

Report from the Fidelity Implementation Study Group

Redacted by: David C. Gross
M/C JR-80
499 Boeing Blvd.
Huntsville, AL 35824
David.C.Gross@Boeing.Com

Keywords:

Fidelity, Fidelity Implementation Study Group, Simulation Quality, Simulation Characterization

ABSTRACT: The Fidelity Implementation Study Group (Fidelity ISG) has the objective of leveraging the current interest and excitement in describing, quantifying, and using simulation fidelity, particularly in the context of the HLA. Recognition of the importance of capability to characterize simulation fidelity has been growing in the SISO community for some time. At least five SIW forums (Analysis, T&E, RD&E, Logistics, VV&A), recommended the establishment of the Fidelity ISG to address a number of specific simulation fidelity issues which cut across the concerns of virtually all SISO forums. The ability to describe and quantify simulation fidelity appropriately will be essential for effective interoperability and support of endeavors such as Simulation Based Acquisition.

The Fidelity was chartered to complete its tasking by the March 1999 SIW, and produce an integrated project report addressing:

- 1) Appropriate simulation fidelity definitions for the SISO community and proposed for inclusion in the DMSO M&S Glossary and similar lexicons.
- 2) Basic simulation fidelity concepts for the SISO and other simulation communities.
- 3) A summary of workable contextual frameworks within which simulation fidelity is considered, with an explicit indication of how such frameworks would relate to the larger theoretical context of modeling and simulation theory.
- 4) Initial identification of methods (or the approach to methods) and metrics for usefully defining, estimating, and measuring aspects of simulation fidelity.
- 5) Areas for further study and discussion.

This paper is the required report from the Fidelity ISG. While the named author has collated submissions and edited the final version, in truth the report is the result of more than a dozen direct contributions in the form of draft sections and the indirect contributions of the more than one hundred subscribers to the Fidelity ISG reflector.

1. Introduction

1.1 Why Fidelity?

Fidelity is important because it is at the heart of what distinguishes computer simulations from any other computer program. A computer simulation is the same as any computer program, except that the purpose of a computer simulation is to represent some behavior of some things in the real (or imagined) world. Without this crucial distinction, a computer simulation, and correspondingly the challenges associated in developing one, are just those of any computer program, and therefore not worthy of study independent of computer science.

However, the unique purpose of simulation, in representing “real” systems, does distinguish computer simulation from other computer programs. Consider: a specification for a simulation will read much as a specification for any other computer program, requirements for controls and displays, functional performance, safety concerns, and so forth. However, such specifications always address a topic unique for simulations, the requirements about the simulation “objects of interest”. These requirements sometimes appear as an extensive discussion, and sometimes a brief

reference to other design criteria. Therefore, the unique measure of goodness for simulation is how well the simulation makes its representation, or its *fidelity*.

Despite its apparent essential relationship to developing and using simulations, fidelity has to be the least consistently used, yet most commonly used term in the simulation community. However, we can identify some small initial consensus about fidelity. We know that fidelity is good and that more of it is better. We at least suspect that fidelity is expensive both to buy and to own. Merriam Webster contributes the following about fidelity:

- **Synonyms:** allegiance, ardor, devotion, faithfulness, fealty, loyalty, piety
- **Related Words:** constancy, staunchness, steadfastness; dependability, reliability, trustworthiness
- **Contrasted Words:** disloyalty, falseness, falsity, perfidiousness, traitorousness, treacherousness, treachery; undependableness, unreliability, untrustworthiness
- **Antonyms:** perfidy; faithlessness

These words certainly support our thought that fidelity is good! More specifically focused on simulation fidelity, the Defense Modeling and Simulation Office Glossary defines fidelity as “The accuracy of the representation when compared to the real world”. The simulation community understands and uses fidelity at least in some general sense. Users inextricably associate fidelity with the simulation's suitability for its purpose, such as analysis, design, training, etc. Fidelity is understood to be one of the largest cost drivers for simulation. If we consider that fidelity uniquely defines simulation uniquely from other computer programs, then fidelity can be seen as the key to simulation *validation* (the determination that the right simulation has been built for a specific purpose).

However, simulation fidelity has proved difficult to use in practical application. For example, very few applications attempt to describe fidelity objectively, much less quantitatively. The typical practice is to default to linguistic terms such as high, medium, and low; however this is unsatisfactory because it is very subjective. Since fidelity is regarded as a primary measure of goodness for simulations, developing an *objective* fidelity measure offers substantial benefit for describing and scoping simulations.

1.2 Background

The importance of fidelity in developing and using simulations is reflected by the ongoing interest in fidelity at the Simulation Interoperability Standards Organization's (SISO's) Simulation Interoperability Workshop (SIW). In response to this interest, SISO established the Fidelity Implementation Study Group (ISG). The Fidelity ISG is attempting to leverage this ongoing interest to make practical progress in describing, quantifying, and using fidelity.

The Fidelity ISG is chartered to develop:

- 1) A lexicon for simulation fidelity terms and concepts.
- 2) A contextual framework related those terms and concepts and simulation theory.
- 3) A set of methods and metric by which fidelity is defined, estimated, and measured.

The significance of fidelity is underscored by the demands of current simulation initiatives such as the High Level Architecture (HLA) and Simulation Based Acquisition (SBA). These initiatives require the simulation community to discover and use new, more powerful ways of describing and quantifying simulations.

The Fidelity ISG realized very early that constructing and using common definitions was a key to making progress. Therefore, much of the ISG's effort has gone into the construction of a Glossary of Fidelity Related Terms, which appears as Appendix A in this document. This glossary builds upon many previous efforts, which are extensively cited. Sources included the DIS Glossary, the current DMSO Glossary, and particularly the work of the Fidelity sub-group of the SIW RDE User Community Forum. This sub-group, led by Ralph Weber (Computer Sciences Corporation), compiled a large number of definitions related to fidelity from various sources, and then sponsored discussion of those terms via the RDE reflector, at two interim meetings of the RDE forum, and at two SIWs. *The definitions in the glossary are appropriate for wide community acceptance.* They will be used in this document. These definitions are not perfect, but until better ones are commonly accepted, the most important thing is that we all use the same terms so that we can communicate with one another most effectively. If we need to invent additional terms for concepts not adequately covered by these terms, then we should do that instead of causing confusion by using the same terms with different meanings.

1.3 Scope

This document is the first report out of the Fidelity ISG. It makes an attempt to gather up in one place recent fidelity author's thinking on fidelity to both report on the current state of research and to form a basis to assess the suitability of fidelity as a subject for standardization. It does not claim to be a draft standard, nor a plan for standardization. It does provide a basis for an assessment of the maturity of fidelity technology and its readiness for standardization.

1.4 Reader's Guide

This document is structured in three major sections. First, it provides an overview of the state of knowledge about fidelity. The next section summarizes four major conceptual frameworks about fidelity, as published through the Simulation Interoperability Workshops. The final section explores the possibility of developing fidelity standards.

This report is the product of many contributors, including every member of the Fidelity ISG. Special acknowledgement is made of the following individuals for the contributions of sections: Furman Haddix, Dean Hartley, Scott Harmon, Geoff Hone, Bruce MacDonald, Dale Pace, Manfred Rosa, and Bill Tucker.

2. Fidelity Overview

2.1 What is Fidelity?

The basic connotation of simulation fidelity is clear, even when there are differences of opinion about the exact definition of fidelity. Simulation fidelity has to do with how well the simulation responses and results correspond to what the simulation represents. What the simulation represents is sometimes called the “real world” and sometimes called something else. How well the simulation represents this is addressed in a variety of ways and is often described by terms such as “the degree to which,” similarity, accuracy, precision, etc. The following is the formal definition, which has emerged out of the Fidelity ISG, and appears in Appendix A.

Fidelity: 1. *The degree to which a model or simulation reproduces the state and behavior of a real world object or the perception of a real world object, feature, condition, or chosen standard in a measurable or perceivable manner; a measure of the realism of a model or simulation; faithfulness. Fidelity should generally be described with respect to the measures, standards or perceptions used in assessing or stating it. See accuracy, sensitivity, precision, resolution, repeatability, model/simulation validation.* 2. *The methods, metrics, and descriptions of models or simulations used to compare those models or simulations to their real world referents or to other simulations in such terms as accuracy, scope, resolution, level of detail, level of abstraction and repeatability. Fidelity can characterize the representations of a model, a simulation, the data used by a simulation (e.g., input, characteristic or parametric), or an exercise. Each of these fidelity types has different implications for the applications that employ these representations.*

2.1.1 Fidelity Descriptions

A large number of ways of describing simulation fidelity exist in the literature, many of which were identified in a synoptic review of the simulation fidelity literature [18]. These are grouped in three basic categories: *short*, *shorthand*, and *long*. These terms were selected for classifying fidelity descriptions in a way that avoided pejorative labels.

- **Short descriptions** of simulation fidelity, such as qualitative labels such as “high,” “medium,” or “low” fidelity and dimensionless characterizations tend to have more public relations utility than

technical utility. They serve mainly as advertising blurbs about simulations and frequently lack the information content, which is necessary to support technical decisions about simulation appropriateness for a particular application.

- **Shorthand descriptions** of simulation fidelity, such as classification of a flight simulation as Level D by Federal Aviation Administration (FAA) Advisory Circular (AC) 120-40, is normally an indication that a simulation satisfies a multiple attribute criterion. And incidentally, the number of attributes satisfied may well exceed 100, which is the case for classification as a Level D flight simulator by the FAA.
- **Long descriptions** of simulation fidelity typically describe simulation fidelity in terms of multiple attributes. The number and kinds of attributes considered varies with the construct being employed for simulation fidelity. Most constructs consider either the scope of the simulation’s treatment of significant factors in the application domain (this usually involves some kind of *enumeration*), the quality of treatment of factors within the simulation (as indicated by parameter accuracy, resolution, etc.), or both [19].

2.1.2 Qualitative vs. Quantitative Fidelity

The *qualitative* nature of fidelity is commonly understood. The *quantitative* nature of fidelity implied by the above definition is often overlooked or neglected. There is a tendency to consider fidelity as somewhat ethereal and thereby un-quantifiable, however it is often possible to decompose all or part of a qualitative assessment into a collection of quantitative assessments. For example, qualitative characteristics can certainly be perceived – such as a good musical performance, a good meal, a bad experience, etc. But each of these qualitative assessments has quantitative corollaries. The good musical performance was one in which the performer closely followed the timing, frequency (pitch), etc. specified by the composer. The good meal was one in which the amount of ingredients was as specified in the recipes, and prepared accordingly.

Qualitative descriptions of simulation fidelity have utility, but that utility is more in the public relations arena than in the technical arena since most qualitative descriptions of fidelity will be *short* (not shorthand) descriptions of fidelity. It may be difficult to develop objective evaluation processes by which one determines a qualitative description of fidelity (such as *high*, *medium*, or *low*). However, subjective evaluation of

simulation fidelity qualitatively by knowledgeable persons may be useful in some cases.

Quantitative descriptions of simulation fidelity are required when specific, objective characteristics of a simulation must be evaluated. If a simulation must produce results such that a critical parameter (e.g., miss distance for a missile flyout simulation) requires specified levels of accuracy and precision in order to support design decisions or concepts of operation assessments, then only quantitative descriptions of simulation fidelity that address such can satisfy determination of simulation appropriateness.

2.1.3 Measurement Issues

There are two obstacles to any fidelity measurement standard. The first is that there must exist a definition of the real or imagined world sufficient to measure the difference between it and the simulation. The second is that the simulation must be similarly defined.

The first obstacle explains why essentially all fidelity papers call for the establishment of some common referent -- the "world" is not a good ruler to measure fidelity. The world is too large and complex and too poorly understood to be a practical measure. Therefore, we must establish a commonly understood standard. Many authors go farther to claim that we should only attempt to assess the fidelity of a simulation against those aspects of the referent that we had intended to simulate, arguing that if the simulation represented all aspects of the simuland it would *be* the simuland. The good news is that this means we need only measure how well a simulation represents a behavior against the behavior it was intended to represent. This only seems fair! The bad news is that in order to use this new, more precise definition, we must very carefully define the referent and how much of it we want to simulate.

Unfortunately, even if we completely understand and specify the referent, we are left with the second obstacle; i.e. specifying the simulation. Carefully defined simulation fidelity is the intuitively correct metric for the simulation unique aspects of our simulation, because in it describes how well the behavior of the simulation matches the interesting parts of the simuland. Unfortunately, the simulation has many other characteristics. These characteristics describe the nature, behavior and character of the simulation, independent of the simuland. Describing the simulation requires a complex, multidimensional set of measures. The simulation's quality (intrinsic and extrinsic), cost (development, scenario, and execution) and extent (decomposition, aggregation and interfaces) must be described, along with its method of control, intended computation environment etc. These characteristics combine with

fidelity to describe the desired simulation. This description could then be used to assess the fitness of a simulation for its intended use. The key point is that carefully specified and measured fidelity is important, but is only one aspect of measuring that "fitness".

2.2 Fidelity-Related Concepts

There are many terms closely related to fidelity such as accuracy, precision, resolution, and so forth, whose casual use adds to the general confusion limiting the practicality of fidelity. The intent of this discussion is to outline the semantic relationship between these terms, as illustrated in Figure 2.2-1. It can be seen from this figure that physical reality, either material or imagined, provides the basis from which all knowledge of reality can be obtained. Known reality manifests this body of knowledge. Known reality also provides the source both for referents, through which application requirements and model or simulation fidelity may be defined, and for abstractions of reality that become models and simulations.

Resolution, error/accuracy, sensitivity, precision and capacity define the terms that describe application requirements and model or simulation capabilities. These concepts also provide a consistent set of measurement units. The fidelity of a model or simulation is defined in terms of the relevant referent and the capabilities of the model or simulation. Fidelity describes the essential characteristics of the model or simulation relative to its referent. Application requirements can be expressed in terms of the tolerances to variation in resolution, error, sensitivity, precision and capacity. These tolerances define the acceptable ranges for all of the dependent and independent variables of all of the dependencies needed to achieve the application's objectives. Comparing an application's tolerances with a model or simulation's fidelity enables the assessment of the model's or simulation's fitness for the application. If a model or simulation meets all of the fitness criteria then it is valid for the application.

Here are the formal definitions for these terms which have emerged from the Fidelity ISG's discussions, and appear in Appendix A.

Accuracy: *The degree to which a parameter or variable or set of parameters or variables within a model or simulation conform exactly to reality or to some chosen standard or referent. See resolution, fidelity, precision.*

Error: *The difference between an observed, measured or calculated value and a correct value.*

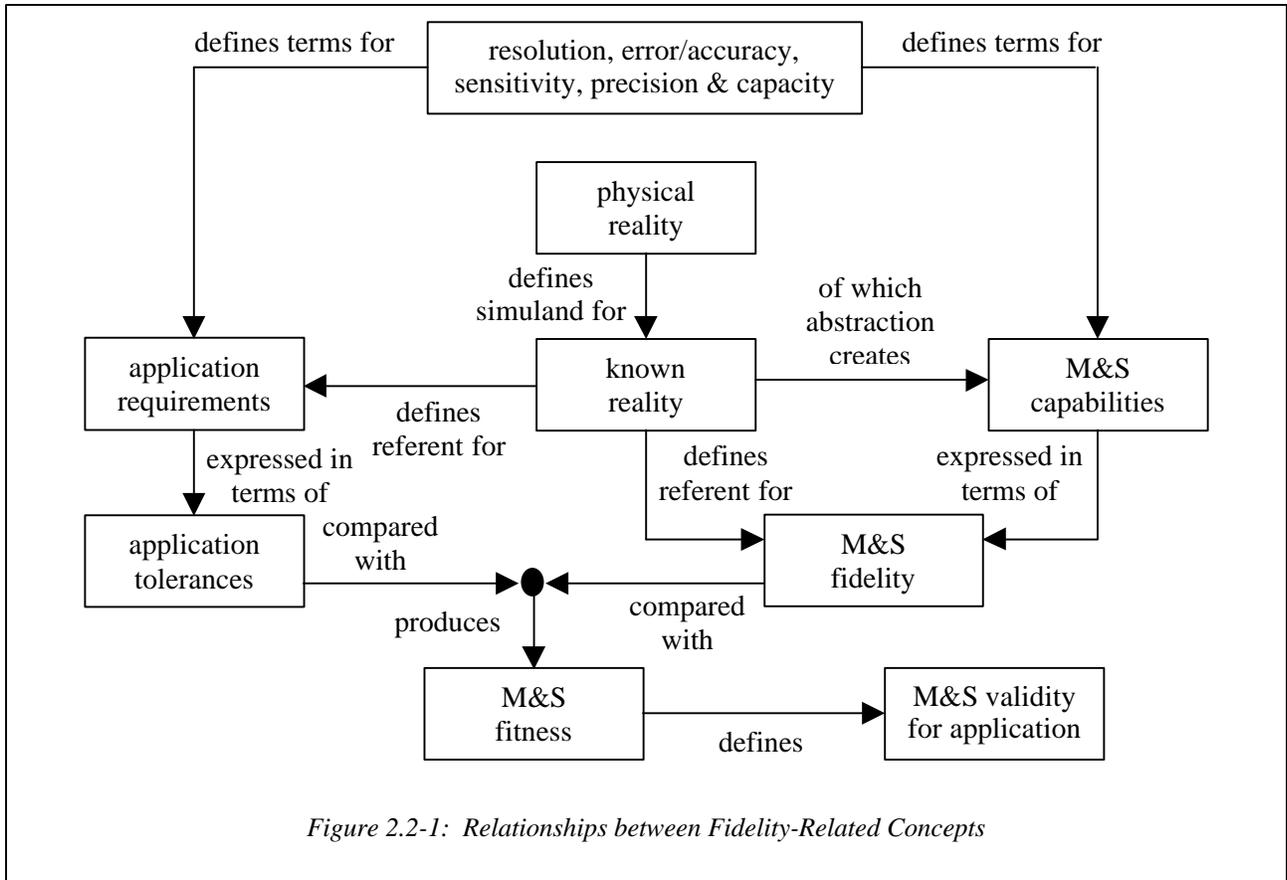


Figure 2.2-1: Relationships between Fidelity-Related Concepts

Fitness. Providing the capabilities needed or being suitable for some purpose, function, situation or application.

Precision: 1. The quality or state of being clearly depicted, definite, measured or calculated. 2. A quality associated with the spread of data obtained in repetitions of an experiment as measured by variance; the lower the variance, the higher the precision. 3. A measure of how meticulously or rigorously computational processes are described or performed by a model or simulation.

Resolution: 1. The degree of detail used to represent aspects of the real world or a specified standard or referent by a model or simulation. 2. Separation or reduction of something into its constituent parts; granularity.

Sensitivity: The ability of a component, model or simulation to respond to a low level stimulus.

Tolerance: 1. The maximum permissible error or the difference between the maximum and minimum allowable values in the properties of any component, device, model, simulation or

system relative to a standard or referent. Tolerance may be expressed as a percent of nominal value, plus and minus so many units of a measurement, or parts per million. 2. The character, state or quality of not interfering with some thing or action.

Validity: 1. The quality of being inferred, deduced or calculated correctly enough to suit a specific application. 2. The quality of maintained data that is found on an adequate system of classification (e.g., data model) and is rigorous enough to compel acceptance for a specific use. 3. The logical truth of a derivation or statement, based on a given set of propositions.

While the specific definitions from the glossary and used in this discussion may be at odds with the definitions from other authors, these formal provide a context for making real progress toward a practical and usable fidelity standard. Where disagreement between these definitions does exist, the effort necessary to restore consistency is minor and the process is well understood. The formal definition of these fidelity-related terms has permitted careful description of their relationships. Hopefully, this clarity will support the work necessary to

create meaningful standards related to modeling and simulation fidelity.

2.3 Applied Fidelity

Despite the confusion about fidelity, its related concepts, and the obstacles to measuring fidelity, the simulation community has made use of fidelity in a number of practical ways. This section addresses the some of the practical ways fidelity has been use.

It is perhaps true that fidelity first becomes a serious issue in simulations when humans participants are introduced. Notice that the number of interactions between a human participant and the various simulation entities are generally much more complex than those

without humans. In low complexity interactions, fidelity can be managed intuitively rather than requiring extensive analysis based. But introducing humans requires an in-depth understanding of human sensory systems and the level of fidelity required in presentations to those sensory systems, in order to achieve a sense of “presence” for the participant.

The following table summarizes how the problem of matching model fidelity to the human sensory system has been addressed. The reader is referred to *Simulation Fidelity in Training System Design* [Hays, R.T. and M. J. Singer, *Simulation Fidelity in Training System Design: Bridging the Gap Between Rreality and Training*, Springer-Verlag, London, 1989].

Table 2.3-1: Fidelity Applications with Human-in-the-Loop Simulation

Sensory Input	Model Features	Issues and Notes
Motion Cues	<ul style="list-style-type: none"> • Fixed base • Motion base (onset G’s) • Centrifuge (sustained G’s) G-suit • G-seat Vibration effects 	<p>Test subjects typically become so immersed in flight simulation exercises that motion will be perceived even when in fixed-base testbeds. Much has been written about the questionable value of motion bases, particularly in view of their high recurring and non-recurring costs. However, motion cues are needed when evaluating PVI techniques under high G conditions (including reach analyses and effects of G loading on perception), vibration effects, and the contributions of G stress to workload. Care should be taken to employ sustained-G techniques when evaluating the effects of continued acceleration, as onset-G techniques, G-suits and G-seats provide only the psychological impression of sustained motion. Vibration effects may be critical in evaluation voice input/output systems and touch-activated screens.</p>
Visual Cues	<ul style="list-style-type: none"> • Out-the-window (OTW) scene: <ul style="list-style-type: none"> – None or rudimentary – Narrow OTW field of view – Wide OTW field of view • Terrain models <ul style="list-style-type: none"> – Area simulated – Moving land-based objects 	<p>The rudimentary OTW scenes of most micro-simulators are appropriate for basic technological capabilities screening. BVR engagements can also generally be supported with simplistic visual cues. PVI devices that are tightly coupled with visual scenery (e.g., helmet-mounted sights), must be evaluated with high-fidelity, wide field-of-view visual imagery. In WVR engagements, much of the SA-related workload depends upon the external visual scene and the mapping of cockpit displays tot he external world; Thus-high-fidelity imagery is often mandated. The price/performance ratio of visual scene generation systems has improved markedly in recent years, rendering it feasible to use such systems in cases where their need may be uncertain. The costs of such use are moderate if the visual system has already been integrated and tested. High-fidelity nap-of-the-earth imagery will be required if low-level evasive flight, terrain masking and clutter are included in the evaluation.</p>

	<ul style="list-style-type: none"> - Realistic aircraft models - Dusk/night/fog effects 	
Artificial Participants	<ul style="list-style-type: none"> • Digital models <ul style="list-style-type: none"> - Threat aircraft - Friendly aircraft - Threat weapons - Threat tactics • Crewstations: <ul style="list-style-type: none"> - Rudimentary controls - Full crewstations 	<p>Care needs to be taken to ensure that digital models are not unrealistically powerful in terms of SA, response time, and error rates. Digital models can, in some scenarios, serve effectively as wingman, if the evaluation does not focus on inter-flight communication or distributed decision-making across the flight. Digital participant models (“digies”) are often used for less active participants, e.g., neutral aircraft or bombers under escort.</p> <p>One or more full crewstation domed simulators are often used in conjunction with interactive stations with rudimentary controls, e.g., MIL-AASPEM. Such stations provide an excellent means of supporting MvN engagement simulation. Caution is warranted in assessing workload or SA for pilots at rudimentary stations. Also, care needs to be taken to ensure that such pilots do not have unrealistically <u>good</u> SA at such workstations, due to the coverage and accuracy sometimes given to the simplistic displays.</p>
Noise Effects	<ul style="list-style-type: none"> • Cockpit-external noise <ul style="list-style-type: none"> - Weapons - Auditory Damage effects - Wind - Engines • Cockpit-internal noise <ul style="list-style-type: none"> - Warning tones - Voice communication 	<p>The high ambient noise levels characteristic of current cockpits should be carefully considered for inclusion in any exercise which seeks to validate a crewstation, particularly those which call for automatic voice input/output. Extended exposure to high noise levels increases stress and should be considered on that basis, especially when dealing with lengthy flights. Engine noise and damage effect noise can be a valuable source of information, and thus may augment SA; conversely, extraneous noise may override other information transfer and thus degrade SA. Evaluations dealing with SA in a robust operational fashion should take these effects into account in determining the need for high-fidelity sound effects.</p>
Cockpit Displays	<ul style="list-style-type: none"> • Number of displays • Size of displays • Resolution • Placement • Color 	<p>The number, type, size and placement of displays are normally explicitly called out as independent variables in the crewstation evaluation process. The CAT process calls for the gradual-buildup of display components through successive levels of evaluation, with the final level (crewstation evaluation) requiring a full suite of high-fidelity displays in realistic juxtaposition. Partial configurations and lesser fidelity must always be considered in light of test objectives and data requirements; Is the validity of the evaluative measures compromised by the display configuration? If displays are omitted, has their future interaction been considered? Or will the present test need to be run? Are the resolution and size of the displays appropriate for the anticipated period of possible deployment? The use of alternative displays in secondary (rudimentary) combat stations is a valid technique, as discussed above.</p>

Tactile	<ul style="list-style-type: none"> • Feel of controls and switches • Personal equipment <ul style="list-style-type: none"> – Gloves – Helmet – Bio/chem defense clothing • Vibration effects 	<p>Crew station validation must take into account the realistic feel of controls and switches, especially when using operational aircraft as baselines, with test subjects who are experienced with those aircraft. Earlier evaluations (capabilities screening, analytic evaluation) may depart from high realism if test objectives and measures are not compromised. Training can compensate for slight variations between simulated and actual crewstations to a limited extent.</p> <p>The interface of personal equipment should be taken into account in the design of PVI concepts. Such equipment is required in any thorough crewstation validation if analysis indicates that it will be a significant factor in PVI effectiveness. Earlier analytic screenings can determine the presence of significant interaction without necessitating high-fidelity simulation.</p>
Temperature Effects	<ul style="list-style-type: none"> • Extreme or prolonged exposure to heat or cold 	<p>These features are mandated when identified as experimental variables of interest.</p>

Fidelity has also been practically applied in physical modeling, for example in modeling aircraft performance.

