole Books, 1997.

[10] F. Kuhl, R. Weatherly, and J. Dahmann, *Creating Computer Simulation Systems: An Introduction to the High Level Architecture*. Upper Saddle River, NJ: Prentice Hall PTR, 1999.

[11] "Free HLA Tutorial and Software." Pitch Technologies. http://www.pitchtechnologies.com/hlatutorial/ (accessed Jul. 31, 2017).

[12] S. Y. Diallo, "Towards a formal theory of interoperability," Ph.D. dissertation, Modeling, Simulation and Visualization Engineering Department Old Dominion University, Suffolk, VA, 2010.

[13] W. L. Oberkampf, S. M. DeLand, B. M. Rutherford, K. V. Diegert, and K. F. Alvin, "Error and uncertainty in modeling and simulation," *Reliability Engineering and System Safety,* vol. 75, no. 3, pp. 333-357, 2002.

[14] P. T. Hester, "Why optimisation of a system of systems is both unattainable and unnecessary," *International Journal of System of Systems Engineering,* vol. 3, no. 3, pp. 268-276, 2012.

[15] S. Wartik, B. Haugh, F. Loaiza, and M. R. Hieb, "Alignment of Army integrated core data model and object management standards category," Institute for Defense Analysis, Alexandria, VA, 2001.

[16] S. Kent, "Words of estimative probability," Central Intelligence Agency, 1964.

[17] A. Tolk, S. Y. Diallo, and C. D. Turnitsa, "Applying the levels of conceptual interoperability model in support of integratability, interoperability, and composability for system-of-systems engineering," *Systemics, Cybernetics and Informatics,* vol. 5, no. 5, pp. 65-74, 2007.

[18] M. J. North, "A time and space complexity analysis of model integration," in *2014 Winter Simulation Conference*, Savannah, GA, Dec. 7-10, 2014: IEEE Press.

[19] R. G. Sargent, "Verification and validation of simulation models," *Journal of Simulation*, vol. 7, no. 1, pp. 12-24, 2013.

[20] L. Bair and A. Tolk, "Towards a unified theory of validation," in *2013 Winter Simulation Conference*, Washington, D.C., R. Pasupathy, S.-H. Kim, A. Tolk, R. Hill, and M. E. Kuhl, Eds., Dec. 8-11, 2013: IEEE, pp. 1,245-1,256.

[21] M. D. Petty, "Verification, validation, and accreditation," in *Modeling and Simulation Fundamentals: Theoretical Underpinnings and Practical Domains*, J. A. Sokolowski and C. M. Banks Eds. Hoboken, NJ: John Wiley & Sons, 2010, pp. 325-372.

[22] A. M. Law, *Simulation Modeling & Analysis*. Boston, MA: McGraw Hill, 2007.

[23] E. Yahia, A. Aubry, and H. Panetto, "Formal measures for semantic interoperability assessment in cooperative enterprise information systems," *Computers in Industry*, vol. 63, no. 5, pp. 443-457, 2012.

[24] I. I. Mitroff and T. R. Featheringham, "On systemic problem solving and the error of the third kind," *Behavioral Science: Journal of the Society for General Systems Research*, vol. 19, no. 6, pp. 383-393, 1974.

[25] S. Y. Diallo, J. J. Gore, H. Herencia-Zapana, and A. Tolk, "Toward a formalism of modeling and simulation using model theory," *Complexity*, vol. 19, no. 3, pp. 56-63, 2014.

[26] B. L. Heath and R. R. Hill, "Final report developing an agent-based modeling verification and validation approach for improving Air Force analytical support," Wright State University, Dayton, OH, 2010.

[27] M. A. Hofman, "Challenges of model interoperation in military simulations," *Simulation,* vol. 80, no. 12, pp. 659-667, 2004.

[28] K. Popper, *The Logic of Scientific Discovery*. London: Routledge, 1959.

[29] G. E. P. Box and N. E. Draper, *Empirical Model-building and Response Surfaces*. John Wiley & Sons, Inc., 1987.

[30] B. P. Zeigler, H. Praehofer, and T. G. Kim, *Theory of Modeling and Simulation*. Amsterdam: Academic Press, 2000.

[31] M. C. Jones, "Composability," in *Modeling and Simulation Support for System of Systems Engineering Applications*, L. B. Rainey and A. Tolk Eds. Hoboken, NJ: John Wiley & Sons, 2015, pp. 45-74.

[32] A. Tolk and C. D. Turnitsa, "Conceptual modeling with processes," in *WSC '12: Winter Simulation Conference*, Berlin, Germany, Dec. 9-12, 2012: IEEE, pp. 2,641 - 2,653.