The following table outlines how fidelity considerations have impacted physical modeling.

Table 2.3-2: Fidelity Applications to Physical Modeling

Aerodynamics	<ul style="list-style-type: none"> • Aircraft models <ul style="list-style-type: none"> – 3- to 6-DOF – Flight control – Propulsion • Weapons models <ul style="list-style-type: none"> – Surface-to-air-missiles – Air-to-air missiles – Guns – Bombs 	<p>Fidelity requirements stem directly from test objectives and evaluative measures. If, for example, tracking error is used as a measure, then the aerodynamic models of the platform and the target must be high enough to provide stable closed-loop performance. Issues include data capture rates vs. model update rates vs. control sampling rates.</p> <p>High-fidelity missile models are not customarily required for PVI evaluations; simplistic 3 DOF models are often sufficient, depending upon the measures to be captured. If the flight characteristics are not essential to the evaluation, a flyout model may be replaced by a rudimentary algorithmic calculation (deterministic or stochastic) of time of flight and endgame results. High fidelity is required in cases in which pilot activities during time of flight are of interest (e.g., evasive actions or missile guidance support). For evaluations in which detailed missile performance data are required, determinations include: minimum and altitudes; number of missiles simultaneously in flight; and sensitivity to target aspect angle. If aspect angle is important, 6-DOF target models will be required. If detailed models are required, classification of the facility and data may be mandated.</p>
Sensor Models	<ul style="list-style-type: none"> • Radar models <ul style="list-style-type: none"> – Airborne (both Blue and 	<p>Most PVI evaluations will be exempt from the stringent modeling requirements imposed upon sensor models used for weapons systems effectiveness analyses, except during crew-</p>

	<p>Red)</p> <ul style="list-style-type: none"> – Surface (Sam and AAA) • EO/IR sensors • RWR 	<p>station validation or when required by the attributes of a particular automation concept. Issues include ECCM vulnerability; clutter effects; simultaneous track capacities; simultaneous search and track capabilities; inter-sensor interference (e.g., self-jamming); and cross section sensitivity (may require 6-DOF models of targets); field of regard; and tactical range. Highly detailed and threat-specific models may require classification of the facility and data. Environmental effects (rain, clouds, glint, sun, etc) are important factors when assessing systems involving IR sensors. Another important factor is the fidelity of the correlation between the sensor display and the out-the-window view. The bottom line: Are the critical test measures sensitive to a given feature?</p>
Avionics Models	<ul style="list-style-type: none"> • Navigation (NAV) • Communication (COMM) • Stores • Mission computers 	<p>Normally fidelity is sufficient to represent the data interface between the cockpit controls and displays. However, higher fidelity in the modeling of performance of some subsystems (e.g., NAV) may be required for cockpit automation and sensor fusion studies.</p>

2.4 Fidelity Requirements

2.4.1 Physical

Physical fidelity refers to the degree to which the layout and feel of surfaces exposed to the participant match the “real” environment. Physical fidelity requirements have generally been dealt with at two levels. First, many applications do not require a great deal of realism, and therefore the resulting physical implementation bears only an incidental relationship to the real system. This approach is frequently seen in system test simulations. The other extreme is when the simulation must very closely approximate reality, in which real system controls and displays are frequently used, and the geometric relationships of the controls and displays is very close to reality. This approach is seen most frequently in training simulators, but occasionally also in design simulations. There is very little work on the range of approaches between these extremes.

2.4.2 Visual

Visual fidelity refers to the fidelity of such data as is electronically generated and presented to the participant’s eyes as a surrogate for the real-world view that would normally obtain. There are (at least) 3 major potential problems for the human participant, with around a dozen contributory factors (depending on how they are decomposed). The problem areas are:

- 1) Depth Compression
- 2) Navigation
- 3) Simulator Sickness

The depth compression and navigation problems both arise from the fact that human visual space is not Euclidean in nature. The perceived distance of an object is less than the real distance by a factor that increases with an increase in real distance. Further, the human visual field is “wrapped round” the participant leading us to talk of a parabolic visual field. A good visual display will provide cues to the human perceptual system that mimic this parabolic field. It has been observed that with low-grade visual displays (e.g., SIMNET) there is a tendency to make navigational errors (in the form of positional judgments) for items not on the main axis of vision. This effect reduces with experience. In these two cases therefore, visual fidelity refers to the degree to which the system replicates natural world data, minimizing judgment errors by participants.

Simulator sickness is generally held to be caused by the simulator not providing the same cues (NOT limited to visual) as would be provided to the participant by the real world equipment. However, manipulation of visual data is regularly used to make up for the Simulator not having the full range of real world movements.

Factors that bear on visual fidelity include:

- Brightness
- Contrast
- Resolution
- Level of Detail
- Pictorial Cues
- Anti-aliasing
- Frame update rate
- Atmospheric effects
- Visual Dynamic effects

Brightness refers to the overall range of brightness of the display -- think midnight to full noon in summer. Strictly speaking, brightness is the subjective term for luminance. Brightness fidelity can be expressed as a percentage (or decimal) of the maximum real world conditions being simulated since currently available displays will not attain the brightness of the real world. On the other hand, a dull day could be effectively reproduced at 100% fidelity.

NOTE: CRT displays can take up to an hour to reach maximum stable luminance, and this will affect both brightness and contrast.

Contrast refers to the difference between light and dark areas of the visual display (strictly, between maximum and minimum luminance). There are at least three accepted (and widely disparate) methods for defining contrast and no current technique exists for easy measurement of detailed scenes or those with rapid change of detail. It is held that the minimum suitable range of contrast is task dependent.

Note: Contrast must be linked to brightness/luminance, and - pending suitable measurement techniques - can only be evaluated in subjective terms.

Resolution, very specifically here the resolution of the visual display, refers to:

- 1) The level of detail in the display in the sense of the number of points available for "drawing" the picture. This is normally expressed as pixels wide x lines deep (assumes raster scan). The point size or "dot pitch" moderates the visual effect.
- 2) The number of polygons 3 or 4 sided, or the number of edges that can be displayed at the specified frame update rate.

Effectively, these factors combine to provide a certain "quality" of display. A reasonable yardstick is the 50,000 edges that an NTSC television can display, but which few simulators can attain. Thus, a measure of fidelity in respect of resolution of the visual display must account for frame rate, drawing quality, number of edges.

Note: European PAL television will show more edges than NTSC, and digital video will do better.

Note: "resolution" as above can be decomposed into more factors.

Level of Detail is related to resolution as above, but is a separate issue. To minimize the processor load/computation time, displayed scenes normally have the level of detail reducing - in steps - with increasing distance from the participant. This can produce some aberrations at the step boundaries. A typical example is the large rock that is visible at 2000 meters, which disappears if the distance increases to 2001 meters revealing an enemy vehicle. This vehicle can be fired upon but not hit as the rock is still there. This is probably more of a problem for ground based tactical simulators, since those for fast jets have a display that changes at a faster rate ("things happen much more quickly"), thus concealing the aberration.

Pictorial cues can enhance, or reduce fidelity. The screen of a CRT is flat for practical purposes leading to a 2-D picture. A frame around this provides more 2-D information. If the simulation is of a tank gun-sight that is optically a framed 2-D picture, then fidelity is improved. A collimated projected display (e.g., a flight simulator) is effectively at optical infinity, which can cause visual problems e.g., when the runway is only some 10 feet below the pilot (think helicopters). The list of pictorial cues is extensive, cues can enhance or reduce (combine or conflict) the subjective fidelity of a visual display, and the relevant cues will be application related. As example, nap-of-the-earth flight, and flight refueling will require different cues.

Anti-aliasing refers to measures taken to avoid the "stair-stepping" of diagonal lines (or edges). This effect reduces as display resolution increases. Some techniques manipulate the luminance of pixels adjacent to the line, and this can have an effect on distance judgments.

Frame update rate. The number of times per second that the display is updated (frequently a function of processor power, and hence cost). As this rate falls below 24 frames/second the display becomes apparently more jerky. Lower rates pose difficulties for such tracking tasks as tank gunnery - at rates below 15/sec, gunnery is almost impossible.

Thus, 24 frames/second should be minimum acceptable, and the effect of lower rates is probably not linear.

Atmospheric effects can be considered as similar to the pictorial cues. In the natural world, atmospheric scatter leads to objects of increasing distance becoming lighter in hue, and of lower contrast.

Visual Dynamic Effects can be transitory (muzzle flash temporarily blanking out a tank sight, smoke from muzzle or munition detonation, cloud) or permanent (dynamic terrain that craters with a shell-burst) and can include lighting due to sunlight and sun movement.

2.4.3 Audio

This can relate to any audio component within a simulation (e.g., communications traffic); the audio effects within a simulator (e.g., engine noise); or the environmental audio effects surrounding a simulation (e.g., thunder).

Aspects of audio fidelity include

- 1) Attenuation due to weather or terrain effects,
- 2) Signal to noise ratio,
- 3) Weapon effects (think of a tank main armament, or an APC firing a chain gun),
- 4) The linkage between throttle setting and engine noise (and gear selected if a land vehicle), and
- 5) The noises from other activity in the vicinity if the situation being simulated.

2.4.4 Motion

In the real world, there are six degrees of freedom (DoF): 3 rotational, 3 translational. While no simulator actually provides this; in practice the combination of a motion platform and appropriate visual cues can get very close to an apparent 6 DoF.

Some areas of motion fidelity are easy to quantify, for example the role rate for an aircraft for example, or the relationship between turn radius and gear ratio on a wide range of British tanks. Other areas are more complex, for example the “twitch” when a tank fires its main gun, or providing both backward and upward recoil on a small arms simulator.

In general, creating high fidelity models of fairly continuous, smooth motion is easier than doing the same for sudden, sharp events. The need for motion effects can be assessed by considering the purpose of the simulation and asking what effect on that purpose will

the non provision of that motion effect have. Quantifying the effect is of course much harder.

2.4.5 Environment

Issues about fidelity in environmental model can relate to the environment within the simulation, or to the environment in which the simulation operates.

Within the simulation, the fidelity of the environment is concerned about correlation with the natural as well as tactical environment in which the simulated entities operate. Weather representation is probably the most straightforward issue in environmental fidelity. Consider the fidelity of simulating rainfall: this may affect visual range, communications attenuation, and artillery ballistics. If we are operating within a federation of simulations, does my weather match yours? How many different aspects of weather need to be considered for fidelity issues? How many other things like “weather” might we have to consider? Can we generate an appropriate abstraction into which different factors could be plugged?

Fidelity considerations for environment outside the simulation generally appear when the simulation is being used as a piece of test equipment. Let us say “test a new C&C structure” and now operate the kit when we are wearing jeans and sweaters at 25C in the lab. Of course it will work when the operators/users are in foul weather gear, won’t it. No simulation is used in isolation, but in the context of some socio-political system; should the simulation have to take account of this and, if so, to what level of fidelity?

2.4.6 Temporal

Temporal fidelity is a special case of environment, which is generally of more interest than other environmental issues because one of the things simulation is intended to study is the change in systems over time. Temporal fidelity relates to the relationship between the simulation and the real-time world being simulated. There are two classes of temporal fidelity issues: those within the simulation and those outside of it.

Temporal fidelity issues within the simulation raise questions such as, “Does acceleration from speed x to speed y take the same time, or, is a shell flight time true to the real world? “. Note that this is closely related to behavioral fidelity, and the relationship can get even more complex. For example, consider a tank gunner tracking a target on a visual display that updates too slowly, and where the target changes speed as a step function rather than by acceleration, and where the turret slewing speed is wrong. Which aspect of fidelity should take precedence?

Temporal fidelity issues outside of the simulation raise questions such as "Can the simulation reproduce all required events at real-time speed?" and "Can the simulation be run at other than real-time speed?".

It is interesting to note that either internal or outside temporal fidelity can be satisfied by discrete event or continuous time simulation engines, although one approach may be more natural than the other for a specific problem.

2.4.7 Behavior

Much of the points in the foregoing discussion could also be considered under the "behavior" heading. Weather should behave like real weather does and with the same effects. Simulator motion (or its lack) could be behavioral. A visual depiction of a vehicle should change size according to optical laws, and do it the same every time; vision or behavior? Does that vehicle have a multi-speed transmission? If so does it accelerate through the gears, or have a number of pre-set speeds?

2.4.8 Aggregation

The fidelity of aggregated entities is extremely complicated, as the dependencies between the entities are obfuscated by the aggregation. How is the fidelity of a model affected by aggregating models?

Let us assume that $F(x)$, the fidelity function, is a function from the space of subsets of simulations/data (S/D) onto the unit interval with the meaning that $F(A)=1$ means that A has perfect fidelity and $F(A)=0$ means that A has no fidelity, where A is a subset of some simulation/data combination. Consider elemental models A and B for which $F(x)$ is defined, with $F(A)=x$ and $F(B)=y$.

What is $F(A + B)$? Is it additive? It is difficult to conceive a fidelity function such that $F(A+B)= x + y$, since fidelity is most often conceived as unitless and defined. Some authors have argued that $F(A+B)=\min(F(A), F(B))$, $F(B)=\min(x,y)$ for each component of fidelity. While this has a certain appeal, one can conceive of cases where the lack of fidelity in one model is compensated by the fidelity in the other. It seems difficult to define a fidelity aggregation function.

Can $F(A + B)$ be defined in terms of the algorithm '+' that is actually used to join the two elements? Perhaps, but in many cases it may be simpler to measure $F(A + B)$ directly (against the designated referent).

2.5 Fidelity Applied to Interoperability

2.5.1 Conceptual Model Development

A principal reason supporting the development of a simulation conceptual model or a federation conceptual model is to provide an explicit mapping between the objectives of the developmental effort and the design implementation. The following quotation describes the significance of fidelity in this process when the objective is to build a federation of compatible participants.

"During the Conceptual Analysis phase, the federation developer produces a conceptual representation of the intended problem space based on his/her interpretation of user needs, federation objectives, and the defined environment. The product resulting from this phase is known as a Federation Conceptual Model (FCM). The FCM is an implementation-independent representation, which serves as a vehicle for transforming objectives into functional and behavioral capabilities, and provides a crucial traceability link between the federation objectives and the design implementation. The FCM can be used as the structural basis for the overall design and development of the federation, and can highlight correctable problems early in the federation development process when properly validated.

As the FCM evolves, it will greatly facilitate the federation developer's understanding of the real world domain. As this knowledge is acquired, a set of detailed federation requirements can be developed. These requirements, based on the original Objectives Statement, should be directly testable and provide the implementation level guidance needed to design and develop the federation. The federation requirements should also explicitly address the issue of fidelity, so that those fidelity requirements can be considered during selection of federation participants. In addition, any programmatic or technical constraints on the federation should be refined and described to the degree of detail necessary to guide federation implementation." [High Level Architecture Federation Development and Execution Process (FEDEP) Model, Version 1.2, May 26, 1998].

Just as the FCM provides an implicit contract between the federation participants, the Simulation Conceptual Model (SCM) provides an explicit representation of what the simulation developer intends to develop in order to satisfy the requirements of the simulation sponsor. In both cases, the levels of fidelity provided down to the level of individual transactions between objects in a

simulation, and federates in a federation should be explicitly addressed, less errors, omissions, and assumptions lead to recriminations later. Fidelity specification is one way of reducing the possibility of non-communicating federates in a federation, and unfulfilled expectations in simulation development. If fidelity is not specified prior to the commencement of model development, the models developed will risk not satisfying sponsor or other federate expectations and not achieving compatible levels of fidelity within cooperating parts of a single model.

2.5.2 Federation Development

A key responsibility of federation developers is to design in such a manner that sponsor needs will be satisfied. In order for sponsor needs to be satisfied they must first be identified.

"The primary purpose of Sponsor Needs Identification is to develop a problem statement. The sponsoring agency is responsible for preparation of this product. The needs statement may vary widely in terms of scope and degree of formalization, but should include, at a minimum, high-level descriptions of critical systems of interest, coarse indications of required fidelity and resolution for simulated entities, and output data requirements. In addition, the federation sponsor should also indicate the resources which will be available to support the federation (funding, personnel, tools, facilities, ...) and any known constraints which may affect how the federation is developed (due dates, security requirements, ...). In general, the federation sponsor should always include as much detail and specific information as is possible at this early stage of development." [High Level Architecture Federation Development and Execution Process (FEDEP) Model, Version 1.2, May 26, 1998].

The identified sponsor needs become the requirements guiding federation development. Specifically, the "required fidelity and resolution" described above constrain the selection of "potential federation members", as described in the following excerpt. "The next major activity (of Federation Design) is to determine the suitability of individual simulation systems to become members of the federation. This is normally driven by the perceived ability of potential federation members to represent required objects and interactions at an appropriate level of fidelity." [High Level Architecture Federation Development and Execution Process (FEDEP) Model, Version 1.2, May 26, 1998].

If the federation is to meet the objectives of its sponsors, it is critical that appropriate fidelity specifications guide

its development. Similarly, if a simulation is to meet the objectives of its sponsors, appropriate specifications of fidelity must guide all developmental phases.

2.5.3 Scenario Development

An important element in translating federation objectives into conceptual models is the federation scenario. Specifying a scenario provides a context for identification of federation components and specification of their behaviors over time.

"The purpose of this phase (Scenario Development) is to develop a functional specification of the federation scenario. The primary input to this activity is the operational context constraints specified in the Objectives Statement. The composition of a federation scenario includes:

- 1) *An identification of the major entities that must be represented by the federation;*
- 2) *A functional description of the capabilities, behavior, and relationships between these major entities over time; and*
- 3) *Specification of relevant environmental conditions which impact or are impacted by entities in the federation.*

Initial and termination conditions should also be provided. Multiple scenarios may be developed during this phase, depending on the needs of the federation. A single scenario may also support multiple vignettes, each representing a temporally ordered set of events and behaviors. The product of this activity is a Federation Scenario Specification (FSS), which provides a bounding mechanism for conceptual modeling activities, and also impacts the security level at which the federation operates." [High Level Architecture Federation Development and Execution Process (FEDEP) Model, Version 1.2, May 26, 1998].

Among the operational context constraints of the Objectives Statement are the specifications of the fidelity required. Following these fidelity requirements in the Federation Scenario Specification provides a more detailed bounding mechanism for use in the conceptual modeling activities.

2.5.4 Federation Integration & Testing

"The purpose of this phase (Federation Integration and Test) is to bring all of the federation participants into a unifying logical

operating environment to test that the federates can interoperate to the degree required to achieve federation objectives. There are three levels of testing defined for HLA applications, which are described as follows:

Compliance Testing: In this activity, each federate is tested individually to ensure that the federate software correctly implements the HLA requirements as documented in the HLA Compliance Checklist.

Integration Testing: In this activity, the federation is tested as an integrated whole to verify a basic level of interoperability. This primarily includes observing the ability of the federates to exchange data as described by the FOM.

Federation Testing: In this activity, the ability of the federation to interoperate to the degree necessary to achieve federation objectives is tested. This includes observing the ability of federates to interact according to the defined scenario and to the level of fidelity required for the application. This activity also includes security certification testing if required for the application." [High Level Architecture Federation Development and Execution Process (FEDEP) Model, Version 1.2, May 26, 1998].

The key phrase in the above is "level of fidelity required for the application." In order that such a level of fidelity can be obtained, several issues must be resolved, including:

- Fidelity in data parameters passed between interoperating models,.
- Fidelity in computational accuracy of interoperating models, and
- Fidelity in response timings of interoperating models.

Without achieving consensus among federates with regard to these issues, models will not interoperate. Although determining computational accuracy of interoperating models may be difficult, it should be determinable if the expected range of inputs will support the expected or desired outputs for specific models.

Note that the above quotation addressed federation integration and testing. Similar concerns are addressed between cooperating parts of a simulation at system integration and testing. According to the system development process employed, these issues may also be addressed earlier in the process.

2.5.5 Exercise Management

A major concern in exercise management is representational fidelity, as discussed above. An issue here being that even the use of an accredited model does not guarantee credible results if different data sets are used than were used in the examinations supporting accreditation. Ideally, the earlier accreditations will cover ranges of values and combinations of values; however, even if this is so, the current exercise may wish to use data values that are outside any such ranges. The key issue here is that exercise requirements will indicate a specific constraint on fidelity. The representation must achieve a fidelity satisfying that constraint based on the appropriate combination of data and models employed.

2.5.6 VV&A

Concepts of fidelity are intimately entwined with the Verification, Validation, and Accreditation (VV&A) Process. Fidelity and related concepts are particularly important to the validation process.

"At the risk of oversimplification, verification focuses on M&S capability, whereas validation focuses on M&S credibility. Verification ensures that a simulation meets all the requirements specified by the user and that it implements those requirements correctly in software; validation ensures that a simulation conforms to a specified level of accuracy when its outputs are compared to some aspect of the real world." [10].

The importance of fidelity concepts in tying simulation or federation objectives and requirements to the simulation representation is described in the following paragraph:

"Application objective and requirements dictate how faithful the representation of a process, phenomenon, or system must be when compared with the real world for simulation results to be considered useful. Sometimes, 60 percent representation accuracy may be sufficient; sometimes 95 percent accuracy may be required, depending on the type or importance of decisions that will be made based on the simulation results. Because the requirement for accuracy varies with the intended application, it is clear that a model's or simulation's credibility must be judged with respect to application-specific requirements and objectives. The objective should always be used in front of terms such as credibility, validity, or accuracy to indicate that the judgment of validity has been made with respect to application-specific requirements." [10].

Representational fidelity is key to dealing with the interactions between model and data. "The credibility of M&S results is related directly to the credibility of data used as input to or resulting from model use. Data need to be reviewed for accuracy and consistency, ..." [10].

This subsection has described how concepts of fidelity support VV&A, particularly validation. Validation is concerned with whether the representations of real world objects satisfy sponsor objectives and requirements, and fidelity concepts, e.g., representation accuracy, are tools used making such determinations. An important concern in this area is the interaction between model and data; in addressing this concern fidelity concepts, e.g., accuracy and consistency, are significant tools.

2.5.7 Security

Many security issues are related to fidelity, and an adequate specification of fidelity as a boundary condition will often greatly simplify their resolution. In the following paragraphs, we discuss three such security concerns.

Simulation development is often complicated by the classification of the object representation. In some cases, the model is not classified but the data is. The costs and efficiency of development can be greatly improved if the development environment can remain unclassified. To obtain substantive benefits from an unclassified development environment, much testing must be done at an unclassified level. This requires unclassified surrogate data sets. Development of these can be very challenging, since the persons responsible for developing them will probably have knowledge of the classified data sets. Clear demarcation of classified/unclassified boundary conditions, including fidelity, will facilitate this process.

A similar condition may exist in the training arena, due to models being unclassified, but representation using classified data being classified. When the training objective is acquisition of basic skills, fidelity requirements are less stringent than when conducting mission rehearsal or actual mission execution. Training is facilitated by use of an unclassified environment. Thus, if unclassified data sets can be used for training, costs and efficiency are improved. Classified data can then be used for mission rehearsal, and if applicable, during mission execution. Again, development of unclassified data sets is facilitated by clear specifications of fidelity boundary conditions.

A knowledge development process may be subject to unintended elevation of classification level. In the defense operations domain, most high-level knowledge is

based on unclassified doctrine, and is not subject to classification. However, building a credible simulation may require a much greater level of detail, which typically would be drawn from subject matter expert experience. The problem is that such detailed knowledge of US command and control systems may expose vulnerabilities, facilitating attacks on infrastructure by enemies. Thus, articulation of unclassified material

2.5.8 M&S Tools

There are many tools supporting M&S development: Hunt et al identify a set of fifteen components supporting HLA development. [15] The following discussion is limited to two tools whose early development was supported by the Defense Modeling and Simulation Office (DMSO).

1) CMMS

One tool becoming widely used in the M&S community is the Conceptual Models of the Mission Space (CMMS) Toolset.

"As employed by DoD simulation programs, CMMS is:

The disciplined procedure by which the simulation developer is systematically informed about the real world problem to be synthesized,

The information standard the simulation subject matter expert employs to communicate with and obtains feedback from the military operations subject matter expert,

The real world, military operations basis for subsequent, simulation-specific analysis, design, and implementation, and eventually verification, validation, and accreditation/certification, and

A singular means for identifying re-use opportunities in the eventual simulation implementation by establishing commonality in the real world activities. " [6].

The significance of fidelity in building domain descriptions is addressed in the following quotation:

"The essence of CMMS is the collaboration (usually called knowledge acquisition) between the warfighter and the simulation developer to:

- *Establish the simulation focus to support a real world purpose,*

- *Select a representation (the combination of model and data which determine granularity, inclusion of detail, and level of fidelity),*
- *Construct a simulation implementation independent, conceptual description of the real world which specifies the representation selected. "[6].*

The proceeding clearly indicates that the appropriate level of representational fidelity is central to the production of a consistent, useful domain knowledge product.

2) Object Model Development Tool

The Object Model Development Tools are the first in a suite of federation development support tools designed to be extensible, open and interoperable.

"Object Model Development Tools (OMDTs) - Automate the data entry and consistency checking needed to specify simulation object models (SOMs) and federation object models (FOMs)." [15]

The Object Model Template (OMT) is a developing IEEE Standard which defines the information which must be provided in a SOM or FOM. Datatypes define the level of attribute fidelity, "accuracy" and "resolution", to be provided to a federation by a federate. The simple datatype shall be used to describe simple, scalar data items. ... The fourth column (Resolution) shall describe the precision of measure for the datatype. For datatypes of scalar numerical measures, the resolution column may contain a single-dimensioned numeric entry for each row of the table. This value may specify the smallest resolvable value separating values that can be discriminated. However, when such values are stored in floating point datatypes, their resolution so defined might vary with the magnitude of the attribute value. Hence, in these cases and others, a better sense of the resolution may be conveyed by the datatype. "N/A" shall be entered in this column for datatypes for which resolution information does not apply. The fifth column (Accuracy) shall describe maximum deviation of the attribute value from its intended value in the federate or federation. This is ordinarily expressed as a dimensioned value, but it may also be perfect for many discrete or enumerated attributes. "N/A" shall be entered in this column for datatypes for which accuracy information does not apply.' [16].

3. Proposed Fidelity Framework(s)

In this section, we summarize four of the different fidelity “frameworks” that have been put forth within the SISO community. The reader is warned that the frameworks may be mutually incompatible, and may not use terms and currently defined in the fidelity glossary developed by the ISG.

3.1 Fitness for Purpose

This work is elaborated in reference [21].

One of the most critical tasks in the simulation development process is the determination of the required level of simulation fidelity in order for a simulation exercise to achieve its goals. Fidelity is defined as a measure of realism of a model or simulation. **Warning:** these definitions are not necessarily consistent with the current Fidelity Glossary.

Technically it is impossible to achieve a 100 % accurate representation of all aspects of the real world. Furthermore, it would be too expensive to do so. In order to achieve affordability or performance goals, almost all simulations have lower accuracy and resolution of representation for aspects of the real world which are not or less relevant to achieve the simulation goals. It is therefore important to help the simulation developer to define the required level fidelity necessary to achieve the simulation exercise goals. Based on this formally defined required level of fidelity it is possible to pick those simulations from a Modelling and Simulation repository that can comply with the required level of fidelity needed for a specific purpose. This requires that its creators, using a similar fidelity description, properly define the available levels of fidelity of existing simulations or models. The approach described next can be used for defining both the required and available level of fidelity for a simulation or model and selecting suitable simulations for a given purpose

3.1.1 Conceptual Framework

The premise of this fidelity measurement approach is that a simulation is first decomposed down to a level of detail that allows the performance of the simulation components to be compared to the equivalent aspects of the real world. Next the actual comparison of these performance characteristics is performed. Therefore, the proposed approach for defining the required and available level of fidelity is based on a fidelity description consisting of two major parts:

Resolution, the extent to which the simulation models each aspect of the real world. **Warning:**

these definitions are not necessarily consistent with the current Fidelity Glossary.

Accuracy, the agreement between the performance of these models of each aspect and the real world performance. **Warning:** these definitions are not necessarily consistent with the current Fidelity Glossary.

A very low-level of decomposition is necessary to determine whether the simulation fidelity supports the simulation exercise goals or purpose. Creating such a low-level decomposition is of course an expensive and time-consuming job, and therefore a tool is necessary to help simulation developers defining the level of fidelity for a simulation.

3.1.1.1 A Process for Defining the Available Fidelity

The simulation owner is the person that conducts the description of the available level of fidelity of a simulation. This goal is achieved by creating a tool, which asks the simulation owner a series of questions that delineate the level of fidelity in terms of resolution and accuracy. The intent of these questions is to walk the owner of the simulation through various aspects of the real world and ask the owner if their simulation has components and sub-components that model that real world aspect. When a question is answered with a yes, the tool would ask progressively more detailed questions to determine the fitness of resolution with which a simulation component models the real world aspect. Finally, it would ask to provide measures of model accuracy for that represented real world aspect. It is assumed that a fidelity or real world referent in a MSRR will describe these measures of accuracy.

3.1.1.2 A Process for Defining the Required Fidelity

To be able to determine the required fidelity it is necessary to obtain the operational definition of the simulation exercise goals. This operational definition consist of the exercise measurements of performance (MOPs) and their derived measurement of effectiveness (MOEs). The focus of a simulation exercise is thus primarily defined by the MOPs, which serve as surrogates for the real world measures of mission success. The premise is Thus to identify those real world aspects that have the greatest impact on the mission success and those having less impact. In order to achieve affordability or performance goals, the aspects with the greatest impact on the MOPs will be represented with higher resolution and accuracy models than those aspects with lesser impact. This goal is achieved by creating a tool, which asks the simulation developer and subject matter experts (SMEs) a series of questions that determine to what extent the real world aspects have

impact on the MOPs. The questions are designed to lead the simulation developer and SMEs through the various aspects of the real world and it is the job of the SME to rate the impact of each real world aspect on the MOP. For this rating a well-defined rating-scale should be used.

3.1.2 Fidelity Metrics

The proposed approach uses a two-staged quantitative index for describing fidelity. The first one is the resolution data to indicate whether the simulation fidelity of a given aspect of the real world is zero (false) or greater than zero (true). A real world aspect with a resolution data value of zero (false) implies that this real world aspect is not taken into account by the simulation. A resolution data value greater than zero (true), implies that the real world aspect is taken into account by the simulation. In that case, the second fidelity index is used to describe how far above zero the simulation fidelity for that real world aspect resides in terms of accuracy data. The second fidelity index may contain various types of accuracy measures depending on what kind of real world aspect it addresses.

For judging the impact of a real world aspect on the simulation MOPs a five point rating scale is proposed: None, Minimal, Significant, Substantial and Critical.

3.1.3 Sample Application(s)

Suppose that the goal of a simulation exercise is to determine the relative cost effectiveness against a next generation threat aircraft of a larger number of less expensive A/A missiles with limited countermeasure capabilities versus a smaller number of more expensive A/A missiles with more sophisticated countermeasures defeating capabilities. As a simulation developer we want to know if there already exist a simulation in our M&S repository that can support this simulation exercise goal. Stated differently search a simulation that fits our purpose.

The first thing that should be done is to define the MOPs for this exercise. A suitable MOPs is the 'Missiles per Red kill for each countermeasures defeating alternative'.

This MOP is entered, along with others, into the fidelity requirements tool. Based on the scenario, the simulation developer determines the entities and their tasks to be simulated, and enters them into the tool. In this case these entities could be a Red and Blue aircraft. Next the simulation developer should ask SME to join him. The tool will present the previously entered MOP, the involved entities and their tasks to the SME, who should determine whether the performance of that entity affects the MOP. For the Red aircraft the SME enters a yes. Then the tool will represent all kinds of aspects of this real world entity that may impact the MOP. In this example, 'Impact of countermeasures on probability of hit in real world' and 'Impact of re-supply of resources', which are rated by the SME as having, respectively, Critical and Minimal effect on the MOP. When the rating for an aspect is other than None the tool will continue asking increasingly more detailed questions about sub-aspects of this aspect. In the end the tool would provide a table for each entity that documents the impact for each real world aspect on the MOP. The next step is to search the M&S repository for matching simulations using the just derived required level of fidelity and the available level of fidelity of the existing simulations. In this case, 'a Red aircraft simulation that has a countermeasure component' is entered in the tool. The tool will present all simulations, which fulfil this fidelity requirement along with their fidelity description. To make the final decision the SME must look at the presented accuracy data. Only an intelligent SME and simulation developer can make this final choose. However the presented approach and tool can help defining the level of fidelity for simulations and to quickly make a good first selection of possible candidates.

Figure 3.1.3-1 illustrates a preliminary notational list of questions to be answered to define resolution and accuracy of platform level simulations. Figure 3.1.3-2 shows a similar notational list of questions to be answered to define the required resolution and accuracy of federation simulations is presented

1. Does simulation model platform motion?
 - 1.1. Position?
 - 1.2. Velocity?
 - 1.3. Acceleration?
 - 1.4. Orientation?
 - 1.5. Other?
2. Does simulation model individual entity components?
 - 2.1. Does simulation model sensors?
 - 2.1.1. For each sensor modeled
 - 2.1.1.1. Sensor type?
 - 2.1.1.1.1. Human eye?
 - 2.1.1.1.2. Radar?
 - 2.1.1.1.3. Infrared?
 - 2.1.1.1.4. Acoustic?
 - 2.1.1.1.5. Magnetic?
 - 2.1.1.1.6. Other?
 - 2.1.1.2. Is sensor effectiveness impacted by target type?
 - 2.1.1.2.1. Physics-based?
 - 2.1.1.2.2. Monte Carlo?
 - 2.1.1.2.3. Other?
 - 2.1.1.3. Is sensor effectiveness impacted by target orientation?
 - 2.1.1.3.1. Physics-based?
 - 2.1.1.3.2. Monte Carlo?

Figure 3.1.3-1: Inquiry into Resolution & Accuracy of Platform Simulations

1. Enter exercise MOEs
2. Enter exercise MOPs
3. Enter entities to be simulated
 - 3.1. For entity No. 1, enter tasks to be performed (extracted from CMMS based on scenario)
 - 3.2. Continue
 - 3.3. For entity No. n, enter tasks to be performed
 - 3.4. For MOP No. 1
 - 3.5. Tool displays MOP No. 1
 - 3.6. Tool lists entity No. 1 tasks
 - 3.7. Does performance of entity No. 1 impact MOP No. 1?
 - 3.7.1. Rate the extent to which each of the following aspects of the real world entity impact MOP No. 1 in the real world. None, Minimal, Significant, Substantial, Critical.
 - 3.7.1.1. Platform motion
 - 3.7.1.2. Position
 - 3.7.1.3. Velocity
 - 3.7.1.4. Acceleration
 - 3.7.1.5. Orientation
 - 3.7.1.6. Other
 - 3.7.2. Individual entity components
 - 3.7.2.1. Sensors
 - 3.7.2.2. For real world sensor No. 1
 - 3.7.2.2.1. Sensor type

Figure 3.1.2-2: Inquiry into Resolution & Accuracy of Platform Simulations

3.2 Cascading Accuracy

This work is elaborated in references [19].

Currently the simulation fidelity is mostly judged using face validation of the end product by Subject Matter Experts (SMEs). Because the SME is unfamiliar with the complex internal structure or architecture of the simulation, the face validation will result in observations that do not correlate to facts. It is a subjective rather than a quantitative judgement, which can lead to wrong conclusions. The only way to ensure successful validation is to have a procedure that determines the fidelity limits of the simulation and models in a quantifiable manner. Fidelity is a measure of realism of model or simulation and is described here in terms of accuracy. Accuracy is defined to be inversely proportional to the error measurements of parameters and variables associated with the components and their interactions in the simulation. **Warning:** these definitions are not necessarily consistent with the current Fidelity Glossary. In short, the less error, the more accuracy and thus also more fidelity. The presented approach consists of error estimation techniques that can be used during conceptual modelling or simulation design to determine and validate the fidelity levels of models or simulations.

3.2.1 Conceptual Framework

In order to assess fidelity, the federation development and execution process needs to be abstracted into three levels, as shown in Figure 3.2.1-1. The first level of abstraction is the reality to be modelled and simulated and is described as a set F. F consists of data gathered from the real world measurements. The second level is

the conceptual model, which contains a description of a part of reality deemed essential for achieving the simulation objectives. In this stage the fidelity requirements, in terms of the maximum allowable error with respect to F, to ensure a valid simulation should be defined. The conceptual model, G, defines in this sense the requirements for the last level of abstraction, the simulation implementation, H, of the conceptual model or federation

Each transition between the levels of abstraction will induce errors. The determination of the conceptual model is based upon the observations of the states of the reality F. Observations of reality are always imperfect and therefore the conceptual model description will differ from reality. Determination of the conceptual model accuracy is thus necessary. Eventually, the conceptual model is implemented in a software and hardware environment. This implementation H will introduce errors due to simulation software design and performance variations of the computer environment. So the total error between reality and the simulation implementation, which represents the fidelity of the federation, is influenced by the two successive abstraction errors. The third abstraction can be decomposed into different layers (see Figure 3.2.1-1). The highest layers are the Behavior models consisting of the high level cognitive behavior and the low-level reactive behavior models. Underneath this layer comes the physical models level which include the sensor, platform and environment models. Below this layer is the infrastructure layer. This layer is composed of the ground truth database, and the software and hardware architecture.

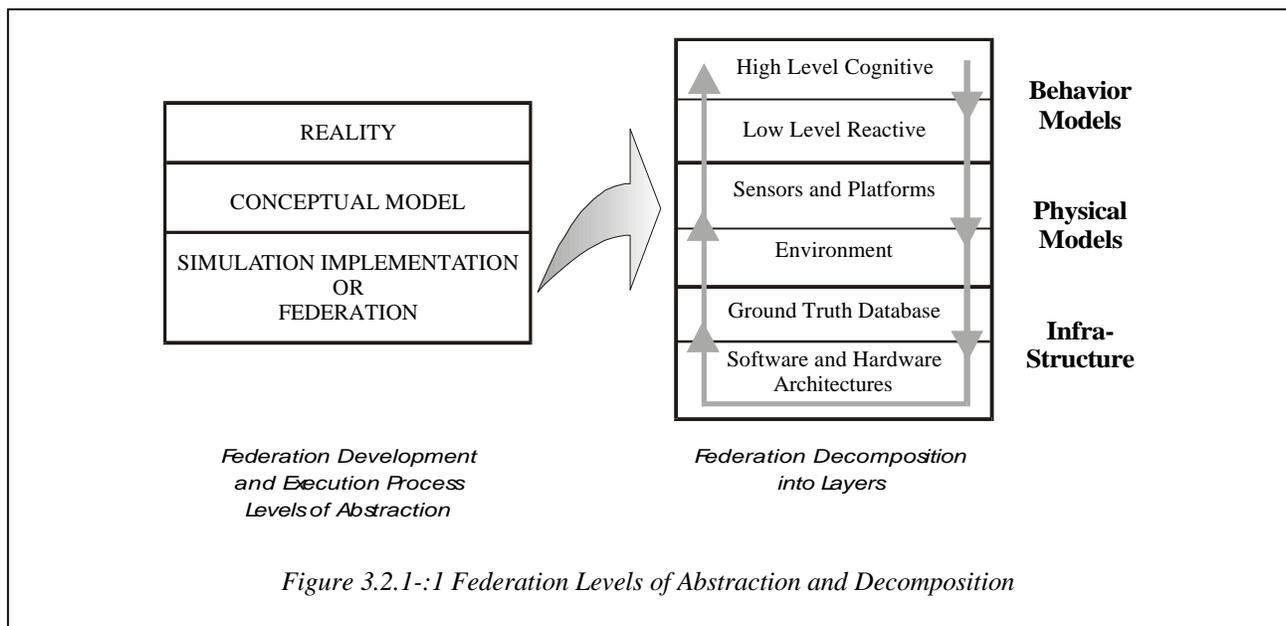


Figure 3.2.1-1 Federation Levels of Abstraction and Decomposition

As showed in Figure 3.2.1-2, a federation is composed of a number of model layers and associated sub-model layers. The required federation accuracy, fidelity, depends upon the highest level model. The greater the number of models layers that data must pass through to reach it, the larger the potential error is for that model. Since each sequential model layer that manipulates a simulation state can cause additional deterioration in

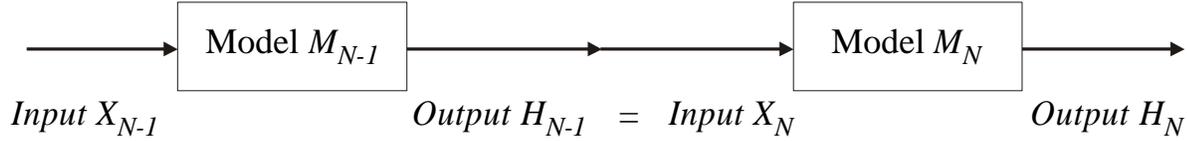


Figure 3.2.1-2 A sequential data flow path

The higher model M_N uses the output H_{N-1} of the lower model M_{N-1} as its input state, X_N . The accuracy of the output from model M_N is dependent on both the accuracy of input X_N and the model's sensitivity to the input error of X_N . Model sensitivity error is a concept that defines how error variations in input data affect the accuracy capabilities of the model's output data. Because each model has an error, each sequential model will compound the error as data flows from one model to the next. Top level fidelity is therefore dependent on a path that is opposite to the direction in which the data flows. Consequently, the accuracy requirements of these high level models define the accuracy requirements of all other models that are along the path to provide input to the top models. To maintain accuracy or fidelity, the recipient of data must receive data that is within the maximum error bounds determined for each of the models. By moving top-down, one can assign accuracy requirements for each model along the path depending on the expected error and the model's sensitivity to errors. Working bottom-up, the top layer accuracy or fidelity can be determined and compared to the required fidelity. To achieve the required level of fidelity for a federation, fidelity must be maintained along each data paths during execution. If the top layer models are always outputting data within their maximum error bounds, the achieved level of fidelity will meet the required level of fidelity. This means that the achieved level of accuracy of the top-layer models remains between the maximum allowable error bounds as defined by the required fidelity. The premise of this approach is that the occurring model errors and error sensitivities are known or at least that they can be estimated in some manner.

accuracy, the error can be compounded as it passes through all the models along a data flow path. A data flow path is a concept describing a path along which models exchange data with each other, in terms of model data input and output. The next figure shows a sequential data flow path.

3.2.2 Fidelity Metrics

For both the simulation development abstraction layers and the federation model layers it is possible to quantify fidelity in a mathematical manner, in terms of accuracy.

3.2.2.1 Accuracy Metrics for Simulation Abstractions

A mathematical expression of fidelity for the abstraction layers is obtained by calculating the difference between these layers. For generality all three errors are multidimensional and may be assessed over a domain of states, events and time. The fidelity or accuracy can be assessed from the error by taking the norm of the errors. The norm of the error between the conceptual model and the reality representation determines the fidelity of the conceptual model:

$$\mathbf{e}_{gf} = \|E_{GF}\| = \|G - F\| \quad (1)$$

Similarly, the error measure between the conceptual model and the simulation implementation is:

$$\mathbf{e}_{hg} = \|E_{HG}\| = \|H - G\| \quad (2)$$

Finally, the norm of the error between the simulation implementation and the reality determines the fidelity of the simulation implementation:

$$\mathbf{e}_{hf} = \|E_{HF}\| = \|H - F\| \quad (3)$$

To find the upper bound on the simulation implementation, equation (3) can be written as follows using equations (1) and (2):

$$\|E_{HF}\| = \|E_{HG} + E_{GF}\|$$

$$\mathbf{e}_{hf}^2 = \mathbf{e}_{hg} + \mathbf{e}_{gf}$$
(4)

Equation (4) shows that the error of a simulation is bounded by the error of the conceptual model and the error increase due to implementation. Federation fidelity is defined here as the inverse of the simulation implementation error norm:

$$Fidelity_{federation} = \|E_{HF}\|^{-1}$$
(5)

3.2.2.2 Accuracy Metrics for Compounding Errors

There are two types of errors to be considered to calculate the model error. The first types of errors are the errors associated with the algorithm, A, used in the simulation itself. Assuming one has perfect data, the algorithmic error, EAF, describes the variations from ground truth of reality one expects as of using the algorithm. The second type of error is the previously described model sensitivity error, EMS. The model sensitivity error determines how the model error is magnified by an error in internal and external model variables and parameters. Stated differently, the error caused by not using perfect data in the model algorithm. The difference or error between a model, M, compared to reality, F, can Thus be expressed as follows:

$$E_{MF} = M - F = (A + E_{MS}) - F = (A - F) + E_{MS}$$

$$E_{MF} = E_{AF} + E_{MS}$$
(6)

The algorithm is described here as a continuous function of the internal and external state variables xi :

$$A = f(x_1, x_2, \dots, x_n) = f(\bar{x})$$
(7)

Next the model sensitivity error can be represented by the variation of A with respect to some operational set of states \bar{x}_0 :

$$E_{MS} = dA|_{\bar{x}=\bar{x}_0} = \frac{\partial A}{\partial \bar{x}}|_{\bar{x}=\bar{x}_0} d\bar{x}$$
(8)

where the partial derivative matrix of A with each state variable is the Jacobian of A. This Jacobian describes the model's sensitivity to data input errors. In practice and in case a model is used without continues derivatives the model sensitivity error is approximated as follows:

$$E_{MS} \approx \Delta A|_{\bar{x}=\bar{x}_0} = \frac{\partial A}{\partial \bar{x}}|_{\bar{x}=\bar{x}_0} \Delta \bar{x}$$
(9)

Using equations (6 and 9) the total model error of the Nth model in a data flow path can now be approximated by:

$$(E_{MF})_N \approx (E_{AF})_N + \frac{\partial A}{\partial \bar{x}}|_{\bar{x}=\bar{x}_0} \Delta \bar{x}_N$$
(10)

3.2.2.3 Sample Application(s)

To demonstrate how error compounds, the sequence model of <<figure 0-2>> is used. The output of model MN-1 feeds the input of model MN-1. This means,

$$\Delta \bar{x}_N = (E_{MF})_{N-1}$$
(11)

Substituting the output error of the N-1th model for the Nth model input and using relationship (10) to describe the error of the models gives the following expression for the error of the Nth model along a data flow path:

$$(E_{MF})_N \approx (E_{AF})_N + \frac{\partial A_N}{\partial \bar{x}} \left(\frac{\partial A_{N-1}}{\partial \bar{x}} \Delta \bar{x}_{N-1} + (E_{AF})_{N-1} \right)$$
(12)

This expression of error shows three types of terms. The first term is the algorithmic error of the Nth model. The second and third term are the compounding errors caused by respectively the system data errors and algorithmic errors of prior models. Even if all models have no algorithmic errors, the data errors will creep into the system and will be modified by the model's sensitivity to input errors. The modifications to errors are a function of the norm of the algorithm Jacobian matrix. If the norm of the Jacobian is less than one, the model diminish errors, otherwise errors can grow. Another factor to consider is that recipients of data usually receive data from more than one source. These multiple source dependencies create a tree structure of data flow paths. The recipient model error analysis must be accommodated for all these error inputs in \bar{x} . The fidelity of the Nth model can now be calculate from expressions (5) and (12):

$$Fidelity_{Nth Model} = \left\| (E_{MF})_N \right\|^{-1} \quad (13)$$

3.3 Sources of Uncertainty

3.3.1 Conceptual Framework

Simulation fidelity descriptions assume a construct tacitly, even if they do not describe such explicitly. This framework presents a context which should accommodate a wide variety of such constructs (and possibly may even accommodate all of them). This context for simulation fidelity, illustrated by Figure 3.3.1-1, will facilitate comparison of underlying theoretical structures as well as comparisons on the contexts for descriptions of simulation fidelity. This context is also the logical basis for identification of simulation fidelity issues in the next section. The material below identifies the elements of this context and discusses critical considerations related to them. Note: Figure 3.3.1-1 addresses both modeling and simulation (M&S). This discussion focuses upon simulation.

The “Real World.” Simulation development starts with the “real world,” but the *real world* contains imagined reality as well as material reality. *Material reality* is defined as the material universe (or those parts of it) which are pertinent to the application domain. *Imagined reality* is a concept that has no specific counterpart in the material universe, such as a unicorn. For example, a horse would be part of the material reality -- a unicorn is imagined reality. Note that parts of the concept may have counterparts in the material universe (as in the case of the unicorn, only the horn is not part of material reality). It should be noted that while imagined reality may overlap material reality, imagined reality cannot be contained completely within material reality (otherwise, the imagined reality would merely be part of the material reality). Some may quibble with inclusion of imagined reality in the “real world,” but that inclusion is simply a appropriate convenience, since elements of an imagined reality frequently become part of material reality. Consider for example the simulation of a **Perceived Reality**. Perceived reality may not be identical with the real world because of limited and imperfect observations of material reality and because of articulation limitations related to imagined reality. Practically speaking this perceived reality is the ultimate referent for simulation fidelity when simulation results are compared with it. This perceived reality can also be a basis for standardized data and descriptions, whether these data are contained in data bases such as the master environmental data base or these descriptions are encapsulated in Conceptual Models of the Mission Space (CMMSs). Simulation Object Models (SOMs) and Federation Object Models (FOMs) may also be

contained in the standardized data and descriptions. Perceived reality is also the basis for description of an application domain. Both perceived reality and standard data and descriptions influence the application domain. Perceived material reality forms the basis for both standardized databases and descriptions as well as the referent for simulation results. A *referent* is the standard for comparison. The three referents identified in Figure 3.3.1-2 are standards like the meter-long bar maintained in Paris as the reference for the length specified as a meter-planned weapon system.