[33] R. D. King and C. D. Turnitsa, "The landscape of assumptions," in *Proceeding of the 2008 Spring Simulation Multiconference (SpringSim'08)*, Ottawa, Canada, Apr. 14-17, 2008: Society for Computer Simulation International, pp. 81-88.

[34] R. D. King, "On the role of assertions for conceptual modeling as enablers of composable simulation solutions," Ph.D. dissertation, Modeling, Simulation and Visualization Engineering Department, Old Dominion University, Suffolk, VA, 2009.

[35] B. Ezell and K. Crowther, "Philosophical issues and their implications for the systems architect," *Foundations of Science*, vol. 12, pp. 269-276, 2007.

[36] J. Denil, S. Klikovits, P. J. Mosterman, A. Vallecillo, and H. Vangheluwe, "The experiment model and validity frame in M&S," in *Symposium on Theory of Modeling & Simulation (SpringSim'17)*, Virginia Beach, VA, Apr. 23-26, 2017, vol. 49, no. 4: Society for Computer Simulation International pp. 1-12.

[37] M. K. Traoré and A. Muzy, "Capturing the dual relationship between simulation models and their context," *Simulation Modelling Practice and Theory*, vol. 14, no. 2, pp. 126-142, 2006.

[38] S. Kaplan and B. J. Garrick, "On the quantitative definition of risk," *Risk Analysis*, vol. 1, no. 1, pp. 11-27, 1981.

[39] D. W. Hubbard, *The Failure of Risk Management: Why its Broken and How to Fix It*. John Wiley & Sons, 2009.

[40] Johns Hopkins University Applied Physics Laboratory Team, "M&S Use Risk Methodology (MURM)," Office of the Assistant Secretary of Defense, Laurel, MD, Technical Report Apr. 2011.

[41] P. D. Leedy and J. E. Ormrod, *Practical Research: Planning and Design*. Boston: Pearson, 2013.

[42] A. Tolk, "Terms and application domains," in *Engineering Principles of Combat Modeling and Distributed Simulation*. Hoboken, NJ: John Wiley & Sons, 2012, pp. 55-78.

[43] B. Downs, "The maritime security risk analysis model: Applying the latest risk assessment techniques to maritime security," in *Proceedings of the Marine Safety & Security Council*, 2007, vol. 61, no. 1: U.S. Coast Guard Headquarters, pp. 36-39.

[44] D. R. Dolk and J. E. Kotterman, "Model integration and a theory of models," *Decision Support Systems*, vol. 9, no. 1, pp. 51-63, 1993.

[45] A. H. Levis and A. A. Jbara, "Multi-modeling, meta-modeling, and workflow languages," in *Theory and Application of Multi-Formalism Modeling*, M. Gribaudo and M. Iacono, Eds. Hershey, PA: IGI Global, 2014, pp. 56-80.

[46] "RtePM: Real Time Evacuation Planning Model." Virginia Modeling, Analysis and Simulation Center (VMASC). http://rtepm.vmasc.odu.edu/ (accessed 2019).

[47] VMASC, "Real time evacuation planning model user's guide," Old Dominion University, Suffolk, VA, User Manual Dec. 2013.

[48] DDL Omni Engineering, "Independent verification and validation (IV&V) of Real time evacuation Planning Model (RtePM)," Old Dominion University, Virginia Beach, VA, Apr. 22, 2013.

[49] R. Weibel, "Validation assessment of the Real-time evacuation Planning Model (RtePM) traffic simulation," MIT, Lexington, MA, Unpublished 2017.

[50] "SUMO Documentation." German Aerospace Center (DLR) and others. https://sumo.dlr.de/docs/SUMO_User_Documentation.html (accessed 2019).

[51] D. Krajzewicz, J. Erdmann, M. Behrisch, and L. Bieker, "Recent development and applications of SUMO - Simulation of Urban MObility," *International Journal on Advances in Systems and Measurements*, vol. 5, no. 3 & 4, pp. 128-138, 2012.

[52] J. Erdmann, "Lane-changing model in SUMO," in *SUMO2014 Modeling Mobility with Open Data*, Berlin-Adlershof, Germany, May 15-16, 2014.