- **Application Domain.** The application domain provides a context for simulation requirements that lead to a simulation development. The application domain is derived from the perceptions and descriptions of (imagined and material) reality and is influenced by standardized databases and descriptions – such as would be contained in the DMSO CMMS vision. The application domain defines the fidelity referent.
- **Fidelity Referent.** The fidelity referent is the standard against which simulation fidelity is measured, both in conceptual validation -- which addresses the simulation concept -- and in results validation -- which addresses performance of the simulation implementation. . In this perspective, simulation fidelity is an absolute – it provides an objective measure of how close the simulation comes to the perception of reality described by the application domain.
- **Simulation Requirements.** Simulation requirements specify characteristics that should be possessed by a particular simulation. These provide guidance for simulation development. Simulation requirements are related to the intended simulation application domain. A subset of these requirements (or derived parameters) forms the acceptability criteria for the simulation.
- **Acceptability Criteria.** Acceptability criteria are specific test points for simulation validation evaluations. Conceptual models provide the linkage between simulation requirements and specifications. The conceptual models coupled with the acceptability criteria define the validation referent. Validation is a relative concept, being application dependent – in contrast to fidelity which is an objective measurement (i. e., fidelity is not application dependent).
- **Validation Referent.** The validation referent is the basis for judging whether a simulation development satisfies its intended function. It should be used for

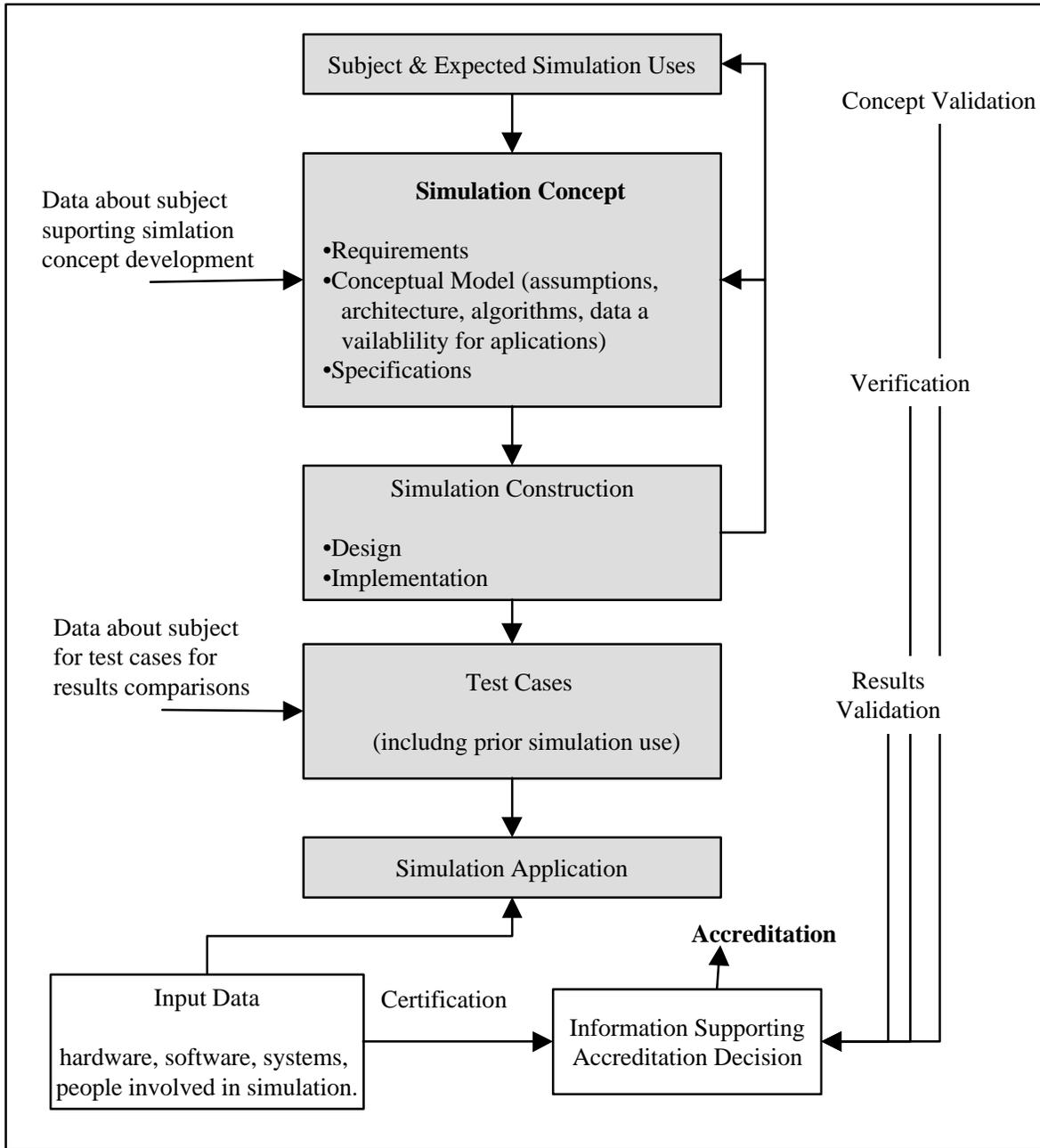


Figure 3.3.1-1: Categories of Fidelity and Reduction Processes

both conceptual validation review and for comparison with simulation performance in results validation.

Results from simulation test cases and from use of the simulation can be used for comparison with the three (validation, fidelity, and simulation results) referents.

It should be noted that this context for simulation development does not promote a particular modeling and

simulation theory. In fact, this context should be compatible with the variety of such theories that have been articulated to date. For example, this context above does not specify how one organizes a description of reality's abstractions, which is an important consideration for theories that derive from Casti [5]. Nor does the context above identify how one approaches the frame of reference used with the simulation, whether the Klir-based system science approach [5A] that Zeigler [28] and others prefer, or some alternative approach.

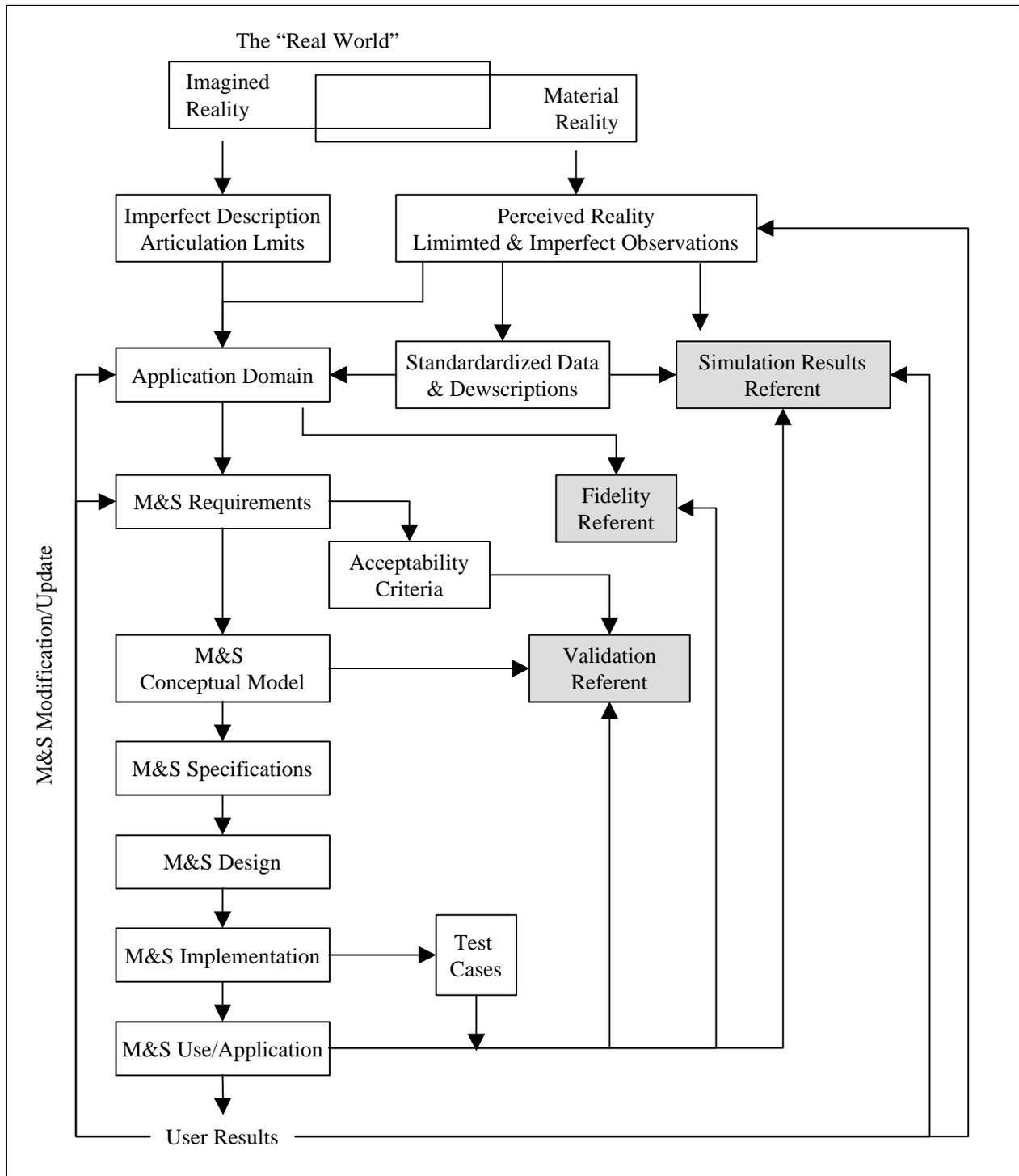


Figure 3.3.1-2: A Fidelity cONSTRUCT

Nor does the context identified here address the details of measurement that lies at the heart of statistical approaches to simulation theory, such as espoused by Oberkamp et al [23] or by Morrison and McKay [22].

Basic elements needed for a theory of simulation development have been identified [29], but such a theory of simulation has not yet been fully articulated.

3.3.1.1 Key Issues

Determination and specification of referents used is a key issue for simulation fidelity. The context provided by Figure 3.3.1-1 and the discussion above should facilitate more meaningful interchanges about this issue since use of this context provides a way to relate one approach to another approach more directly than would be possible without such a context.

Descriptive formats used for elements related to simulation fidelity is an important issue. The simulation items for which there may be different descriptive formats include:

- 1) real world,
- 2) perceived reality,
- 3) application domain,
- 4) standardized data and descriptions,
- 5) requirements,
- 6) acceptability criteria,
- 7) conceptual model(s),
- 8) specifications,
- 9) design,
- 10) implementation,
- 11) results,
- 12) review reports (including VV&A products), and
- 13) referents (real world, fidelity, validation).

Every transformation from one descriptive format to another introduces the potential for additional errors in the simulation. The current conclusion of the group within one of the International Standards Organization Technical Committees addressing semantic unification of static models is:

“there are no universally recognized techniques for unambiguously translating a model from one formal language to another, or even for determining whether two models in different languages were making equivalent statements”
[17]

The practicality of simulation fidelity descriptions and computation is a pragmatic simulation fidelity issue that will depend upon the fidelity construct employed.

This framework takes a systems engineering approach to a construct for simulation fidelity. This construct decomposes simulation fidelity into dimensions and attributes. **Dimensions** of simulation fidelity are concerned with the portion of pertinent entities, factors, and relationships represented within the simulation. In other words, dimensions of simulation fidelity address the extent to which pertinent elements of the problem space (subject domain, mission space, etc. – choose a term) are represented within the simulation. **Warning:** this definition is not necessarily consistent with the Fidelity glossary. **Attributes** of simulation fidelity, on the other hand, are concerned with the quality of parameter treatment within the dimensions of simulation fidelity and address such characteristics as accuracy, precision, timeliness (especially in distributed simulation), potential error sources, consistency, and repeatability.

3.3.1.2 Dimensions of Simulation Fidelity

Dimensions of simulation fidelity are concerned with the portion of pertinent entities, factors, and relationships represented within the simulation. *In order to pursue this concept of fidelity dimensions, there must be an articulation of the characteristics of the reality which the simulation represents.* For many simulations (probably for most simulations), no formal (i.e., comprehensive and documented) description of the characteristics of the reality exists. This kind of formal description defines the application domain of the simulation development construct presented in the previous section.

Articulation of the reality to be represented by the simulation (i. e., description of Figure 2’s application domain) begins with an enumeration of the critical aspects of the problem being addressed. Who prepares this enumeration? Normally enumeration of the critical aspects of the problem will be done by the one attempting to assess the fidelity of the simulation if an accepted enumeration does not exist within the community. If that party (person/team/agency/etc.) is different than the one that developed the simulation, the aspects identified for enumeration may not be the same as those used to design the simulation. The upshot of this situation is that the dimensions of simulation fidelity will be potentially in the eye of the beholder, at least until the community develops standards for how significant aspects of a problem are identified (and described) an accepted enumeration of the significant aspects of the application domain.

Comprehensive articulation of critical aspects of the application domain (“mission space” in Defense terminology) is the foundation for identifying the dimensions of simulation fidelity. For the remainder of this discussion, assume that such an articulation exists or that one can be created as the basis for describing simulation fidelity. If the subject is not well enough understood to support such an articulation, then discussion of simulation fidelity can have little or no meaning as for technical usefulness in determining a simulation’s appropriateness for an intended application or suitability for combined use with other simulations.

Enumeration of the entities involved is the first dimension of simulation fidelity. The entities must be addressed in both scope and depth:

- **Scope** addresses the spectrum of entities represented by the simulation. For example, if fluid flow in a pump is the subject of the simulation, the scope considers the entities represented in the simulation. Such entities include input and output pipes, any chambers in the pump, pistons or other moving parts, the fluid flowing through the pump, etc. The scope dimension of simulation fidelity can be described as a number (m entity types of n possible types represented) or as a percentage (x% of entity types are represented).
- **Depth** addresses the level at which entities represented in a simulation can be individually (and distinctly) identified. This quantification of the simulation can extend in either direction: aggregation or de-aggregation. Simulation depth in this sense is sometimes called the “level of detail” or “resolution.” Again, the pump simulation will be used to illustrate this fidelity dimension. Entities may be represented at a component level, at parts of the components, at molecular levels, at atomic levels, etc. Thus far, we have addressed depth as it moved down from some level of representation to more elemental levels of the physical entities. However, depth also extends in the other direction. Some entities of the pump simulation, such as the control mechanisms for pumping speed, may be represented in an aggregated fashion with a single algorithm combining inputs of several sensors and control processes. The depth dimension of simulation fidelity can be described as a number (m of n, relative to the number of levels from the lowest to highest possible entity consideration) or as a percentage (x% of entity levels are represented).

The second dimension of simulation fidelity is **identification** of factors involved. The factors involved in a simulation relate to processes that influence, impact,

or describe (characterize) entity states and behavior. Thus, factors include such considerations as components and materials, parameters internal to entities that effect behavior, algorithms that define possible entity states and behavior, and parameters related to measures of performance/effectiveness/merit (MOPs/ MOEs/ MOMs). As in the enumeration dimension of simulation fidelity, this dimension of fidelity is either a number of the possibilities (n of m factors represented) or a percentage of those factors.

The third dimension of simulation fidelity is **specification** of the significant relationships among entities involved. For simplicity in simulations (or because it’s easier), or in some cases because of ignorance, too often independence is erroneously assumed to characterize relationships among entities and factors (including processes within an entity when such interact). At the very least, such assumptions about dependencies or independence should be made explicit and not left in the tacit domain. Such specification of relationships can describe them as independent, dependent of a specific sort, or otherwise – the *otherwise* description is a catchall for dependent relationships which can not be described more precisely.

Technically meaningful fidelity descriptions help one to decide if a particular simulation is appropriate for a particular application or suitable for use with another simulation. The enumeration and identification dimensions of fidelity can not be assessed with articulation of the application domain – this is the starting point for a community that has serious concern about simulation fidelity: development of the fidelity taxonomy that allows articulation of the application domain.

3.3.1.3 Attributes of Simulation Fidelity

Attributes of simulation fidelity are concerned with the *quality* of parameter treatment within the dimensions of simulation fidelity and address such characteristics as factor order, accuracy, precision, timeliness (especially in distributed simulation), potential error sources, consistency, and repeatability.

Factor Order. The quality of parameter treatment within a simulation begins with the order of the parameter’s description for a factor in the simulation. Order is similar to a degree of freedom (DOF) as that term is used about an entity’s spatial location and orientation (attitude). Another example of factor order is treatment of the radar cross-section (RCS) or other kind of signature of an entity within a simulation. The zero order description of this parameter would be a constant value, the same from any aspect angle and for all frequencies. A first order description would make the

RCS vary in one way (stochastically according to some statistical distribution, by frequency with different values for different frequencies, by aspect, etc.). A second order description of RCS would make RCS vary in two ways. A third order in three ways, and so on. In addition to the three ways for RCS variation indicated already, RCS can also vary with whether the signal is treated monostatically or bi-statically, with the waveform of the radar signal, etc. The importance of the order of treatment of this parameter depends upon the intended simulation application. In comparison of simulations to determine their suitability for combined use in a distributed simulation, different orders for treatment of a particular factor should be an easy way to identify factors which need careful consideration.

In general, the higher the order, the greater the fidelity. A simulation can have a high order for some factors (such as entity location), but a low order for other factors (such as entity RCS) that will cause simulation fidelity to be “high” or “low”, depending upon the application. This is why simulation fidelity must be addressed in terms of factors (decomposition of the problem), not simplistically in some aggregated fashion.

Accuracy, precision (which is functionally equivalent to resolution and granularity), and timeliness are characteristics that describe how close the representation of an individual parameter is to reality. **Warning:** this definition is not necessarily consistent with the Fidelity glossary. Accuracy is always limited by precision. Precision, the level of resolution or granularity with which a parameter can be determined, places fundamental limits on accuracy. As is well known, one can be very precise but inaccurate, but one can not be very accurate and very imprecise at the same time. Accuracy is determined by how well simulation algorithms represent the subject simulated. Accuracy can be measured against reality (when such material reality data exists) or against the articulation of the application domain (mission space) for the simulation. In the first of these two possibilities, accuracy is a function both of the correctness of the abstraction and of the simulation’s representation. In the second possibility, accuracy is only a function of the simulation’s representation of the abstraction. It is possible to distinguish these two kinds of accuracy with descriptive labels: *real accuracy* and *abstraction accuracy*. Accuracy has to relate to a single parameter or set of parameters (accuracy can be construed as a multi-dimensional vector with a dimension for each parameter in the set). Simulation accuracy is difficult to determine if material reality data are sparse or do not exist.

Precision, on the other hand, is much more amenable to quantification than accuracy. **Warning:** this definition is

not necessarily consistent with the Fidelity glossary. Typically precision (resolution, granularity) in a simulation can be determined by examination of computational processes used in the simulation (number of significant digits, round-off procedures, interpolation intervals, minimum step sizes, update/refresh rates, number of pixels for displays, etc.), and quality of the real world data. However, in hardware in the loop (HWIL) simulation, precision of the simulation may also be limited by facility considerations -- such as how well alignment of physical elements of the simulation (such as antennas and receivers) can be maintained. Such alignment may also be dependent upon other physical aspects of the facility, such as control of temperature and humidity in an anechoic chamber used in the simulation.

Timeliness must be given special attention in distributed simulations, in unitary simulations employing distributed processing, in discrete event simulations, and in other kinds of simulations in which some parts of the simulation may advance more rapidly or more slowly than other parts. Manifestation of timeliness impact on simulation fidelity depends upon how time is managed in the simulation: continuous, time step, discrete event with complex roll-back capabilities, etc. In the most simple situations, a parameter update may be missed for a time cycle or two in a particular simulation implementation – because it took too long to compute the update, because there was a delay in communications between parts of a distributed simulation, etc. The maximum magnitude of error between the parameter value with and without the missed updates quantifies the potential impact of timeliness on that parameter. However time is managed (simplistically or in a sophisticated manner in such a simulation), timeliness issues create fuzziness for parameter accuracy and precision which must be addressed in considering simulation fidelity.

Errors. Pace, Oberkampf et al, and others have identified a variety of simulation **errors** that must be considered in a comprehensive and workable treatment of simulation fidelity. Simulation errors include:

- Imperfect observation/measurement of input data,
- Deviation from correct input data,
- Less than perfect algorithms for description of entity state, behavior, and performance, and
- Finite limitations in computational and logical processes.

It is possible in some cases to define simulation accuracy in terms of such errors. Sometimes it is possible to identify and quantify some errors even when total

simulation accuracy can not be determined completely because other errors have not proven amenable to quantification. For example, errors introduced by interpolation between values in a table lookup process can be quantified rather easily even if the errors in the values of the table itself can not be quantified. Thus, errors provide a partial way to address accuracy when accuracy of a parameter can not be fully determined.

When such errors are independent of one another, standard statistical processes may be employed to estimate their combined impact. Foster has described the way that independent errors (inaccuracy) propagate through a distributed simulation. When errors are not independent, it becomes very difficult to estimate the impact of error combinations since the combined error may be greater or less than individual errors – and in many situations, simulation errors are not independent.

Consistency is also an attribute of simulation fidelity. It addresses whether simulation results are biased (consistent error direction) and stable in terms of the dispersion of results induced by simulation processes. Quantification of some consistency parameters can be estimated by test cases that use boundary conditions values (such as values of 0 or 1 for probabilities within the simulation) and by sensitivity analysis.

Repeatability is a simulation fidelity attribute that many assume whether it has been demonstrated or not. Repeatability simply means that the simulation should produce the same results/responses given the same stimuli (inputs, decisions, operator actions, etc.). This requires potential for complete control of stochastic processes within a simulation (such as pseudo-random seed draws) if repeatability is to be ensured. Dewar et al document a case of major changes in simulation results caused simply by running the same simulation with the same inputs on two different computers. Likewise, variation in communication delays among parts of a distributed simulation can make it impossible to replicate simulation results exactly at times. Quantification of such potential variability in results is an important aspect of simulation fidelity.

People Issues. There are special fidelity concerns when a simulation involves people. This is true whether the people are operators, “players”, or analysts. The key is that the people can impact simulation results. Pace [1998a] identified a variety of validation issues that must be addressed when people are involved in a simulation; there are also a number of simulation fidelity considerations. People impact accuracy, precision, consistency, and repeatability – both as stochastic processes of the reality represented in the simulation and

as aspects of simulation implementation. Often this human aspect of simulation fidelity is totally ignored.

Ideally it would be desirable to develop a general simulation fidelity taxonomy top-down, but there is little likelihood that such will happen – at least it appears unlikely that such will happen in the near future. However, it may be possible for specific simulation communities to come to agreement about a taxonomy for simulation fidelity within their restricted application domain within a reasonable amount of time (and resources). This will allow them to make meaningful assessments of simulation fidelity, i.e., assessments which facilitate more objective determination of the technical appropriateness of simulation use for a particular application. When a number of simulation fidelity taxonomies exist, it will then be possible to determine if a generic simulation fidelity taxonomy can be developed. This bottom-up approach is the only one which appears viable at present.

3.3.2 Fidelity Metrics

There are many simulation fidelity issues. This section will only address issues directly related to CFD simulations and computational mechanics codes. It will not include issues that are outside this domain. This discussion will take the perspective that the taxonomy for simulation fidelity is a matter of concern to a particular group, that group consists of the simulation developer(s) and user(s) – both defined in the broadest sense possible. The “developer(s)” include the one(s) providing authorization to create the simulation, the one(s) funding its development, and the ones who actually design and build the simulation. The “user(s)” include those who operate the simulation, those who analyze its responses, and those who use (make decisions, etc.) information derived from simulation results. Possibly there may be others who are also concerned with the fidelity of the simulation, but the developer(s) and user(s) are the primary players. They are the ones who must develop a consensus about simulation fidelity.

The first issue is *articulation* of the referent to be used as the basis for comparison (the “standard”) to evaluate simulation fidelity. Figure 2 has two possibilities: the “fidelity referent” and the “simulation results referent.” Reasons were presented early for why the fidelity referent is essential (i. e., why the real world referent for simulation results is not the only thing needed). It is easy to map other paradigms to these referents, paradigms such as the “Proposed Phases for Computational Modeling and Simulation” suggested by Oberkampf et al. Which ever approach is chosen, it should be defined in an explicit and distinct fashion.

Lack of such an explicitly defined standard is a common problem in simulation fidelity. If a small community (such as the simulation's developers and users) can not come to agreement about (and articulate) such a standard, there is little likelihood of that community coming to agreement about the technical meaning of simulation fidelity. The Defense community has been working for several years on its approach to the Conceptual Model of the Mission Space (CMMS), which Gross et al proposed as a candidate for this kind of referent for simulation fidelity. CMMS insights should be of interest to those who have not progressed as far in how to describe an application domain. Other efforts to articulate application domain characteristics have not made as much progress in determining how such fidelity referents should be described. This issue concerns both what the content of the fidelity referent should be and how its content should be described.

The second issue in simulation fidelity is *identification* of the simulation fidelity taxonomy that should be employed: What dimensions and attributes of simulation fidelity should be specified? No standard way to identify or describe these yet exists. In one sense, this is good since it allows a particular community of simulation developers and users to create a structure that works for them exactly without having to include extraneous parts of a "standard" approach.

The HLA community has developed a series of templates that guide how an individual simulation (i. e., a federate) in a distributed simulation (i. e., a federation) is defined and described. These templates prescribe how the federate interacts with the services of the HLA environment provided by its Run-time Infrastructure (RTI) and with other federates in the federation. These templates play an essential role in making HLA work. Comparable templates are required for articulation of the dimensions and attributes of fidelity referents. As narrowly focused application communities within the larger simulation world develop, test, and improve such templates for their particular kinds of applications, a collection of usable templates for fidelity referents can come into existence which can become the basis for generalization about templates suitable for fidelity referents. Proven and widely accepted application specific templates do not yet exist.

The third issue is how to *measure* the dimensions and attributes of simulation fidelity which have been identified. Some measurements will be easy. They will simply enumerate factors considered. Other measurements will be difficult – such as assessing the impact on a parameter or factor accuracy when errors may exist in data about that parameter as well as errors may exist in the algorithms related to the factor, and

more than one algorithm may be involved with complex relationships among the algorithms. No viable general methods currently exist for establishing parameter representation accuracy in such complex situations. Titles of papers for Caltech's December 1998 V&V Symposium imply that approaches to this issue will be discussed. An important issue is the availability of data with which to compare the simulation and its results. When there is an abundance of data (a data-rich environment), the normal techniques of measurement and comparison, with its use of standard statistical techniques, can be employed. When data are limited (a data-sparse environment), it may be necessary to use a variety of techniques to allow fidelity assessments to be made -- techniques such as combining expert opinion with data to create a larger amount of "data." In any case, it is desirable to try to generate as much data as possible so that fidelity assessments can be based as solidly in reality as possible.

A fourth issue is how to determine the level of fidelity *required* for a simulation application to be valid for its intended purpose. It is desirable to have a meaningful way to describe a simulation's fidelity, but if the fidelity required by the intended application of the simulation can not be specified, then it will be impossible to answer the validity question. As simulations are used in development of systems with stringent performance requirements and in support of safety-critical systems, ability to articulate fidelity requirements as well as estimate and measure simulation fidelity will become increasingly important.

3.3.3 Application

There are not yet published applications of this fidelity framework.

3.4 Fidelity Differentials Framework

This work is elaborated in references [12] and [13].

We need the ability to quantify simulation "goodness", but there is no generally accepted method for doing so. This is evidenced in the lack of agreement on even a definition for fidelity. A variety of concepts [3] have been advanced such as:

- 1) The similarity, both physical and functional, between the simulation and the referent.
- 2) A measure of the realism of a simulation.
- 3) The degree to which the representation within a simulation is similar to a real world object, feature, or condition in a measurable or perceivable manner.

These ideas might be captured as the following:

Fidelity is the extent to which the model reproduces the referent, along one or more aspects of interest. **Warning:** this definitions are not necessarily consistent with the current Fidelity Glossary.

This suggests that the crucial issue in fidelity is the differential between a simulation model and its referent. The framework outlined here provides a way that might be used to understand and measure that differential.

3.4.1 Conceptual Framework

Any consideration of the fidelity problem immediately suggests two problems: (1) how to define the referent, and (2) how to measure the difference between the referent and the simulation. Leaving aside the first problem for the moment, we note that the definition itself provides some assistance to measuring differences. Namely, it recognizes that there are different aspects, or dimensions to fidelity – which immediately suggests that differences in some aspects may be measured differently than others.

Many authors have recognized this attribute of fidelity. For example, Schow et al, suggest fidelity dimensions including: “attributes, and interactions of the objects to be inter-operating; and simulation components: local and global (visual displays, computer system, local area and wide area networks)” [3]. This references also notes that “each of the ... dimensions are ... measurements subject to differences in the subjects”. This is self-evidently true, but what are the aspects or dimensions of fidelity? Different sources have suggested various lists, such as [3]:

- 1) Visual system fidelity
- 2) Synthetic environment fidelity
- 3) Physical hardware
- 4) Interactions
- 5) Dynamics

While the specific of the different fidelity aspects vary from subject to subject, we argue that any aspect of fidelity can be classified as one of three kinds:

- 1) Existence
- 2) Attributes
- 3) Behavior

Figure 3.4.1-1 illustrates these three aspects of fidelity categories, and the process by which the referent is reduced to the model. It may be comforting to note that these three categories parallel closely the theoretical foundation for object oriented design.

It may be useful to think through the implications of this figure. First, for entities which *exist* in the referent, the model is built through a process of *aggregation*, Thus reducing the model’s fidelity in this category. For example, a battlefield model may aggregate individual soldiers into a platoon.

Second, for *attributes* in the referent, the model is built through a process of *clarification*, Thus reducing the model’s fidelity in this category. For example, a battlefield model may clarify by eliminating irrelevant attributes such as commissioning date.

Finally, for *behavior* in the referent, the model is built through a process of *simplification*, Thus reducing the model’s fidelity in this category. For example, a battlefield model may simply by reducing the details of a platoon’s marching behaviors to simple movement.

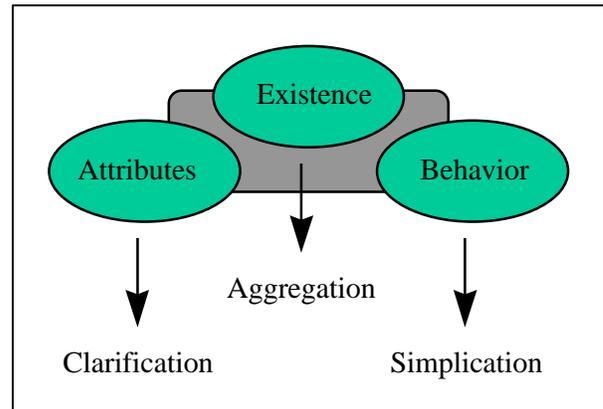


Figure 3.4.1-1: Categories of Fidelity

Within this framework, fidelity is a characteristic function of simulation models which exhibits certain properties. We assert these properties as a set of four theorems.

Theorems:

I: $0 \leq F(A) \leq 1$

II: *if* $F(A) = 1$ *then* $A \equiv R$

III: $F(A) \geq F(MetaA)$

$$\text{IV: } F(A) + F(B) = \min(F(A), F(B))$$

where:

- 1) A , and B are models of interest.
- 2) $F(A)$ is the fidelity of model A .
- 3) $MetaA$ is a model of a referent including A .
- 4) R is the referent of model A .

Figure 3.4.1-2: Properties of Fidelity

Theorem I defines the mathematical range of fidelity is 0 to 1 – note the similar to probability. As an aside, we note that in this context a higher fidelity value is “better” than a lower value. Theorem II states that if model has perfect fidelity, then it is no longer a model but is the referent, for the aspect being measuring. Theorem III states that the fidelity an entity incorporating another entity, is limited by what it incorporated. Theorem IV states that the fidelity of two objects working in corporation is limited by the lower fidelity one. Therefore, the fidelity of the combination can not be greater than the fidelity of the individual incorporated models. Theorem IV has the most serious implications for interoperable simulations.

We should clarify our earlier paper by pointing out that this means we see fidelity is a n-tuple, where $n = 3$. By this, we mean that there is no meaningful way to combine these three different aspects of fidelity into a single measure. In our approach, we measure fidelity along each of these aspects, and report the three scores as “the” model fidelity. Of course, with it a specific aspect (such as attributes), it is possible to make a composite number representing the fidelity of the system. Therefore, the fidelity of a model $F(A)$ means:

$$F(A) = 1 - f(\text{Aggregation, Simplification, Clarification})$$

where $f(x)$ is a function that measures the degree to which model “ A ” is aggregated, simplified, and clarified from reality. We might call $f(x)$ the “infidelity” function. We see from this that it may be easier to measure fidelity indirectly (i.e., as the complement of infidelity) than directly

The foregoing discussion is a useful theoretical foundation for fidelity, however it does not address the first of the two problems raised in our definition: how to define the referent. This problem has been the prime deterrent to creating useful definitions of fidelity. Without a useful definition of the referent, it is not

possible to precisely measure fidelity. However, as soon as we define the referent, we have created a new model with its own fidelity problems. This vicious circle has meant that fidelity remained an abstract concept rather than a usable measure of simulations.

However, the High Level Architecture (HLA) offers a way to break out of this vicious circle through its differentiation between the Conceptual Model of the Mission Space (CMMS), Federation Object Model (FOM), and Simulation Object Model (SOM). The SOM defines what an actual simulation can do; the FOM defines what is desired. FOMs are refined by comparison with the SOMs in a process fondly known as a “FOM-o-rama”. Prior to the FOM-o-rama, the FOM represents a requirement specification, afterwards it becomes the design for the simulation. The SOMs represent an implementation.

The CMMS by contrast, although it is not a simulation element, defines the *context* within the FOMs and SOMs create a simulation for exercise. The CMMS is DMSO’s response to DoD Modeling and Simulation Master Plan Sub-Objective 1-2, “Develop a conceptual model of the mission space for each DoD mission area to provide a common basis for development of consistent and authoritative M&S representations.” As such, the CMMS should not be a simulation – instead, it is focused on describing the battlespace of real mission areas, such as military operations, operations other than war, training, etc. We believe that one way the CMMS could fulfill its objective is by including sufficient information to serve as a referent for fidelity measures. There will be several CMMS corresponding to broad mission areas such as conventional combat operations, other military operations, training, acquisition and analysis). The mission space structure, tools and resources will provide both an overarching framework and access to the necessary data and detail to permit development of consistent, interoperable, and authoritative representations of the environment, systems, and human behavior in DoD simulation systems. [11]

Therefore, the CMMS becomes a usable definition of the referent. The CMMS along with a theory of fidelity such as presented enable us to make meaningful use of fidelity for the first time.

Some authors have pointed out that the CMMS is a model in and of itself, and therefore it too has a fidelity [12]. This argument leads to a dead-end. Any attempt to define reality is inevitability a model, and reduces fidelity to a circular argument. Furthermore, it is not meaningful to discuss the fidelity of the CMMS for the following reasons:

- 1) CMMS “is” reality *in the context of HLA*, and violating this assumption unhinges the fidelity measure,
- 2) Since they are different in form, fit, and function, there is no direct comparison of the CMMS and the FOM,
- 3) By definition, Fidelity is not a consideration in building the CMMS [27], the only consideration is capturing reality
- 4) Finally, arguing that F(CMMS) is significant is really arguing that there is a 1:1 ratio between the CMMS and FOMs, which it is clearly not the case.

Therefore, we assert that for HLA simulations, fidelity is the difference between the CMMS and FOMs/SOMs.

We believe that HLA offers a way around these obstacles through the Conceptual Model of the Mission Space (CMMS), the Federation Object Model (FOM), and Simulation Object Model (SOM). We understand that this position is controversial. We simply argue that the CMMS is our best hope in finding a universal referent descriptor, and that attempts to use fidelity will inevitably flounder unless some such descriptor is found.

FOMs and SOMs provide definitions of simulation models. If we can arrive at a consensus as to a referent, via the CMMS or some other approach, then one measure of fidelity is the difference between the FOM or SOM and the referent. Since FOM development precedes searches for existing SOMs, or development of new SOMs in most Federation Development and Execution lifecycles, it will be useful to consider the fidelity of the Federation – the delta between the referent and the FOM – and the fidelity of the Federate – the delta between the referent and the SOM. Since FOMs and SOMs are described in the same formal notation, once we have determined a way to measure the delta between the reference and any object model, we have done it for both.

Not only would this give us a way to objectively measure fidelity, but it immediately suggests uses for the resulting fidelity measurement. For example, one might design a Federation, measure its fidelity from its FOM, and then seek suitable Federates to participate based among other factors on the fidelity of the proposed Federate from its SOM. In another example, such a fidelity measure help us verify that a Federate would be able to participate in a Federation on a level playing field by comparing fidelity levels.

Before leaving this topic, we should point out that HLA is not a panacea on the issue of fidelity. For example,

HLA offers nothing on the physical fidelity of human-in-the-loop simulators. Furthermore, the approach we have outlined does not look inside of the simulations to measure the fidelity of its internal models. However, we do believe that we have outlined a viable starting point.

3.4.2 Fidelity Metrics

We come now to the problem of measuring fidelity, specifically the difference between CMMS and FOMs/SOMs. It is necessary to define two additional terms:

*Accuracy is absolute maximum deviation allowed from referent to model. **Warning:** this definition is not necessarily consistent with the current Fidelity Glossary.*

*Tolerance is the relative accuracy. **Warning:** this definition is not necessarily consistent with the current Fidelity Glossary.*

In specifying fidelity, we will be occasionally interested in the accuracy, but more generally in the tolerance for the fidelity issue in question. The following addresses the various techniques available for measuring fidelity differentials.

3.4.2.1 Counting/Checking

The most obvious technique for measuring fidelity is brute force counting and checking of required attributes. This is relatively straightforward in HLA simulations, since it is certainly possible to, for example, confirm or deny the existence in the FOM or SOM every entity defined in the CMMS. The difficulty is that this is not a particularly powerful tool, since it considers only one fidelity category (existence).

3.4.2.2 Tolerance Bands

The most typical way of measuring fidelity is to compare the numeric output of a model against the same output for the referent. Figure 3.4.2-1 illustrates a data sample from a referent and a model.

Fidelity here would be defined as:

$$F_i = 1 - \left(\frac{|M_i - R_i|}{M_i} \right)$$

where:

F_i: Fidelity

M_i : Model Data Point

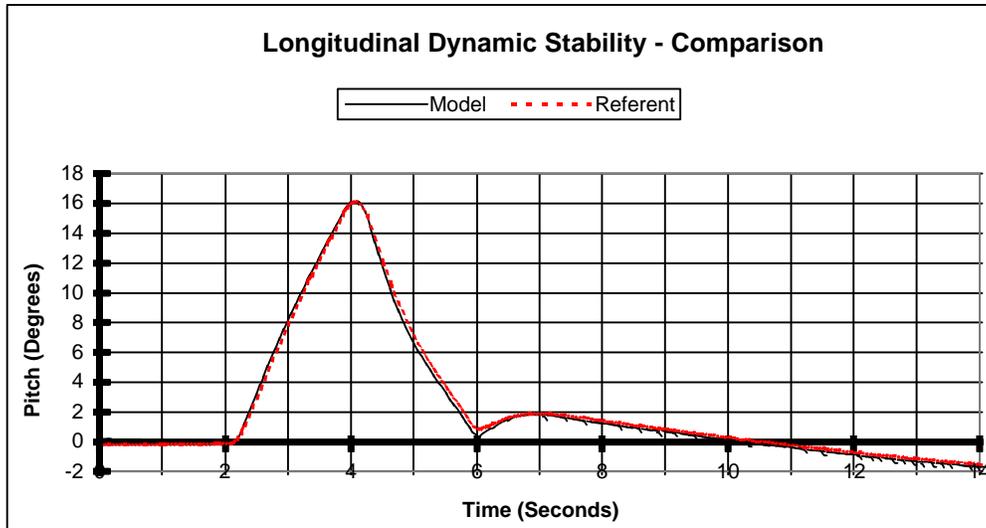


Figure 3.4.2-1: Sample Data for Fidelity Computations

R_i : Referent Data Point

One would verify the fidelity by comparing the resulting fidelity to one minus the required tolerance. Notice this measure conforms to the fidelity properties discussed before. This technique is widely used and powerful, however it has the significant limitation of requiring repetition for each aspect of fidelity. This technique is as useful for HLA simulations as for any simulation, but it is a very single dimensional view of the system, generally considering the behavior of a single attribute of a single entity.

3.4.2.3 Expert Opinion

A second method for measuring fidelity is expert opinion. Of course, this technique is more properly an estimate than a measurement unless more than one expert opinion is combined to produce the fidelity measure. This technique is widely used, despite its obvious limitations as to subjectivity and misinformation. Expert opinion is most useful in considering the integrated behavior of entities.

This technique is useful for HLA simulations, however as other authors have noted [27], the complexity of large numbers of objects interacting is generally beyond the comprehensive of any one mind.

3.4.2.4 Soft Computing

Given the important advances in artificial intelligence techniques loosely grouped under the heading “soft computing” (expert systems, fuzzy logic, neural networks), it would be possible to build a black box to

measure fidelity. Imagine feeding such a device a CMMS and a FOM, and creating a statistic, like a fuzzy set membership value, which states the degree of match between the CMMS and the FOM. Since the CMMS is captured in English, and the FOM is in a more structured template, this should be doable. This technique is more theoretic than practical, although it would be an interesting line of research. It offers a real chance to increase our ability to meaningfully measure fidelity.

Measurement of individual fidelity aspects continues to be a problem however these four techniques describe ways to deal with all three categories of fidelity aspects.

The real value of fidelity is in its use as verifying and validating the simulation. We use these terms here in the sense of DMSO’s Verification, Validation, & Accreditation document [10]. Namely, verification is the determination that the system was built right, and validation is the determination that the right system was built. In the context of this paper, it should be clear that the fidelity of the FOM is a validation question, and the fidelity of the SOMs are verification questions.

Measurement of individual fidelity aspects continues to be a problem, but just as important is the capability to combine fidelity measures into a single statement of fitness. Schow et al [27] laid the foundation of a fidelity differential calculus which is available technique describing system fidelity.

The total fidelity of a system may be represented as:

$$F_s = \sum F_i W_i$$

over the range $i = 1$ to n , where n is the total number of relevant system capabilities (i.e., visual system, user interface, etc.), F_i is the fidelity measure of each relevant characteristic and is a function of the simulation, $F_i = f(\text{simulation})$, and where W_i is a weighting factor based on the importance of this characteristic in the performance of a particular task, and is a function of task, $W_i = f(\text{task})$. As can be seen, the total fidelity of a simulator can and will change depending on the task or exercise at hand. Therefore, in general for two different tasks i and j we see that:

$$\sum F_i W_i \neq F_j W_j$$

Due to the differing requirements of each application, the fidelity metric of the same simulation for different applications will seldom be identical. [27]

The question arises, “Is it necessary that all fidelity aspects be measured in the same way?” No, only that they be done in a defined way the results conform to the fidelity properties detailed herein.

It is this “defined way” that requires addressing by the SISO community at large. The time is ripe for the community to put its collective heads together and “standardize” a method or methods to define and measure fidelity in an objective manner. The measurement and understanding of fidelity measurement is a critical issue for the various user communities and demands the attention of the SISO SAC.

Thus we have a viable method for combining the different fidelity measures into a single system fidelity. How can this be used? We assert that if the simulation specification includes a required fidelity level, then comparing the fidelity captured in the FOM to the SOMs, on a system by system basis is a direct *verification* of the system. *Validation* is considering the question of whether the abstract fidelity level captured in the FOMs is sufficient.

But how will we conduct this verification? To answer this question, we will borrow from regression analysis.

Regression analysis is a well known technique for developing a mathematical function describing a collection of points. Regression analysis is at the root of many behavior models embedded in simulations. Figure 3.4.2-2 illustrates a regression analysis.

Without discussing the mechanics involved, the process for regression analysis is first proposing a model equation for the function, and then calculating a statistic indicating whether the function is sufficient to describe

the data. If not, a new model is proposed and a new statistic calculated. Each model increases in complexity in terms of the number of dependent variables, the number of terms of each, or the powers of each.

The statistic for model sufficiency arises from a direct application of the Central Limit Theorem, namely that over a reasonably large sample the residuals (see Figure 3.4.2-2) for an adequate model will be normally distributed. Since some points will naturally be on one side of the model line, and some on the other, the statistic is the ratio of the mean sum of the squares of the residuals to the mean sum of the squares of the model. Recall by definition that ratio of two squared normal distributions follows the F continuous probability distribution. Then the computed statistic is compared to the theoretic F distribution to find the likelihood that the residuals are in fact normal. The equation for the computed statistic is:

$$F_0 = \frac{\sum MSM}{\sum MSR}$$

where:

MSM : Mean Sum for Model

MSR: Mean Sum of Squares

total degrees of freedom (DOF) = # of data points

Residuals DOF = number terms in the model

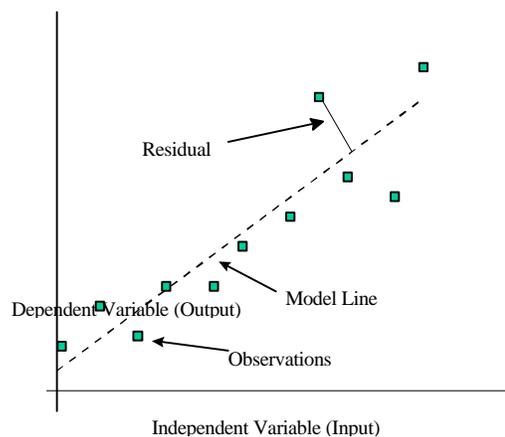


Figure 3.4.2-2: Regression Analysis

We can apply this same approach to verifying the fidelity of a simulation. Figure 3.4.2-3 illustrates a series of

fidelity aspects (1, 2, 3, ...), and the corresponding data points.

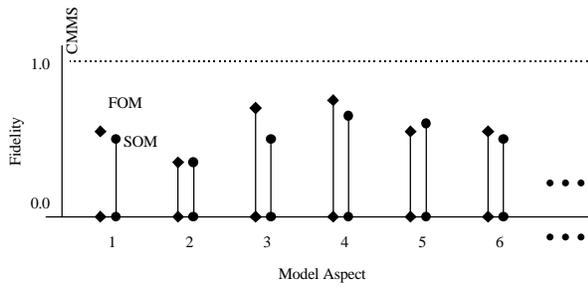


Figure 3.4.2-3: Series of Fidelity Observations

The limit of each fidelity aspect is 1.0, as defined in the CMMS. The FOM defines a fidelity requirement, which serves as the observations in a regression model. The SOMs define the model results, which aim to produce the fidelity required by the FOM. The deviation between the fidelity of the FOM and the fidelity of the SOMs becomes the residuals. By applying the same mechanics as in regression analysis, we can produce an objective statistical measure indicating whether the SOMs produce the fidelity specified by the FOMs or not. Per the earlier discussion, this statistic is based on a series of fidelity measures, as many as deemed suitable, measured in any appropriate way.

Making significant use of fidelity for the specification, design, implementation, verification, and validation of simulations, particularly interoperable simulations, requires adoption of a meaningful theoretical basis. There are presently two open significant holes in any such theory: how reality is defined, and how fidelity is measured. HLA provides a critical element of such a fidelity theory, namely a definition of reality. This true only if the CMMS is accepted as reality, as opposed to another model. In regards the measurement problem, the first critical step is to recognize that there are fundamental different aspects of fidelity, and they must therefore be measured in different ways. There are three classical approaches to measuring fidelity, and promise is offered by a fourth. There are mathematically sound approaches to combining all four into an objective measure of system and simulation fidelity.

3.4.3 Applications

We attempted to apply this framework to a practical problem. We set out confidently to acquire CMMS, FOMs, SOMs data sets, and immediately discovered how difficult it is to get real data!

First, we sought to acquire different CMMS to serve as our referent. However, we were discouraged from doing

this, and were unable to acquire any CMMS products to examine. It was suggested that while the CMMS describes the entities and tasks in a battlespace, it would provide no modeling inputs. Of course, we were not seeking modeling inputs. What we wanted to do, and still would like to do, is invent a way to compare the CMMS' descriptions against object models. Obviously, there is no direct way to make that comparison now. Figure 3.4.3-1 is an illustration from the CMMS presentations on the DMSO web page, that illustrates our point. What we are trying to do, is define another way in which the CMMS would be useful for front end analysis for simulation implementations.

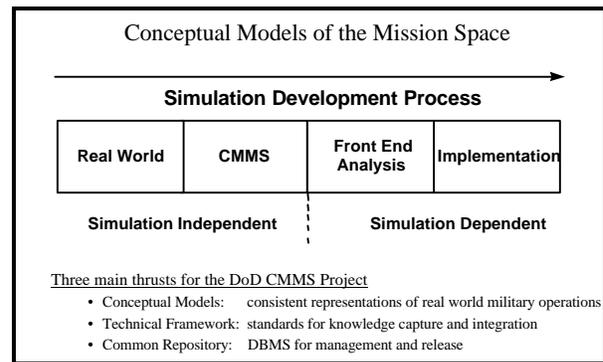


Figure 3.4.3-1: The CMMS Stated Intention[1]

Next, we sought FOMs and SOMs of different real world entities. Ideally, we would have liked to find several of each type for the same referent, in order to compare the different fidelities against each other. We thought that this would help us define our comparison mechanisms. Once again, we discovered how hard it is to acquire such data. We were unable to find multiple FOMs or SOMs for referent that we were sufficiently familiar to make judgments.