APPENDICES

APPENDIX A. ENNUMERATION OF RISK SCENARIOS

Case	Alignment Description				Object Similarity				Description	Consequence
	Objects		Processes		Model A -> Model B	Model B -> Model A	Model A -> Model B	Model B -> Model A		
1	Model A's Objects are well-aligned with Model B's Objects	Model B's Objects are well-aligned with Model A's Objects	Model A's Processes are well-aligned with Model B's Processes	Model B's Processes are well-aligned with Model A's Processes	100%	100%	100%	100%	These models are well-aligned	Trivial
2	Model A's Objects are well-aligned with Model B's Objects	Model B's Objects are well-aligned with Model A's Objects	Model A's Processes are well-aligned with Model B's Processes	Model B's Processes are partially aligned with Model A's Processes	100%	100%	100%	Partial	These models fairly well-aligned	The models behave differently in at least some aspects and could have differing outputs

Case	Alignment Description				Object Similarity				Description	Consequence
	Objects		Processes		Model A -> Model B	Model B -> Model A	Model A -> Model B	Model B -> Model A		
3	Model A's Objects are well-aligned with Model B's Objects	Model B's Objects are well-aligned with Model A's Objects	Model A's Processes are partially aligned with Model B's Processes	Model B's Processes are well-aligned with Model A's Processes	100%	100%	Partial	100%	These models fairly well-aligned	The models behave differently in at least some aspects and could have differing outputs
4	Model A's Objects are well-aligned with Model B's Objects	Model B's Objects are well-aligned with Model A's Objects	Model A's Processes are partially aligned with Model B's Processes	Model B's Processes are partially aligned with Model A's Processes	100%	100%	Partial	Partial	These models moderately well-aligned	The models behave differently in at least some aspects and could have differing outputs

Case	Alignment Description				Object Similarity				Description	Consequence
	Objects		Processes		Model A -> Model B	Model B -> Model A	Model A -> Model B	Model B -> Model A		
5	Model A's Objects are well-aligned with Model B's Objects	Model B's Objects are partially aligned with Model A's Objects	Model A's Processes are well-aligned with Model B's Processes	Model B's Processes are well-aligned with Model A's Processes	100%	Partial	100%	100%	These models fairly well-aligned	These models contain different information regarding the entities being modeled
6	Model A's Objects are well-aligned with Model B's Objects	Model B's Objects are partially aligned with Model A's Objects	Model A's Processes are well-aligned with Model B's Processes	Model B's Processes are partially aligned with Model A's Processes	100%	Partial	100%	Partial	These models are moderately aligned	These models contain different information regarding the entities being modeled and their behaviors

Case	Alignment Description				Object Similarity				Description	Consequence
	Objects		Processes		Model A -> Model B	Model B -> Model A	Model A -> Model B	Model B -> Model A		
7	Model A's Objects are well-aligned with Model B's Objects	Model B's Objects are partially aligned with Model A's Objects	Model A's Processes are partially aligned with Model B's Processes	Model B's Processes are well-aligned with Model A's Processes	100%	Partial	Partial	100%	These models are moderately aligned	These models contain different information regarding the entities being modeled and their behaviors
8	Model A's Objects are well-aligned with Model B's Objects	Model B's Objects are partially aligned with Model A's Objects	Model A's Processes are partially aligned with Model B's Processes	Model B's Processes are partially aligned with Model A's Processes	100%	Partial	Partial	Partial	These models are moderately aligned	These models contain different information regarding the entities being modeled and their behaviors

Case	Alignment Description				Object Similarity				Description	Consequence
	Objects		Processes		Model A -> Model B	Model B -> Model A	Model A -> Model B	Model B -> Model A		
9	Model A's Objects are partially aligned with Model B's Objects	Model B's Objects are well-aligned with Model A's Objects	Model A's Processes are well-aligned with Model B's Processes	Model B's Processes are well-aligned with Model A's Processes	Partial	100%	100%	100%	These models moderately well-aligned	These models contain different information regarding the entities being modeled but behave similarly
10	Model A's Objects are partially aligned with Model B's Objects	Model B's Objects are well-aligned with Model A's Objects	Model A's Processes are well-aligned with Model B's Processes	Model B's Processes are partially aligned with Model A's Processes	Partial	100%	100%	Partial	These models partially align	These models differ in their entities and processes

Case	Alignment Description				Object Similarity				Description	Consequence
	Objects		Processes		Model A -> Model B	Model B -> Model A	Model A -> Model B	Model B -> Model A		
11	Model A's Objects are partially aligned with Model B's Objects	Model B's Objects are well-aligned with Model A's Objects	Model A's Processes are partially aligned with Model B's Processes	Model B's Processes are well-aligned with Model A's Processes	Partial	100%	Partial	100%	These models partially align	These models differ in their entities and processes
12	Model A's Objects are partially aligned with Model B's Objects	Model B's Objects are well-aligned with Model A's Objects	Model A's Processes are partially aligned with Model B's Processes	Model B's Processes are partially aligned with Model A's Processes	Partial	100%	Partial	Partial	These models differ noticeably	These models differ in their entities and processes