In the end we settled on a simplified case to work as a sample problem. We choose to use the Avenger Weapon System as our referent. We choose Avenger because we could access sufficient data: a) they are manufactured at Boeing-Huntsville; b) there are a number of different fidelity Avenger simulations in existence; and c) they are based on the ubiquitous High Mobility Multipurpose Wheeled Vehicle (HMMWV) and the well-known Stinger missile.

The Avenger consists of a turret mounted on a HMMWV. The turret has two gyro stabilized launch arms each holding a Standard Vehicle-Mounted Launcher (SVML). Each SVML contains four ready-to-fire Stinger missiles. The turret also contains a gunner's firing station, necessary electronics, controls, displays communications, etc. Under the left arm is mounted a

sensor suite. Under the right launch arm is mounted a 50 caliber machine gun. In the HMMWV cab is mounted a remote Control Unit (RCU) which permits operation of Avenger at a distance away from the HMMWV. Avenger is normally operated by a two man crews, the driver and the gunner. [1]

3.4.3.1 Sample Referent

Our approach to fidelity depends on defining a meaningful referent. Among other things, the referent provides a frame of reference for the analysis. For example, if a study is aimed at simulated large force structures, it is not helpful or practical to spend large amounts of time considering the fidelity of the electronic data flows between integrated circuits. So, we choose to define our frame of reference as the Avenger weapon system.

We researched the available literature on Avenger, including the Avenger Fielding Manual [1], the Avenger Platoon/Squad Manual [3], the Avenger Fire Unit Functional Test Procedure [2], the TREDs Avenger Database of Consolidated Table of Operations and Equipment [7], and the Avenger’s Operator Manual to define a referent. While many of these sources might prove useful in other studies, for this study the Avenger’s Operator Manual was sufficient.

3.4.3.2 Sample FOM

We searched the DMSO web site for registered FOMs, and other private sources. We found no particularly satisfactory FOMs for operational Federations which included Avenger, so compromised on using the Real-Time Player (RPR) FOM [26]. We understand that the RPR-FOM is a *reference* FOM, intended for use in *developing* a operational FOM but it was sufficient for our purposes. We used the class definition for *MilitaryPlatformLand* for the comparison, because it seemed the closest match to Avenger. Figure 3.4.3-2 illustrates RPR-FOM by giving the inheritance structure for the *MilitaryPlatformLand* class [12].

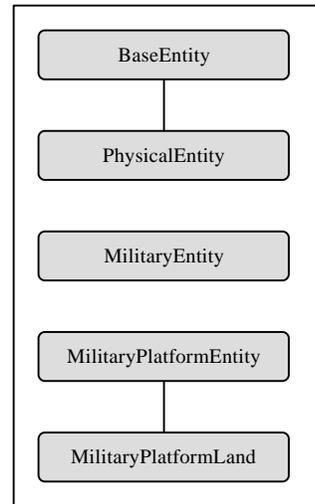


Figure 3.4.3-2: *MilitaryPlatformLand* Inheritance Tree

3.4.3.3 Sample SOM

Since we were not able to find a operational SOM for an Avenger simulation, we wrote one for Boeing-Huntsville’s Human-in-the-Loop Avenger simulator. This simulator has served as a research platform for many simulation research project inside and out of the company. For example, it was one of the very first simulators certified as DIS-compliant. Since the simulator is DIS-compliant, it was not difficult to draft a SOM for it. We built the SOM using the Object Model Development Tool available from DMSO [24], and information about the simulation provided by one of the developers [9]. This SOM is not really operational, but it is a working draft. We intend to make this simulator HLA compliant in 1998, and use this draft SOM as a point of departure.

We evaluated the fidelity of the Sample FOM and SOM against the Sample Referent for our Fidelity N-tuple. Figure 3.4.3-3 shows the fidelity scores for our sample problem, and is followed by a discussion of each score.

<i>Fidelity Aspect</i>	$F_i(FOM)$	$F_i(SOM)$
Existence	1.00	1.00
Behavior	0.41	0.41
Attributes	--	--

Figure 3.4.3-3: *Fidelity n-Tuple for Sample Problem*

Although we had to adopt some measurement technique for each fidelity aspect, we are not proposing that these techniques are “correct” in any general sense. We are

simply experimenting with what might be done. Indeed, we present alternative techniques for each aspect. These examples should serve to underscore the importance of consistent approaches to measuring fidelity, which of course is one of our primary objectives in pursuing fidelity in the first place.

3.4.3.4 Existence/Aggregation

The obvious approach to measuring the degree of aggregation in the model is to check for existence. Considering the Sample Referent for example, if a simulation presents not individual Avengers but instead models a squad of four (4) vehicles, then the fidelity aggregate component would be $\frac{1}{4}$ or 0.25. This is the approach that resulted in the fidelity scores shown in Figure 3.4.3-3, since both the Sample FOM and SOM included models of individual Avenger Weapon Systems.

Although the “perfect” fidelity result in our sample is only coincidental, this result may be un-satisfying. A different approach to existence is to attempt to look “under the skin” of the model. We could consider the system decomposition one level below the frame of reference. Figure 3.4.3-4 illustrates a typical system decomposition.

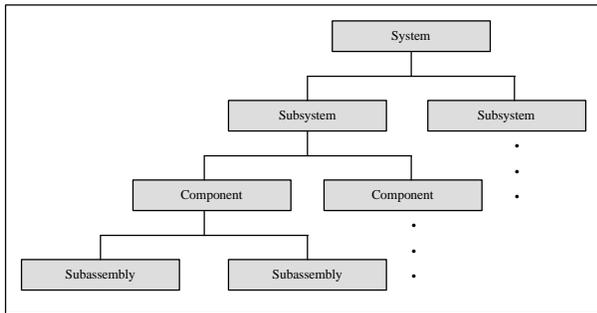


Figure 3.4.3-4: A Typical System Decomposition

The Avenger Operator’s Manual [25] lists twelve (12) elements as the first level of system decomposition below the weapon system: Radio Antenna, Missile Pod, Canopy, Laser Range Finder, FLIR, IFF Antenna HMMWV, Battery Box, Ammunition Box, Ammunition Chute, Machine Gun, Heater/Ventilator. The Sample SOM includes two: Moving Model and Avenger. The Moving Model provides data equivalent to the HMMWV, the Avenger class adds turret data. Notice that the Stinger and Threat/Target classes are outside of the system boundary of the Avenger Weapon System. This alternative approach yields a fidelity score of 0.16 – substantially different!

3.4.3.5 Behavior/Simplification

The Avenger Operator’s Manual [25] lists three main categories of operations: Operating, Troubleshooting, and Maintenance. There are a total of 44 individual procedures/operations. By considering the interactions and databases of the Sample FOM, it is possible to credit it with 18 of the required operations. Since we built the Sample SOM by adding abstractions and data to the Sample FOM, it did not gain any additional behaviors.

Many thinkers on fidelity have argued that fidelity must be measured within the purpose of the simulation [20]. In the case of our example, this might lead one to consider only the Operating operations – 24 of the total 44. Neither the Sample FOM nor SOM was intended for simulating troubleshooting or maintenance. This has the presumably happy effect of raising the fidelity score to 0.75. However, we do not think that considering the simulator’s purpose while measuring fidelity is wise for the following reasons:

- 1) simulations are frequently used for purposes other than intended,
- 2) it is difficult enough to find an “apples-to-apples” comparison for all behaviors – attempted to find the purpose-specific subpopulation will make it worse.
- 3) there is no general agreement on what the categories of simulation purposes are either.

3.4.3.6 Attributes/Clarification

Our limited time and resources did not permit us to make an adequate measure of the fidelity of the attributes in our sample problem. We did begin to assess attributes, and immediately discovered that many of the attributes required for “high” fidelity are internal to the simulation, not intended for publication through HLA interfaces.

3.4.3.7 Accuracy

It is necessary to comment on accuracy, and its relationship to fidelity. Our preliminary investigation of attributes and the difficulty of collecting information about them clarified for us the difference between accuracy and fidelity.

4. Toward Fidelity Standard(s)

4.1 Foundations

4.1.1 What Should be Standardized

A good standard provides a description of industry consensus on some subject. It describes, with varying degrees of prescriptiveness, what can or should be done to participate in that consensus. Examples range from the national electric code, to standards for nut and bolts, wire and pipes, to recommendations on improving quality in manufacturing, software design and systems engineering. Standards may be retrospective; “Our experience shows that...”, or prospective; “If this is ever to be accomplished, it should (or must, or may) be done this way.”. Because standards cover such a variety of subjects and concepts, we have a wide (perhaps bewildering) variety of forms of standard from which to choose. Almost any technical subject is appropriate for standardization if there exists a sufficiently broad based consensus about what must or should be done, or even that a particularly good way exists to accomplish, or represent, or measure a thing or subject. A form of standard (and a precedent) can be found to fit most any desired subject. Standards exist to promote the interchange of ideas and the interoperability of objects and are essential to dialog and commerce. On the other hand, the most important part of a standard is not the hard work of the participants, nor even its approval as a standard, but rather its acceptance and adoption in the marketplace. Therefore, before we commit ourselves to a standard development effort, it is good to pause and reflect. We need to establish first that the thing or idea we wish to standardize upon is sufficiently mature. Second, we need to establish that we have reasonable hope to develop not only sufficient consensus during the development of the standard to approve the standard, but that our work is reasonably likely to be rewarded by adoption in the marketplace.

4.1.2 What Should Not Be Standardized

In general, things or ideas should become standards, when the existence of the standard promotes the general welfare. Standards should not be developed that overly constrain the marketplace, or unduly restrain creativity. In our society, we have decided to standardize the general location of light switches in a room, but not on the location of the light switch in a car. In the case of fidelity, we must both measure the extent of our consensus and determine the degree and extent of standardization necessary to implement that consensus and so improve the state of practice in the M&S community. It is too early to ask what form of document we should prepare, or who should approve it, but rather

we should ask, “How best may we address to improve our community?”, and “Are we ready to address those areas in a useful way?”

4.1.3 What Needs to Be Done to Create a Standard

If we decide to pursue one or more standards, we must first develop a standard proposal with sufficient justification to convince our sponsor (In our case, the Standards activity committee) that they should fund this development. The sponsor will need to be convinced of the maturity of the concepts, that a reasonable chance of consensus for approval and adoption exists, and that a sufficient pool of volunteers exists to conduct the hard work of developing a standard. Once the sponsor is convinced (which may itself require the formation of a study group and the development of a detailed plan), a standards development group is chartered and established, or the standard development may be assigned to an existing committee working on a similar task (if one exists). The Standards development group is responsible to actually develop the standard document and they must perform their work within a specified time frame (usually a few years). The completed document is then voted upon in accord with the balloting process specified in the charter. If approved by this ballot and by the sponsor, the document becomes a standard. The marketplace then adopts the standard and uses it in business if it is successful. Depending upon the type of standard, it will expire in a specified number of years (typically five) and then must be re-approved. Clearly, the decision to develop and adopt a standard is a strong and long lasting (indeed unending) commitment and must be focused on a serious and recurring need.

4.1.4 Specifying a Standard

A specification is used to describe a products desired (required) characteristics. The characteristics described in the specification should be objective and measurable. The specification should state the circumstance under which the performance should be measured and the degree of precision to which they should be measured. Specifications include:

- 1) physical characteristics (e.g., how big, what color),
- 2) functional characteristics (e.g., the product shall do this, this well),
- 3) intrinsic quality characteristics (e.g., the software shall possess no more than 1 defect per 100 source lines of code), and

4) extrinsic quality characteristics (e.g., The product shall perform its functions without degradation under these test conditions).

Fidelity is an intrinsic quality of a simulation. The resemblance between a computer simulation product and the referent is entirely internal to the product. In order for fidelity to be measurable, the specification must clearly define the referent, along with how the resemblance between the simulation product and the referent is to be measured. This requires analysis to establish the aspects of the simuland that are interesting and important for the purpose of the simulation. This analysis is properly simulation requirements analysis, not fidelity analysis.

This kind of specification is used to procure training systems. In the case of commercially procured aircrew training systems to be used by the airlines to train their pilots. The Federal Aviation Administration requires that the internal representation of the aircraft resemble the representation of the same aircraft provided by the aircraft manufacturer to a specified accuracy. In the same way, the Department of defense specifies that training systems it procures must resemble the approved design criteria to a specified accuracy. The development of these careful specifications takes a great deal of effort. It requires that the body of knowledge about the simuland be carefully considered, and the aspects of that knowledge that is relevant to the immediate simulation need be codified. In some cases the amount of knowledge about the simulation is not adequate to obtain the desired simulation is not adequate. In these cases, either addition testing or analysis of the simuland must be performed, or assumptions must be made and documented.

The key point is that all of this work, and it is substantial and difficult, does not directly establish the fidelity required in the simulation. This effort defines the referent. If a particular simulation needs a single degree of freedom point mass representation of an object, and the simulation specification defines its referent in this way, and the resulting simulation resembles the defined referent to the defined precision, the simulation possesses 100% fidelity. It has all of the required intrinsic quality. It is useless to compare the simulation product against the entire body of knowledge about the simuland (be it well or poorly understood, real or imaginary). In order for a specification based fidelity assessment to have any meaning, the specification must completely define the referent, the environment in which the resemblance between the simulation and the referent will be measured, and the required precision of the measurement. This measurement, when completed, is the fidelity.

This conclusion is counter intuitive. It is commonly accepted that any six degree of freedom model possesses higher fidelity than any three degree of freedom model, and that the three degree of freedom model is higher fidelity than some table look up or data driven model. These perceptions exist despite often cited counter examples. If we are to make fidelity a useful concept, then we must make it measurable. The concept of defining the simulation referent in a rigorous specification is a promising approach. This approach is used today in cases where the referent is well understood and the cost of performing the needed analysis is justified by the importance of the accuracy of the simulation, i.e. FAA certification of devices used to train commercial aircraft pilots, which is a central part of all of our air safety.

The challenges of this approach are that specifications tend to be domain specific, and that the effort required to rigorously define the referent as part of the simulation specification may be more effort than available resources permit.

4.2 The Potential for Fidelity Standard(s)

Part of deciding to produce a fidelity standard must be to consider the intended use of the standard. A standard which tells us how to describe the amount of fidelity we need from simulation product, one which tells us how to measure the amount of fidelity present in an existing simulation production, or a set of standard tools, processes and data interchange standards improving our ability to produce products with better assurance all appropriate.

The very kernel of the pro-fidelity argument is that a fidelity standard would allow us to write better simulation requirements. And what is a better requirement? A better requirement is one which allows me to more easily 1) identify, 2) verify, and 3) validate what the system must be or do. Whereas there are many kinds of requirements that must be written for simulations the same as for any other system, simulations face unique challenges in defining performance and physical requirements. The challenge for simulations is unique because such requirements for simulations must tie back to a simuland. The justification of our argument is that since these are the hardest (and only unique) simulation requirements, and the most natural way to describe them is in terms of fidelity, therefore fidelity based requirements will be better requirements.

4.2.1 Validation

There is general consensus, at least in the Fidelity ISG, that simulation validation is the application with the single best potential return for a fidelity standard.

Validation is the process of answering the question “Did we build the right simulation?” [10] Ideally, this question is resolved by questioning the specified requirements – leaving to verification the somewhat more straightforward question of whether the requirements were correctly implemented. Verification can assume that the requirements are correct; validation has no such luxury.

How are we to question simulation requirements? Requirements might be classified [4] as addressing one of seven categories:

- 1) Mission Definition (e.g., operators/users)
- 2) Use cases (e.g., hours of operation)
- 3) Deployment (e.g., installation plan)
- 4) Operational Lifecycle (e.g., life span)
- 5) Effectiveness (e.g., mean time to failure)
- 6) Operational Environment (e.g., humidity)
- 7) Performance and Physical Parameters

Careful consideration of these categories reveals that simulations do not have requirements uniquely different than any other system in the first six of these categories, and the issues surrounding questioning the such requirements are the same as for any other system and not addressed by a technique such as fidelity. It is not that these requirements are unimportant, it is simply that solving their challenges is the purview of systems engineering not simulation science.

However, simulations do have uniquely different requirements in the last category, performance and physical parameters. The difference is that while other systems define performance requirements explicitly (e.g., maximum speed), simulations define them by reference to a simuland. Therefore, questioning simulation requirements requires specialized techniques. We assert that fidelity is such a technique.

Questioning a simulation performance requirement requires us to decide what the simulation must represent to achieve the simulation’s purpose. Fidelity does not directly tell us how to make this decision! Knowing the fidelity of a particular simulation’s flight dynamics algorithm does not tell us if representing flight dynamics is necessary. A flight dynamics simulation is clearly required for a full flight trainer, and not for a cockpit maintenance trainer – and this decision is unrelated to the fidelity of a particular flight dynamics simulation.

Fidelity decisions for such gross considerations require no sophisticated analysis, and this explains why simulation development has been able to proceed as far as it has without grappling with fidelity. However, our current inability to measure and express the fidelity of a simulation limits our ability to make more fine grained decisions such as what kind of flight dynamics algorithm is appropriate for a cockpit procedures trainer. This inability is even more crippling when one considers the difficulties that arise when creating fair federations of models that were never intended to work together.

Fidelity does not by itself tell us how to decide is a simulation is “good enough”, however, fidelity does give us a way to form the question. The conceptualization of fidelity that emerged from the ISG provides a powerful means of expressing how the simulation will or does correspond to the referent. This ability to express the problem arms us to begin making real progress in scientifically developing simulations. For example if we can define and measure fidelity, we can develop simulations of measurable different fidelities for the same referent, and test their ability to satisfy specific purposes. By such approaches, we can thus show that a specific fidelity level is appropriate for a specific purpose, and write meaningful simulation requirements.

Interestingly, the FAA has foreshadowed this approach. In the FAA training device certification, the FAA circular that specifies training devices includes some specific features that relate to the specific usability of a training device for its intended purpose. Similarly, the airline or training center acquiring the training device will specify features they desire in their system. Usability relates directly the purpose of the simulator, and not directly to fidelity as discussed herein. And those levels of features are direct statements of the simulator’s fidelity. By clarifying their conceptualization of fidelity and their ability to express it, the FAA was able to bring real order into the business of procuring and developing flight simulators. By careful analysis, the FAA has substantially simplified the validation problem by telling trainers what features a given purpose requires.

Of course, the FAA’s success was built on years of diligent effort focused in a relatively restricted domain. Fidelity will succeed as a powerful validation technique in so far as we are able to define domain independent ways to characterize it. We think that the key to this success lies in defining measurement techniques for fidelity.

4.2.2 Verification

A related potential application for a fidelity standard is as a verification technique. Verification is often described

as answering the question “Did I build the simulation right?” [10] As discussed, this ideally reduces to the determination that the specified requirements have been correctly implemented.

Obviously, a fidelity standard would provide us with a powerful way of tracking the implementation of simulation unique requirements through a development. If we had standard techniques for measuring fidelity, then we could measure it during the development process and thereby demonstrate that the simulation unique requirements are correctly represented.

Returning to the FAA example described previously, we note that FAA verification techniques assess fidelity via an automated test guide, which is a carefully designed test set up that verifies that the aircraft model used in a civil aircrew trainer does indeed replicate the referent. We note again that this kind of check against the referent is the only simulation unique aspect of verification. The computer simulation is, after all, a computer program, and the assessment of its quality is identical to the assessment of any other computer program. The thing that defines a computer program as a simulation is that its intended purpose is to represent something else, i.e. the referent. The measurement of the quality of that representation (the measurement of its fidelity) is the only simulation unique aspect of simulation verification. All other verification activities are identical to those applied to any other computer program of a similar complexity and criticality.

The implication of this conclusion is that fidelity verification should be a rewarding area for standardization. If done properly, this standard development would harmonize with software verification research and lend support to the development of simulation unique fidelity verification tools that work well with commercial verification tool suites.

4.2.3 Interoperability

Research and development to date in HLA based interoperability, while making significant progress, has not really addressed the most significant interoperability challenges. Work to date as focused on what might be termed “intercommunication” rather than interoperability. The intercommunication problem is largely solved and fidelity has little to add.

However, real interoperability is more than simply data flows between simulations. The dream of technologies like HLA is the rapid assembling of never before conceived federations, with little or no a priori consideration of the issues involved in making them work together. If we are to even approach this dream, then we will need powerful new techniques to

characterize simulations and predict their ability to work together “just-in-time”.

Since these issues are again unique to simulation, we believe that a fidelity standard would provide traction on grappling with real interoperability. Just as fidelity can give us the language to question a requirement, it can give us the language to question whether simulations will operate on level playing fields.

4.2.4 Requirements

The foregoing discussion should make clear that we believe that standardized fidelity techniques would provide substantial improvement in the way that requirements unique to simulation are captured, expressed, and utilized.

5. Recommendations

The Fidelity ISG has formulated a number of specific recommendations to the SISO and large simulation communities for consideration and action. The Fidelity ISG was chartered to develop:

- 1) A lexicon for simulation fidelity terms and concepts.
- 2) A contextual framework relating those terms and concepts and simulation theory.
- 3) A set of methods and metric by which fidelity is defined, estimated, and measured.

The original term of the ISG was through the March 1999 SIW. While we believe that the ISG has substantially achieved its original charter, we believe that the charter should be expanded to address the following recommendations, and remove the defined end time. The ISG has not had sufficient time to prototype and experiment with fidelity standards, which is a required step prior to a formal standards development group. We suggest that an ISG, in preference to a new forum, is still the correct vehicle for this activity. However, the ISG should make provision for rotation of officers as interest and availability change.

5.1 Encourage Use of Common Terminology

The simulation fidelity literature continues to reveal much confusion in terminology and constructs. It would be very helpful for the community to start using a common vocabulary consistently when addressing simulation fidelity. Up to this point, individual authors, and certainly individual sub-communities within simulation have defined and used fidelity-related terms to suit their own purposes. This makes it very difficult for ideas from one part of the simulation community to be as useful for others as one would desire. The terminology of each sub-community should be as compatible as possible with the terminology of the large simulation community.

We assert that the Fidelity Glossary, developed by the ISG and appearing as an appendix to this document forms the basis for a common fidelity terminology. The ISG should formulate a formal recommendation for revising the *DMSO M&S Glossary* with the core fidelity terms, and explore revising all of the effected terms in the *DMSO M&S Glossary*.

We recommend that all SISO Planning & Review Panels (PRPs) should use the Fidelity Glossary for correction of fidelity related papers. The PRPs should be encouraged to review the Fidelity Glossary and recommend changes.

5.2 Prototype a Fidelity Framework Standard

The Fidelity ISG should develop and build consensus on a standard fidelity framework, defining the mathematical relationship between fidelity-related terms and concepts. This framework should provide the context within which prototypes and experiments with standards would occur. Appendix B provides our best work to date in this arena.

5.3 Identify Standard Fidelity Referents

Progress in making practical use of fidelity depends upon the availability of proven fidelity referents for specific application domains. The ISG should work with individual simulation sub-communities should generate examples of proven fidelity referents. Those referents that have demonstrated their usefulness in particular application domains can then become an input to more generalized fidelity referents. This includes the recommendation that the ISG work with the CMMS project to incorporate its findings on fidelity.

5.4 Prototype a Fidelity-Based Validation Process

As discussed, fidelity offers much promise for simulation validation, however there are currently no proposed processes for using fidelity in validation. The ISG should develop and experiment with a prototype fidelity-based validation process which directly addresses the question: "How good is good enough?", or, "How does one determine level of fidelity is required for a particular application?".

5.5 Prototype a Fidelity-Based Verification Process

As discussed, fidelity offers much promise for simulation verification, however there are currently no proposed processes for using fidelity in verification. The ISG should develop and experiment with a prototype fidelity-based validation process which directly addresses the question: "Does this implementation have the specified level of fidelity?".

Bibliography

This bibliography only addresses the references in the main body. The appendices include separate bibliographies.