Case	Alignment Description				Object Similarity				Description	Consequence
	Objects		Processes		Model A -> Model B	Model B -> Model A	Model A -> Model B	Model B -> Model A		
13	Model A's Objects are partially aligned with Model B's Objects	Model B's Objects are partially aligned with Model A's Objects	Model A's Processes are well-aligned with Model B's Processes	Model B's Processes are well-aligned with Model A's Processes	Partial	Partial	100%	100%	These models differ noticeably	These models differ in their entities, but dynamic behaviors are similar
14	Model A's Objects are partially aligned with Model B's Objects	Model B's Objects are partially aligned with Model A's Objects	Model A's Processes are well-aligned with Model B's Processes	Model B's Processes are partially aligned with Model A's Processes	Partial	Partial	100%	Partial	These models differ noticeably	These models differ in their entities, and dynamic behaviors are divergent

Case	Alignment Description				Object Similarity				Description	Consequence
	Objects		Processes		Model A -> Model B	Model B -> Model A	Model A -> Model B	Model B -> Model A		
15	Model A's Objects are partially aligned with Model B's Objects	Model B's Objects are partially aligned with Model A's Objects	Model A's Processes are partially aligned with Model B's Processes	Model B's Processes are well-aligned with Model A's Processes	Partial	Partial	Partial	100%	These models differ noticeably	These models differ in their entities, and dynamic behaviors are divergent
16	Model A's Objects are partially aligned with Model B's Objects	Model B's Objects are partially aligned with Model A's Objects	Model A's Processes are partially aligned with Model B's Processes	Model B's Processes are partially aligned with Model A's Processes	Partial	Partial	Partial	Partial	These models are not well-aligned	There is different information AND different behaviors in these models that could drive to divergent outputs

APPENDIX B. SAMPLES FOR THE CANNONICAL EXAMPLE

	Series Alpha	Series Beta	Series Gamma	Series Delta
Arithmetic Mean	21.55	19.65	23.325	18.95
Geometric Average	16.24380939	14.63702271	19.68149398	13.76830719
Harmonic Mean	9.509536122	8.230467342	15.46725774	8.214734621
X1	4	24	22	20
X2	25	30	19	45
X3	28	33	12	4
X4	8	10	46	22
X5	13	28	38	12
X6	8	7	16	5
X7	9	1	29	9
X8	3	23	31	41
X9	35	20	24	18
X10	25	3	14	2
X11	19	4	25	12
X12	36	33	41	9
X13	13	9	15	25
X14	27	36	7	29
X15	44	13	21	2
X16	3	26	13	7
X17	26	25	5	12
X18	14	9	47	28
X19	1	3	17	10
X20	22	19	46	25
X21	37	17	41	14
X22	4	39	29	20
X23	19	29	7	13
X24	26	2	35	21
X25	27	17	34	45
X26	29	28	3	5
X27	24	21	27	46
X28	42	38	23	9
X29	9	12	10	34

	Series Alpha	Series Beta	Series Gamma	Series Delta
Arithmetic Mean	21.55	19.65	23.325	18.95
Geometric Average	16.24380939	14.63702271	19.68149398	13.76830719
Harmonic Mean	9.509536122	8.230467342	15.46725774	8.214734621
X30	33	23	21	1
X31	47	2	10	24
X32	12	45	15	47
X33	38	22	21	35
X34	25	23	29	6
X35	27	12	11	6
X36	30	25	19	25
X37	23	22	35	27
X38	35	5	35	19
X39	8	25	31	10
X40	4	23	9	14

VITA

John Brendon Young is a senior Operations Research Analyst with the U.S. Air Force Headquarters for Studies, Analysis, and Assessments. He received his B.S. in Industrial and Systems Engineering from Virginia Tech in 2002, and a M. Eng. in Modeling and Simulation from Old Dominion University in 2007.

John has had a diverse career developing and applying Models, Simulations, and Analysis (MS&A) largely in the national security sector. John contributed to a variety of analytic disciplines while a government civilian at the Joint Warfare Analysis Center (JWAC) from 2003 to 2012. John worked on strategic analysis at the Joint Chiefs of Staff J8 (Force Structure, Resources, and Assessments) as a Senior Operations Research Analyst from 2012 to 2015. John was a lead modeling and simulation engineer at the MITRE Corporation from 2015 to 2019, applying modeling and simulation techniques beyond analytic domains and into experimentation and engineering. John returned to civil service in 2019 at the U.S. Air Force as a study leader and senior campaign analyst.

John's research interests are in combat modeling and simulation and improving the quality of decisions at strategic levels.

John lives in Fredericksburg, Virginia with his wife Mary of 17 years and their two sons Matthew (11) and Theodore Young (4).