- [1] Avenger Fielding Manual, D433-20635-1, Boeing August 1991.
- [2] Avenger Fire Unit Functional Test Procedure, 13264901, Boeing, 25 April 1988.
- [3] Avenger Platoon/Squad, FM 44-44, US Army, August 1991.
- [4] Blanchard, Benjamin S. and Wolter J. Fabrycky, Systems Engineering and Analysis, Prentice Hall, 1981
- [5] Casti, John L., *Alternative Realities – Mathematical Models of Nature and Man*, John Wiley & Sons, 1989.
- [6] Conceptual Models of the Mission Space (CMMS) Technical Framework, Defense Modeling and Simulation Office, February, 1997
- [7] “Consolidated Table of Operations and Equipment“, TREDIS Avenger Database, US Army, November 1997.
- [9] Crispen, Robert, Personal Communication, 15 November 1997.
- [10] Department of Defense Verification, Validation and Accreditation (VV&A) Recommended Practices Guide, Office of the Director of Defense Research and Engineering, Defense Modeling and Simulation Office, November, 1996
- [11] DMSO web page, <http://www.dmsomil.com>, 1996.
- [12] Foster, Lester. “Fidelity in Modeling and Simulation”, Proceedings of the Spring 1997 Simulation Interoperability Workshop, 97S-SIW-073.
- [13] Gross, David et al. “Measuring Fidelity Differentials in HLA Simulations”, Proceedings of the Fall 1997 Simulation Interoperability Workshop, 97F-SIW-125.
- [14] Gross, David et al. “A Case Study in Fidelity Differential Measurements”, Proceedings of the Spring 1998 Simulation Interoperability Workshop, 98S-SIW-185.
- [15] Hunt, Ken, Dr. Judith Dahmann, Robert Lutz, and Jack Sheehan, "Planning For the Evolution of Automated Tools in HLA," , 97S-SIW-067, Simulation Interoperability Workshop, Spring, 1997
- [16] IEEE P1516.2/D3 Draft Standard for Modeling and Simulation (M&S) High Level Architecture (HLA) – Object Model Template (OMT), Simulation Interoperability Standards Committee (SISC) of the IEEE Computer Society, November, 1998
- [17] IPO Dictionary / Methodology Committee draft, 4 Sept 1997. [5A] Klir, George J., *Facets of Systems Science*, Plenum Press, 1991.
- [18] Pace, Dale K., “Synopsis of Fidelity Ideas and Issues,” 98 *Spring Simulation Interoperability Workshop Papers*, Volume 1, March 1998, pp. 420-429.
- [19] Pace, Dale K., “Dimensions and Attributes of Simulation Fidelity,” 98 *Fall Simulation Interoperability Workshop Papers*, September 1998.
- [20] Lorenzo, Max, Communications via RDE Forum Reflector, 13 November 1997.
- [21] MacDonald, Bruce. “TBD”, Proceedings of the Fall 1997 Simulation Interoperability Workshop, 97F-SIW-TBD.
- [22] Morrison, John Dale, and Michael D. McKay, “Toward practical Mathematical Methods for Conducting Model Design and Validation,” Proceedings of the Fall 1998 Simulation Interoperability Workshop, September 14-18, 1998, Orlando, FL.
- [23] Oberkampf, William L., Kathleen V. Diegert, Kenneth F. Alvin, and Brian M. Rutherford, “Variability, Uncertainty, and Error in Computational Simulation,” *HTD-Vol. 357-2, ASME Proceedings of the 7th. AIAA/ASME Joint Thermophysics and Heat Transfer Conference* (Book No. H1137B – 1998).
- [24] Object Model Development Tool, Version 1.1.9 AEGIS Research , 10 December 1997.
- [25] Operator’s Manual for Avenger, TM 9-1425-433-10, US Army, March 1989.
- [26] Real-time Platform Reference FOM Version 1.0, GEC-Marconi RDS Ltd. Simulation and Training Division, 9 March 1997.

- [27] Schow, Greg, et al. "Mathematical Foundations for a Fidelity Differential Calculus", Proceedings of the Spring 1997 Simulation Interoperability Workshop, 97S-SIW-022.
- [28] Zeigler, Bernard P., "A Framework for Modeling and Simulation," Chapter 3 in Modeling and Simulation: An Integrated Approach to Development and Operation, Cloud D.J. and L.B. Raines (eds.), McGraw-Hill, 1998, pp. 67-103.
- [29] Zeigler, Bernard P., "Components of a Theory of Modeling and Simulation," *Volume 9 Modeling and Simulation, Technology for the United States Navy and Marine Corps 2000-2035: Becoming a 21st-Century Force*, (NRC Naval Studies Board) National Academy Press, 1997.

Appendix A: The Fidelity ISG Glossary

A

absorbing Markov chain model. A Markov chain model that has at least one absorbing state and in which from every state it is possible to get to, at least, one absorbing state. [1, 2]

absorbing state. In a Markov chain model, a state that cannot be left once it has been entered. [1]

abstraction. 1. The process of selecting the essential aspects of a simuland to be represented in a model or simulation while ignoring those aspects that are not relevant to the purpose of the model or simulation. 2. The set of elements produced by this process. [3] 3. The act or process of separating the inherent qualities or properties of something from the actual physical object or concept to which they belong. 4. A product of this process, as a general idea or word representing a physical concept. [4]

accessibility. The ease of approaching, entering, obtaining, or using. [5]

accreditation. Official acceptance or certification that a model, the data for a simulation or a simulation is suitable for a specific purpose or application. [41]

accuracy. The degree to which a parameter or variable or set of parameters or variables within a model or simulation conform exactly to reality or to some chosen standard or referent. See resolution, fidelity, precision. [41]

activity model. A model of the processes that make up a functional activity showing inputs, outputs, controls, and mechanisms through which the processes of the functional activity are or will be conducted. [6]

activity. A task that consumes time and resources and whose performance is necessary for a system to move from one event to the next. [2]

activity-based simulation. A discrete simulation that represents the components of a system as they proceed from activity to activity (e.g., a simulation in which a manufactured product moves from station to station in an assembly line). [1]

affected attributes. The specific attributes of an object class instance whose value in a federation execution may be affected by that instance's participation in a dynamic interaction with another instance of the same class or an instance of another object class. [7]

aggregation. 1. The ability to group items, whether entities or processes, while preserving the effects of item behavior and interaction while grouped. [8] 2. A relationship between objects in the data model where one object contains other objects. See disaggregation. [9]

aggregator. An object that is comprised of other objects; a 'has-a' relationship exists between the aggregator object and its component objects (e.g., a polygon is an aggregator for its vertex objects). [9]

algorithm. A prescribed set of well-defined, unambiguous rules or processes for solving a problem in a finite number of steps. [10]

alternate key. A property or characteristic that can be used as a secondary identifier for an entity or entity class. [11]

analytical model. A model consisting of a set of solvable equations (e.g., a system of solvable equations that represent the laws of supply and demand in the world market). [1, 2]

application. 1. The specific use or purpose that a model or simulation serves. 2. A simulation or simulation-related software package that is executed within a larger simulation framework or infrastructure. [41]

architecture. The structure of components in a program or system, their interrelationships, and the principles and guidelines governing their design and evolution over time. [8]

association. 1. A type of static relationship between two or more object classes, apart from class-subclass or part-whole relationships. [7] 2. The weakest relationship between two or more objects in a data model including the multiplicity of objects at either end of the relationship. [9]

associative entity. An entity that inherits its primary key from two or more other entities (i.e., those that are associated). An associative entity may represent many-to-many relationships. [12]

asynchronous transmission. A transmission in which each information character is individually synchronized, usually by the use of start elements and stop elements. [13]

atmosphere. 1. The mass of air surrounding the earth and the features embedded within it, including clouds, smoke, and fog. 2. A kind of mission space entity representing the atmosphere. [14]

attribute overloading. The ability of an attribute to carry one of two or more separate facts. [17]

attribute ownership. The property of a federate to have the responsibility to publish values for a particular object attribute. [7]

attribute. 1. A property or characteristic of one or more entities or objects (e.g., COLOR, WEIGHT, SEX). 2. A property inherent to an entity or associated with that entity for database purposes. [6, 15, 16] 3. A quantifiable property of an object (e.g., the color of a building or the width of a road). [9]

attributive entity. An entity that has the same primary key as the parent and additional attributes that eliminate the occurrence of repeating groups in the parent. [14]

authoritative data source. A data source whose products have undergone producer data verification, validation and certification activities. [3]

authoritative representation. Models, algorithms, and data that have been developed or approved by a source which has accurate technical knowledge of the entity or phenomenon to be modeled and its effects. [3]

automated force (AFOR). The most automated computer-generated force that requires little or no human interaction to accomplish its mission. [8]

axiom. 1. A statement or proposition used in the premises of arguments and assumed as self-evidently true without proof. [39] 2. A well formed formula that is stipulated rather than proved to be so through the application of rules of inference. [35]

B

battlespace entity. A simulated entity that corresponds to actual equipment, supplies, and personnel that can be seen or sensed on a real battlefield. [13]

battlespace. The physical environment in which the simulated warfare will take place and the forces that will conduct the simulated warfare. All elements that support the front line forces (e.g., logistics, intelligence) are included in this definition of battlespace. [8]

behavior. 1. For a given object, how attribute value changes affect or are affected by the attribute value changes of the same or other objects. [14] 2. The way in which a system responds to stimuli over time. [41]

benchmark. 1. The activity of comparing the results of a model or simulation with an accepted representation of the process being modeled. [1] 2. An accepted

representation or standard of a process being modeled or simulated against which the results of other models or simulations are compared or judged. [41]

benchmarking. The comparison of a model's output with the outputs of other models or simulations, all of which represent the same input and environmental conditions. [18]

bit. 1. The smallest unit of information in the binary system of notation. [1, 2] 2. A unit of information representation capacity where the information capacity of a device is equal to the logarithm to the base two of the number of possible representation states of that device. [19]

black box model. A model whose inputs, outputs, and functional performance are known, but whose internal implementation is unknown or irrelevant (e.g., a model of a computerized change-return mechanism in a vending machine, in the form of a table that indicates the amount of change to be returned for each amount deposited); input/output model. See glass box model. [1, 2]

boundary condition. The values assumed by the variables in a system, model, or simulation when one or more of them is at a limiting value at the edge of the domain of interest. See final condition, initial condition. [1, 2]

broadcast. A transmission model in which a single message is sent to all network destinations (i.e., one-to-all); a special case of multicast. See multicast, unicast. [1, 2]

built-in-simulation. A special-purpose simulation provided as a component of a simulation language (e.g., a simulation of a bank that can be made specific by stating the number of tellers, number of customers, and other parameters). [1, 2]

built-in-simulator. A simulator that is built into the system being modeled (e.g., an operator training simulator built into the control panel of a power plant such that the system can operate in simulator mode or in normal operating mode). [1, 2]

C

cancellation. A mechanism used in optimistic synchronization mechanisms such as Time Warp to delete a previously scheduled event; a mechanism used within the Time Warp executive, normally not visible to the federate, and implemented (in part) using the Runtime Infrastructure event retraction mechanism. [7]

candidate key. An attribute or group of attributes that might be chosen as a primary key. [12]

causal order. A partial ordering of messages based on the "causally happens before" (\rightarrow) relationship; message delivery service is said to be causally ordered if for any two messages M_1 and M_2 (containing notifications of events E_1 and E_2 respectively) that are delivered to a single federate where $E_1 \rightarrow E_2$, then M_1 is delivered to the federate before M_2 . [7]

characteristic data. Empirical, synthesized or otherwise provided parameters describing the characteristics of the system or component being simulated, e.g., gross vehicle mass, gear ratio, terrain resistance coefficient. See parameter, data. [41]

class hierarchy. A specification of a class-subclass, or "is-a" relationship between object classes in a given domain. [7]

class. A description of a group of objects with similar properties, common behavior, common relationships, or common semantics. [7]

coenetic variable. A variable that affects both the system under consideration and that system's environment. [2]

complex data. Data that cannot be characterized as a single concept or atomic data element including highly derived data (e.g., probability hit/kill); objects utilizing the concepts of multiple inheritance (e.g., student-assistant is subclass of student class and employee class), multiple root hierarchies (e.g., a tank is a vehicle and a tank is a weapon where "vehicle" and "weapon" are each roots), and polymorphic attributes (e.g., "capacity" for different types of aircraft may mean number of people, pounds of cargo, or gallons of fuel); compositions such as command hierarchies, road networks, images (binary large objects), compound documents; and artifacts of legacy systems and physical constraints (e.g., aircraft category and mission in one data element, intelligence facility code where the first few bytes define how the rest of the field is used. [8]

component class. An object class which is a component, or part of, a "composite" object which represents a unified assembly of many different object classes. [7]

component. An object that is a part of an aggregator object. [9]

composite attribute. A single attribute that is composed of a specific set of identifiable pieces of information

(e.g., an address made up of a street number, city, state, and zip code). [12]

compression. Any of several techniques that reduce the number of bits required to represent information in data transmission or storage, therefore conserving bandwidth and/or memory, so the original form of the information can be reconstructed; Compaction. [13]

computational model. A model consisting of well-defined procedures that can be executed on a computer (e.g., a model of the stock market, in the form of a set of equations and logic rules). [2]

computer generated force (CGF). A computer representation of forces in simulations that attempt to model human behavior sufficiently so that those forces will take some actions automatically (without requiring man-in-the-loop interaction); semi-automated force. [8]

computer hardware. Devices capable of accepting and storing computer data, executing a systematic sequence of operations on computer data, or producing control outputs; such devices can perform substantial interpretation, computation, communication, control, or other logical functions. [20]

computer resources. The totality of computer hardware, firmware, software, personnel, documentation, supplies, services, and support services applied to a given effort. [14]

computer simulation. A dynamic representation of a model, often involving some combination of executing code, control/display interface hardware, and interfaces to real-world equipment. [14]

computer software. A set of computer programs, procedures, and associated documentation concerned with the operation of a data processing system (e.g., compilers, library routines, manuals, and circuit diagrams); software. [14]

Conceptual Model of the Mission Space (CMMS). First abstraction of the real world that serves as a frame of reference for simulation development by capturing the basic information about important entities involved in any mission and their key actions and interactions; simulation-neutral view of those entities, actions, and interactions occurring in the real world. [14]

conceptual model. 1. A description of the content and internal representations that are the user's and developer's combined concept of the model including logic and algorithms and explicitly recognizing assumptions and limitations. [1] 2. An implementation-independent description of the content and internal

representations that represent the sponsor's, user's and developer's combined concept of the system or simulation under development including logic, architecture, algorithms, available data and explicitly recognising assumptions and limitations. [41]

conceptual schema. A descriptive representation of data and data requirements that supports the "logical" view or data administrator's view of the data requirement. This view is represented as a semantic model of the information that is stored about objects of interest to the functional area. This view is an integrated definition of the data that is unbiased toward any single application of data and is independent of how the data is physically stored or accessed. [6]

concrete model. A model in which at least one component represented is a tangible object (e.g., a physical replica of a building). [1, 2]

condition. The values assumed at a given instant by the variables in a system, model, or simulation. See boundary condition; final condition; initial condition; state. [1, 2]

conditional event. A sequentially dependent event that will occur only if some other event has already taken place. See time-dependent event. [1, 2]

configuration. A collection of an item's descriptive and governing characteristics, which can be expressed: in functional terms (i.e., what performance the item is expected to achieve); and in physical terms (i.e., what the item should look like and consist of when it is built). [14]

conservative synchronization. A mechanism that prevents a federate from processing messages out of time stamp order (e.g., Chandry/Misra/Bryant null message protocol). See optimistic synchronization. [7]

consistency. Data maintained so that it is free from variation or contradiction. [5, 6]

constant. A quantity or data item whose value cannot change. [2]

constrained simulation. A simulation where time advances are paced to have a specific relationship to wall clock time; real-time or scaled-real-time simulations (e.g., human-in-the-loop (e.g., training exercises), hardware-in-the-loop (e.g., test and evaluation simulations)). [7]

constructive model or simulation. Models or simulations that involve simulated people operating simulated systems. Real people may make inputs to such

simulations, but are not involved in determining their outcomes. [8]

container. See aggregator. [9]

context. 1. The material surrounding an item that helps define its meaning. [36] 2. The circumstances in which a particular event occurs; the situation. [4]

continuous model. A mathematical or computational model whose output variables change in a continuous manner. See discrete model. [1, 2]

continuous simulation. A simulation that uses a continuous model. [1, 2]

continuous system. A system for which the state variables change continuously with respect to time. [10]

coordinate system. An organized system for describing 2- or 3-dimensional locations. [9]

coordinate. 1. Linear or angular quantities which designate the position that a point occupies in a given reference frame or system. 2. A general term to designate the particular kind of reference frame or system, such as Cartesian coordinates or spherical coordinates. [13] 3. One of a set of numbers that determines the location of a point in a space of a given dimension. 4. Any of a set of two or more magnitudes used to determine the position of a point, line, curve or plane. [4]

coordinated time advancement. A time advancement mechanism where logical clock advances within each federate only occur after some coordination is performed among the federates participating in the execution (e.g., to ensure that the federate never receives an event notice in its past). [7]

correlated initial environment. The convergent representation of the same physical environment in two or more separate synthetic environments prior to their use in a combined exercise. [9]

correlated levels of detail. The equal representation of synthetic environment objects at comparable levels of presentation (i.e., the same object seen or detected at a distance of 15 meters.). [9]

correlation. 1. A convergent relationship between parallel representations of the same data. [9] 2. A causal, complementary, parallel, or reciprocal relationship, especially a structural, functional, or qualitative correspondence between comparable entities. [4]

critical event simulation. A simulation that is terminated by the occurrence of a certain event (e.g., a model depicting the year-by-year forces leading up to a volcanic eruption, that is terminated when the volcano in the model erupts. See also: time-slice simulation). [1, 2]

cultural features. Features of the environment that have been constructed by man including such items as roads, buildings, canals, marker buoys; boundary lines, and, in a broad sense, all names and legends on a map. [14]

D

data architecture. The framework for organizing and defining the interrelationships of data in support of an organization's missions, functions, goals, objectives, and strategies. Data architectures provide the basis for the incremental, ordered design and development of databases based on successively more detailed levels of data modeling. [6]

data attribute. A characteristic of a unit of data such as length, value, or method of representation. [15, 21]

data collection. The process of obtaining information that supports a functional activity, or information requirement. [6]

data derivation. The calculation or interpolation of information not present in the original data. [9]

data dictionary. A specialized type of database containing metadata that is managed by a data dictionary system; a repository of information describing the characteristics of data used to design, monitor, document, protect, and control data in information systems and databases. [15, 22]

data element. A basic unit of information having a meaning and subcategories (data items) of distinct units and values (e.g., address). [22]

data exchange standard. Formally defined protocols for the format and content of data messages used for interchanging data between networked simulation and/or simulator nodes used to create and operate a distributed, time and space coherent synthetic environment. [23]

data integrity. The condition in which data is accurate, current, consistent, and complete. [6]

data logger. A device that accepts Protocol Data Units (PDUs) from the network and stores them for later replay on the network in the same time sequence as the PDUs were originally received. See protocol data unit. [1, 2]

data loss. The loss of original information through multiple conversions or transformations of data. [9]

data model. 1. The user's logical view of the data in contrast to the physically stored data, or storage structure. 2. A description of the organization of data in a manner that reflects the information structure of an enterprise. [6, 15, 16] 3. A description of the logical relationships between data elements where each major data element with important or explicit relationships is captured to show its logical relationship to other data elements. [9]

data quality. The correctness, timeliness, accuracy, completeness, relevance, and accessibility that make data appropriate for use. Quality statements are required for source, accuracy (positional and attribute), up-to-dateness/currency, logical consistency, completeness (feature and attribute), clipping indicator, security classification, and releasability. [6, 8]

data repository. A specialized database containing information about data, such as meaning, relationships to other data, origin, usage, and format, including the information resources needed by an organization. [6]

data representation. 1. A format used to describe some type of data. [41] 2. A variety of forms used to describe a terrain surface, the features placed on the terrain, the dynamic objects with special 3-D model attributes and characteristics, the atmospheric and oceanographic features, and many other forms of data. [9]

data source. 1. An organization or subject matter expert who, because of either mission or expertise, serves as a data producer. [14] 2. A publication that serves as an authoritative source of data used in a model or simulation. [41]

data structure. The logical relationships that exist among units of data and the descriptive features defined for those relationships and data units; an instance or occurrence of a data model. [15, 21]

data synchronization. The timing requirements of a data element, or between and/or among data elements. [6]

data validation. The documented assessment of data by subject area experts and its comparison to known values. [8]

data value. A value associated with a data element; one of the allowable values of a data element. [6, 22]

data verification. Data producer verification is the use of techniques and procedures to ensure that data meets

constraints defined by data standards and business rules derived from process and data modeling. Data user verification is the use of techniques and procedures to ensure that data meets user specified constraints defined by data standards and business rules derived from process and data modeling, and that data are transformed and formatted properly. [8]

data. 1. A representation of facts, concepts, or instructions in a formalized manner suitable for communication, interpretation, or processing by humans or by automatic means. [6, 15, 16] 2. Assumed, given, measured, or otherwise determined facts or propositions used to draw a conclusion or make a decision. [4]

database directory. A database of entries each of which represents information about a database or a directory of databases including the name of a database or directory, ownership, point of contact, access path to the database or directory, description of purpose of database. [14]

database management system (DBMS). A system that provides the functionality to support the creation, access, maintenance, and control of databases, and that facilitates the execution of application programs using data from these databases. [14]

database. A collection of interrelated data, often with controlled redundancy, organized according to a schema to serve one or more applications; the data are stored so that they can be used by different programs without concern for the data structure or organization. A common approach is used to add new data and to modify and retrieve existing data. [6, 15, 16]

dead reckoning. The process of extrapolating emulation entity position/orientation based on the last known position/orientation, velocity, and (sometimes) higher-order derivatives of position vs. time and/or other vehicle dynamic characteristics; remote entity approximation. [1, 13]

deaggregate. See disaggregate. [14]

deaggregation. The ability to separate grouped items, whether entities or processes, while preserving the effects of item behavior and interaction whether grouped or separated. [3]

dependent variable. A variable whose value is dependent on the values of one or more independent variables. See independent variable. [1, 2]

descriptive model. A model used to depict the behavior or properties of an existing system or type of system (e.g., a scale model or written specification used to convey to potential buyers the physical and performance

characteristics of a computer). See prescriptive model. [1, 2]

detail. 1. A separately considered part or item. [44] 2. Any of the small parts that go to make up something; particular. 2. The act of dealing with things item by item. See abstraction, level of detail, resolution. [45]

deterministic algorithm. A process that yields a unique and predictable outcome for a given set of inputs. [10]

deterministic model. A model in which the results are determined through known relationships among the states and events, and in which a given input will always produce the same output (e.g., a model depicting a known chemical reaction). See stochastic model. [1, 2]

deterministic. Pertaining to a process, model, simulation or variable whose outcome, result, or value does not depend upon chance. See stochastic. [2, 13]

digital simulation. 1. A simulation that is designed to be executed on a digital system. 2. A simulation that is designed to be executed on an analog system but that represents a digital system. 3. A simulation of a digital circuit. See analog simulation, hybrid simulation. [1, 2]

disaggregate. Activity that decomposes an aggregated entity into multiple entities representing its components. [1]

disaggregation. The ability to represent the behavior of an aggregated unit in terms of its component entities. If the aggregate representation did not maintain state representations of the individual entities, then the decomposition into the entities can only be notional. [8]

discrete model. A mathematical or computational model whose output variables take on only discrete values; that is, in changing from one value to another, they do not take on the intermediate values (e.g., a model that predicts an organization's inventory levels based on varying shipments and receipts). See continuous model. [1, 2]

discrete simulation. A simulation that uses a discrete model. [1, 2]

discrete system. A system for which the state variables change instantaneously at separated points in time. [10, 24]

domain analysis. The process of identifying, acquiring and evaluating the information related to a problem domain to be used in specifying and constructing a model or simulation. [41]

domain. The physical or abstract space in which the entities and processes operate. The domain can be land, sea, air, space, undersea, a combination of any of the above, or an abstract domain, such as an n-dimensional mathematics space, or economic or psychological domains. [18]

dynamic model. A model of a system in which there is change, such as the occurrence of events over time or the movement of objects through space (e.g., a model of a bridge that is subjected to a moving load to determine characteristics of the bridge under changing stress). [1, 2]

dynamic natural environment. The natural environment which is constantly changing as a result of man-made efforts (battlefield smoke) and natural phenomenon (weather). [23]

E

edge. A one dimensional primitive used to represent the location of a linear feature and/or the border of faces. [9]

elevation. The vertical component in a 3-dimensional measurement system measured in reference to a fixed datum. [9]

emitter. A device that is able to discharge detectable electromagnetic or acoustic energy. [1, 13]

empirical. Pertaining to information that is derived from observation, experiment, or experience. [1, 2]

emulate. To represent a system by a model that accepts the same inputs and produces the same outputs as the system represented (e.g., to emulate an 8-bit computer with a 32-bit computer). [1, 2]

emulation. A model that accepts the same inputs and produces the same outputs as a given system. See simulation. [1, 2]

emulator. A device, computer program, or system that performs emulation. [1, 2]

encapsulation. The process of hiding the details of an object that do not contribute to its essential characteristics. [25]

endogenous variable. A variable whose value is determined by conditions and events within a given model; internal variable. See exogenous variable. [1, 2]

enterprise model. An information model(s) that presents an integrated top-level representation of processes, information flows, and data. [6, 26]

entity coordinates. Location with respect to a simulation entity. [1]

entity perspective. The perception of the synthetic environment held by a simulation entity based on its knowledge of itself and its interactions with the other simulation entities including not only its own view of the simulated physical environment (terrain, air, and sea), but also its own view of itself, the other entities in the synthetic environment, and of the effects of the other entities on itself and the synthetic environment; world view. [1]

entity relationship diagram (ERD). A graphic representation of a data model. [14]

entity. 1. A distinguishable person, place, unit, thing, event, or concept about which information is kept. [12]
2. Something that exists as a particular and discrete unit. [4]

environment. The texture or detail of the natural domain, that is terrain relief, weather, day, night, terrain cultural features (such as cities or farmland), sea states, etc.; and the external objects, conditions, and processes that influence the behavior of a system (such as terrain relief, weather, day/night, terrain cultural features, etc.). [1]

environmental database. See synthetic environment database. [9]

environmental domain. The physical or abstract space in which the entities and processes operate. The domain can be land, sea, air, space, undersea, a combination of any of the above, or an abstract domain, such as an n-dimensional mathematics space, or economic or psychological domains. [9]

environmental effect model. A numerical model, parametric model, or database for simulating a natural environmental effect on an entity of a simulation exercise, such as a sensor or platform. [14]

environmental effect. The impact that the natural environment or environmental feature has on some component or process in the simulation exercise such as the propagation of energy and image formation, the performance of a weapon system, platform or sensor, or other non-visualized combat process. [14]

environmental entity. A simulation entity that corresponds to dynamic elements of the natural state of the geographic, atmospheric, and bathyspheric environment, of the synthetic environment, that can be seen or sensed on a real battlefield (e.g., craters, smoke, building collapse, weather conditions, and sea state). [1]

environmental features. An individual element of the natural environment (e.g., a rain system, fog, cloud). [14]

environmental model. A numerical model, parametric model, or database designed to produce an accurate and consistent data set for one or more parameters that characterize the state of the natural environment. [14]

environmental phenomenon. An individual element of the physical environment (e.g., a rain system, fog, cloud). [9]

environmental representation. An authoritative representation of all or a part of the natural or man-made environment, including permanent or semi-permanent man-made features. [8]

environmental simulation. A simulation that depicts all or part of the natural or manmade environment of a system (e.g., a simulation of the radar equipment and other tracking devices that provide input to an aircraft tracking system). [2]

equation of state. 1. A relation, empirical or derived, between the properties describing the state of a substance or system. [48] 2. The relationship between observables in a natural system. See state, observables, natural system, input, output. [41]

equilibrium. See steady state. [1] A condition in which all acting influences are canceled by others, resulting in a stable, balanced or unchanging system. [4]

error model. 1. A model used to estimate or predict the extent of deviation of the behavior of an actual system from the desired behavior of the system (e.g., a model of a communications channel, used to estimate the number of transmission errors that can be expected in the channel). 2. In software evaluation, a model used to estimate or predict the number of remaining faults, required test time, and similar characteristics of a system. [1, 2]

error. The difference between an observed, measured or calculated value and a correct value. [3]

Euler angles. A set of three angles used to describe the orientation of an entity as a set of three successive rotations about three different orthogonal axes (x, y, and z). The order of rotation is first about z by angle (ψ), then about the new y by angle (θ), then about the newest x by angle (ϕ). Angles ψ and ϕ range between $\pm\pi$, while angle θ ranges only between $\pm\pi/2$ radians. These angles specify the successive rotations needed to transform from the world coordinate system to the entity coordinate system. The positive

direction of rotation about an axis is defined by the right-hand rule. [1]

event notice. A message containing event information. [7]

event. 1. A change in an object attribute value, an interaction between objects, an instantiation of a new object, or a deletion of an existing object that is associated with a particular point on the federation time axis. [7] 2. An individual stimulus from one object to another at a particular point of time. [41]

event-oriented simulation. A simulation in which attention is focused on the occurrence of events and the times at which those events occur; for example, a simulation of a digital circuit that focuses on the time of state transition. [1, 2]

exercise. The execution of a simulation configured with specific parameters, characteristic data, initial conditions, players and external systems, and intended to represent a specific or general scenario. See simulation execution. [41]

exogenous variable. A variable whose value is determined by conditions and events external to a given model; external variable. See endogenous variable. [1, 2]

extensibility. The ability of a data structure to accommodate additional values or iterations of data over time without impacting its initial design. [5, 6]

external schema. A logical description of an enterprise that may differ from the conceptual schema upon which it is based in that some entities, attributes, or relationships may be omitted, renamed, or otherwise transformed. [6]

F

face validation. The process of determining whether a model or simulation seems reasonable to people who are knowledgeable about the system under study, based on performance. This process does not review the software code or logic, but rather reviews the inputs and outputs to ensure they appear realistic or representative. [1, 13]

face. A region enclosed by an edge or set of edges. Faces are topologically linked to their surrounding edges as well as to the other faces that surround them. Faces are always non-overlapping, exhausting the area of a plane. [9]

fair fight. A condition when the differences between the performance characteristics of two or more

interoperating simulations have significantly less effect on the outcome of a simulated situation than the actions taken by or resources available to the simulation participants. See level playing field. [41]

fast time. 1. Simulated time with the property that a given period of actual time represents more than that period of time in the system being modeled (e.g., in a simulation of plant growth, running the simulation for one second may result in the model advancing time by one full day, i.e., simulated time advances faster than actual time). 2. The duration of activities within a simulation in which simulated time advances faster than actual time. See real time, slow time. [1, 2]

feature. A static element of the synthetic environment that exists but does not actively participate in synthetic environment interactions. Features are represented in the implementation environment by cartographic databases that are used by simulation assets. Entities can interact with features (building them, destroying them, colliding with them, etc.), but features are passive in that they do not initiate action. When features are dynamic (e.g., dynamic terrain) they are called environment entities. See environmental entity, synthetic environment. [1]

federate time. Scaled wall clock time or logical time of a federate, whichever is smaller. Federate time is synonymous with the "current time" of the federate. At any instant of an execution different federates will, in general, have different federate times. [7]

federate. A member of a High Level Architecture federation. All applications participating in a federation are called federates. This may include federation managers, data collectors, real world ("live") systems (e.g., C4I systems, instrumented ranges, sensors), simulations, passive viewers and other utilities. [7]

federation element. Term applied to an individual model and/or simulation that is part of a federation of models and simulations. [27]

federation execution data (FED). Information derived from the Federation Object Model (class, attribute, parameter names, etc.). Each federation execution needs one. In the abstract, creation of a federation execution is simply the binding of a federation execution name to a federation execution data. The organization of federation execution data will become the subject of standard so Federation Object Model tools can automatically generate them for any vendor's Runtime Infrastructure. [7]

federation execution. The actual operation, over time, of a subset of the federates and the Runtime Infrastructure initialization data taken from a particular

federation. It is the step where the executable code is run to conduct the exercise and produce the data for the measures of effectiveness for the federation execution. [7]

Federation Object Model (FOM). An identification of the essential classes of objects, object attributes, and object interactions that are supported by a High Level Architecture federation. In addition, optional classes of additional information may also be specified to achieve a more complete description of the federation structure and/or behavior. [7]

federation objective. The statement of the problem that is to be addressed by the establishment and execution of a federation. The description of the problem domain implicit in the objectives statement is critical for focusing the domain analysis activities in the conceptual analysis phase. It specifies the top level goals of the federation, and may specify the operational need or shortfall from which federation developers will derive a scenario for the federation execution. The federation objectives drive this specification, as the scenario development phase must utilize the statement of the objectives to generate a viable context for system evaluations intrinsic to the federation objectives. High-level testing requirements implied in the federation objectives may also drive the identification of well-defined "test points" during development of the federation scenario. [7]

federation time axis. A totally ordered sequence of values where each value represents an instant of time in the physical system being modeled, and for any two points T_1 and T_2 on the federation time axis, if $T_1 < T_2$, then T_1 represents an instant of physical time that occurs before the instant represented by T_2 . Logical time, scaled wall clock time, and federate time specify points on the federation time axis. The progression of a federate along the federation time axis during the execution may or may not have a direct relationship to the progression of wall clock time. [7]

federation time. The time used to coordinate the activities between federation members. Runtime Infrastructure services are specified in terms of federation time and are independent of the discipline used by federation members to advance to their individual temporal states. [7]

federation. A named set of interacting federates, a common federation object model, and supporting Runtime Infrastructure, that are used as a whole to achieve some specific objective. [7]

fidelity management. The process of monitoring and controlling the specification of fidelity characterizations

and fidelity quantification and of transforming fidelity characteristics from one stage to the next in the federation development and related verification, validation and accreditation processes. [41]

fidelity. 1. The degree to which a model or simulation reproduces the state and behavior of a real world object or the perception of a real world object, feature, condition, or chosen standard in a measurable or perceivable manner; a measure of the realism of a model or simulation; faithfulness. Fidelity should generally be described with respect to the measures, standards or perceptions used in assessing or stating it. See accuracy, sensitivity, precision, resolution, repeatability, model/simulation validation. 2. The methods, metrics, and descriptions of models or simulations used to compare those models or simulations to their real world referents or to other simulations in such terms as accuracy, scope, resolution, level of detail, level of abstraction and repeatability. Fidelity can characterize the representations of a model, a simulation, the data used by a simulation (e.g., input, characteristic or parametric), or an exercise. Each of these fidelity types has different implications for the applications that employ these representations. [41]

field instrumentation. An internal or external recording, monitoring, and relaying device employed by live instrumented entities, usually platform, facility, or exercise-unique, and not typically part of the operational system or equipment. These devices provide an independent source of data to assess the performance of operational systems involved in the exercise. [1]

field. A series of contiguous bits treated as an instance of a particular data type that may be part of a higher level data structure. [1, 13]

final condition. The values assumed by the variables in a component, system, model, or simulation at the completion of some specified duration of time; final state. See boundary condition, initial condition. [1, 2]

final state. A final condition. [41]

fitness. Providing the capabilities needed or being suitable for some purpose, function, situation or application. [44, 45]

formal language. In logic, a set of symbols together with a set of formation rules that designate certain sequences of symbols as well formed formulas, and a set of rules of inference (transformation rules) that, given a certain sequence of well formed formulas, permit the construction of another well formed formula. The symbols chosen vary from language to language, but typically they contain both logical constants and

nonlogical vocabulary, e.g., in the language of the propositional calculus the logical constants are truth-functional connectives and the nonlogical vocabulary consists solely of sentence letters, in the predicate calculus, variable, predicates and quantifiers are needed. The formation rules will naturally reflect the chosen vocabulary. The rules of inference are to be thought of as governing only the manipulation of symbols, independently of any interpretation they may have. Although formal languages do not require at any state the notion of an interpretation, they are nevertheless constructed with interpretations in mind, and rules of inference that do not preserve truth, although not formally unsatisfactory, are of no interest. [35]

formal system. A formal language together with a set of axioms. [35]

formation rules. In logic, the rules of a formal language for constructing well formed formulas from symbols. [35]

G

game. A physical or mental competition in which the participants, called players, seek to achieve some objective within a given set of rules. See game theory. [1, 2]

game theory. 1. The study of situations involving competing interests, modeled in terms of the strategies, probabilities, actions, gains, and losses of opposing players in a game. See management game, war game. 2. The study of games to determine the probability of winning given various strategies. [1, 2]

gateway. A device that connects two systems, especially if the systems use different protocols (e.g., a gateway is needed to connect two independent local networks, or to connect a local network to a long-haul network). [13]

generic domain. A domain type where the attribute is constrained only by the data type assigned by the database management system, or implied by the record type in a flat file, whichever is applicable. [12]

generic element. The part of a data element that establishes a structure and limits the allowable set of values of a data element. A generic element has no functional or application context other than to define a general class of data and ensure consistency in structure and domain. [15]

geodetic coordinate system. A measurement system that relates Earth-centered angular latitude and longitude (and optionally height) to an actual point near or on the earth's surface. [9]

geometry. 1. A very abstract class, encapsulating both the concepts of traditional geometry as well as other classes containing measured data and organizational methods used to organize these traditional geometry and other 'real' data classes within a synthetic environment. 2. A geometry primitive such as a point, line, or polygon, or an assembly of such primitives or assemblies.[9] 3. The mathematics of the properties, measurement, and relationships of points, lines, angles, surfaces, and solids. [4]

glass box model. A model whose internal implementation is known and fully visible (e.g., a model of a computerized change-return mechanism in a vending machine, in the form of a diagram of the circuits and gears that make the change); white box model. See black box model. [1, 2]

granularity. Resolution. [3]

graphical model. A symbolic model whose properties are expressed in diagrams (e.g., a decision tree used to express a complex procedure). See mathematical model, narrative model, software model, tabular model. [1, 2]

Greenwich Mean Time (GMT). A measure of time that conforms, within a close approximation, to the mean diurnal rotation of the Earth and serves as the basis of civil time-keeping. Universal time (UT1) is determined from observations of the stars, radio sources, and also from ranging observations of the Moon and artificial Earth satellites. The scale determined directly from such observations is designated Universal Time Observed (UTO); it is slightly dependent on the place of observation. When UTO is corrected for the shift in longitude of the observing station caused by polar motion, the time scale UT1 is obtained. When an accuracy better than one second is not required, Universal Time can be used to mean Coordinated Universal Time (UTC) Also called "Universal Time [Coordinated]" or "Zulu Time." [24]

ground truth. The actual facts of a situation, without errors introduced by sensors or human perception and judgement. See perceived truth, truth. [1, 3]

guise. A function that provides the capability for an entity to be viewed with one appearance by one group of participants, and with another appearance by another group. [1, 13]

H

happens before, causal (-->). A relationship between two actions A_1 and A_2 (where an action can be an event, an Runtime Infrastructure message send, or an Runtime Infrastructure message receive) defined as follows: a. if

A_1 and A_2 occur in the same federate/Runtime Infrastructure, and A_1 precedes A_2 in that federate/Runtime Infrastructure, then $A_1 \rightarrow A_2$; b. if A_1 is a message send action and A_2 is a receive action for the same message, then $A_1 \rightarrow A_2$; and c. if $A_1 \rightarrow A_2$ and $A_2 \rightarrow A_3$, then $A_1 \rightarrow A_3$ (transitivity). [7]

happens before, temporal (-->_t). A relationship between two events E_1 and E_2 defined as follows: if E_1 has a smaller time stamp than E_2 , the $E_1 \rightarrow_t E_2$. The Runtime Infrastructure provides an internal tie-breaking mechanism to ensure (in effect) that no two events observed by a single federate contain the same time stamp. [7]

heterogeneous. Consisting of or involving dissimilar elements or parts. [14]

heterogeneous network. A collection of simulations with partially consistent behaviors and/or partially correlated data bases (e.g., simulators of different fidelity, mixed virtual and live simulations, and mixes of virtual and constructive simulations). [1]

heuristic. Relating to or using a problem-solving technique in which the most appropriate solution of several found by alternative methods is selected at successive stages of a program for use in the next step of the program. [14]

hierarchical model. A model of information in which data are represented as trees of records connected by pointers. [12]

hierarchy. A ranking or ordering of abstractions. [25]

High Level Architecture (HLA). Major functional elements, interfaces, and design rules, pertaining as feasible to all DoD simulation applications, and providing a common framework within which specific system architectures can be defined. [13]

higher order model (HOM). A computer model representing combat elements, their functions and/or the terrain they operate on in an aggregated manner. A HOM may represent a battalion as a specific entity which is a conglomeration or averaging of the characteristics of its real-world components. "Higher Order" generally refers to echelons battalion and above with greater than 100m resolution (e.g. 3km, and with faster than real-time performance (e.g., days compressed into minutes, hours into seconds)). See war game. [1, 13]

homogeneous network. A network of DIS objects with fully consistent behaviors and fully correlated data bases. [1, 13]

host. A computer that supports one or more simulation applications; host computer. All host computers participating in a simulation exercise are connected by network(s) including wide area networks, local area networks, and RF links. [1, 2]

human-in-the-loop (HITL). A model that requires human interaction. See interactive model. [1]

human-machine simulation. A simulation carried out by both human participants and computers, typically with the human participants asked to make decisions and a computer performing processing based on those decisions. [1]

hybrid simulation. A simulation that combines constructive, live, and/or virtual simulations, typically in a distributed environment. Such simulations typically combine simulators with actual operational equipment, prototypes of future systems, and realistic representations of operational environments. [13]

I

iconic model. A physical model or graphical display that looks like the system being modeled (e.g., a non-functional replica of a computer tape drive used for display purposes). See scale model. [1, 2]

identity simulation. A simulation in which the roles of the participants are investigated or defined (e.g., a simulation that identifies aircraft based on their physical profiles, speed, altitude, and acoustic characteristics). [1]

imagined reality. A concept that has no exact counterpart in the material universe although parts of it may have counterparts in the material universe, e.g., a unicorn. Imagined reality may have a nonzero intersection with but can never be a proper subset of material reality. [41]

implementation. The means by which a synthetic environment, or portions of a synthetic environment, is realized. [1]

in-basket simulation. A simulation in which a set of issues is presented to a participant in the form of documents on which action must be taken (e.g., a simulation of an unfolding international crisis as a sequence of memos describing relevant events and outcomes of the participant's actions on previous memos). [1, 2]

independent time advancement. A means of advancing federate time where advances occur without explicit coordination among federates. Distributed

Interactive Simulation uses independent time advancement. [7]

information model. A model that represents the processes, entities, information flows, and elements of an organization and all relationships between these factors. [15]

information system (IS). The organized collection, processing, maintenance, transmission, and dissemination of information in accordance with defined procedures, whether automated or manual. [6, 15]

information. Any communication or reception of knowledge such as facts, data, or opinions, including numerical, graphic, or narrative forms, whether oral or maintained in any medium, including computerized databases, paper, microform, or magnetic tape. [6, 15, 26]

infrastructure. An underlying base or foundation; the basic facilities, equipment, and installations (e.g., systems and applications, communications, networks, architectures, standards and protocols, and information resource repositories) needed for the functioning of a system. [8, 10, 14]

inheritance. The object-oriented concept where a child class also has the features (attributes and methods) of its parent class; one of the types of relationships between objects in the data model. [9]

initial condition. The values assumed by the variables in a component, system, model, or simulation at the beginning of some specified duration of time; initial state. See boundary condition, final condition. [1]

initial state. An initial condition. [41]

input. 1. An event external to a system that modifies the system in any manner. 2. A variable at the boundary of an organism or machine through which information enters; the set of conditions, properties or states that effects a change in a system's behavior. [36] 3. Something introduced into a system or expended in its operation to attain a result or output. See output, data. [4] 4. The externally-supplied data to which a simulation responds and from which it calculates its output, e. g., operator controls, weapon detonation, wind speed and direction. 5. Observables in a natural system that are independent of other observables. See observables, natural system. [41]

instantiation. To represent an abstraction by a concrete instance. [14]

instructional simulation. A simulation intended to provide a simulation equivalent of a real or hypothesized stimulus that could occur in the synthetic environment for the purpose of training. [1]

intelligent agent. A software entity that carries out a set of operations on behalf of a user with some degree of independence or autonomy, and in so doing, employs knowledge or representation of the user's goals or desires. [14]

interaction parameters. The information associated with an interaction which objects potentially affected by the interaction must receive in order to calculate the effects of that interaction on its current state. [7]

interaction. 1. An explicit action taken by an object, that can optionally (within the bounds of the Federation Object Model) be directed toward other objects, including geographical areas etc. [7] 2. The way in which object, components, systems, models or simulations affect or influence each other. [41]

interactive model. A model that requires human participation; human-in-the-loop model. [1]

internal schema. An internal schema describes data as it is physically stored and includes all aspects of the environment in which a database is to reside. [6, 16]

interoperability. The ability of a set of models or simulations to provide services to and accept services from another models or simulations and to use the services so exchanged to enables them to operate effectively together. [3, 8]

interval-oriented simulation. A continuous simulation in which simulated time is advanced in increments of a size suitable to make implementation possible on a digital system. [1, 2]

J

K

knowledge. 1. The rules, environment, etc. that form the structure humans use to process and relate to information, or the information a computer system must have to behave in an apparently intelligent manner. [14] 2. The sum or range of what has been perceived, discovered or learned. [4]

knowledge-based system. A system in which the domain knowledge is explicit and separate from the system's operational instructions/information. [14]

known object. An object for which the federate is reflecting or updating any attributes. [7]

L

lag variable. 1. In a discrete simulation, a variable that is an output of one period and an input for some future period. 2. In an analog simulation, a variable that is a function of an output variable and that is used as input to the simulation to provide a time delay response or feedback. [1]

latency. 1. The observable delay between stimulus and response. 2. The time interval required by a simulation to respond to a stimulus in excess of the time interval required for the corresponding real world or standard event. 3. The time interval required for a device to begin output of data after presented with a stimulus or stimuli (e.g., input of data, occurrence of an event). [41]

lead variable. 1. In a discrete simulation, a variable that is an output of one period and that predicts what the output of some future period will be. 2. In an analog simulation, a variable that is a function of an output variable and that is used as input to the simulation to provide advanced time response or feedback. [1]

level of detail. Resolution. [41]

levels of topology. Level 0 topology manipulates the purely geometric aspects of the spatial data. No topological information is stored in level 0 topology. Level 1 topology maintains a non-planer graph. Level 2 topology maintains a planar graph. Level 3 topology explicitly represents the faces defined by the planer graph. [9]

linear network. A geographic entity that defines a linear (one-dimensional) structure (e.g., a river, a road or a state boundary). [9]

link association. A class containing the attributes of an association (link) between two other classes for which the attributes are properties of the association, not the classes linked by the association. [9]

littoral region. 1. From seaward, the area from the open oceans to the shore that must be controlled to support operations ashore. From landward, the area inland from the shore that can be supported and defended directly from the sea. [9]

live entity. A perceptible object that can appear in the virtual battlespace but is unaware and non-responsive (either by intent, lack of capability or circumstance) to the actions of virtual entities. See field instrumentation. [1]

live simulation. A simulation involving real people operating real systems. See virtual simulation, constructive simulation. [8]

local area network. A class of data network that provides high data rate interconnection between network nodes in close physical proximity. [28]

local time. The mean solar time for the meridian of the observer. [7]

logical data model. A model of the data stores and flows of the organization derived from the conceptual business model. [15]

logical time axis. A set of points (instants) on the federation time axis used to specify before and after relationships among events. [7]

logical time. A federate's current point on the logical time axis. If the federate's logical time is T , all time stamp ordered messages with time stamp less than T have been delivered to the federate, and no time stamp ordered messages with time stamp greater than T have been delivered; some, though not necessarily all, time stamp ordered messages with time stamp equal to T may also have been delivered. Logical time does not, in general, bear a direct relationship to wall clock time, and advances in logical time are controlled entirely by the federates and the Runtime Infrastructure. Specifically, the federate requests advances in logical time via the Time Advance Request and Next Event Request Runtime Infrastructure services, and the Runtime Infrastructure notifies the federate when it has advanced logical time explicitly through the Time Advance Grant service, or implicitly by the time stamp of time stamp ordered messages that are delivered to the federate. Logical time (along with scaled wall clock time) is used to determine the current time of the federate (see definition of federate time). Logical time is only relevant to federates using time stamp ordered message delivery and coordinated time advances, and may be ignored (by requesting a time advance to "infinity" at the beginning of the execution) by other federates. [7]

logical verification. The process of identifying a set of assumptions and interactions for which a model or simulation correctly produces the results intended by its developers. [29]

long-haul network (LHN). A communications network of devices which are separated by substantial geographical distance. A LHN could be any of numerous networks available commercially or through the government that can accommodate the requirements of the DIS virtual battlefield for long distance network services; wide area network. [1, 13]

lookahead. A value used to determine the smallest time stamped message using the time stamp ordered service that a federate may generate in the future. If a federate's current time (i.e., federate time) is T , and its lookahead is L , any message generated by the federate must have a time stamp of at least $T+L$. In general, lookahead may be associated with an entire federate (as in the example just described), or at a finer level of detail e.g., from one federate to another, or for a specific attribute. Any federate using the time stamp ordered message delivery service must specify a lookahead value. [7]

lower bound on the time stamp (LBTS). Lower Bound on the Time Stamp of the next time stamp ordered message to be received by a Runtime Infrastructure from another federate. Messages with time stamp less than LBTS are eligible for delivery by the runtime infrastructure to the federate without compromising time stamp order delivery guarantees. Time stamped ordered messages with time stamp greater than LBTS are not yet eligible for delivery. LBTS is maintained within the runtime infrastructure using a conservative synchronization protocol. [7]

M

machine simulation. A simulation that is executed on a machine. See computer simulation. [1, 2]

management game. A simulation game in which participants seek to achieve a specified management objective given pre-established resources and constraints (e.g., a simulation in which participants make decisions designed to maximize profit in a given business situation and a computer determines the results of those decisions). See war game. [1, 2]

Markov chain model. A discrete, stochastic model in which the probability that the model is in a given state at a certain time depends only on the value of the immediately preceding state; Markov model. See semi-Markov model. [1, 2]

Markov chain. A discrete Markov process. [2]

Markov process. A stochastic process that assumes that in a series of random events, the probability for occurrence of each event depends only on the immediately preceding outcome. See semi-Markov process. [1, 2]

mass storage. Any device that can store large amounts of data and retrieve it at some later time, even after system power-down. Mass storage devices are usually categorized in terms of being either on-line storage or off-line storage. [14]

material reality. The material universe (or those parts of it) that are pertinent to an application domain. [41]

mathematical model. 1. Any system of assumptions, definitions and equations that represents particular physical phenomena. See model, simulation, conceptual model, software model. [35] 2. A document describing the assumptions, definitions and equations that represent particular physical phenomena to be simulated for a specific application. [41]

mean solar time. A time measurement where time is measured by the diurnal motion of a fictitious body (called "mean Sun") which is supposed to move uniformly in the celestial Equator, completing the circuit in one tropical year. Often termed simply "mean time." The mean Sun may be considered as moving in the celestial Equator and having a right ascension equal to the mean celestial longitude of the true Sun. At any given instant, mean solar time is the hour angle of the mean Sun. In civil life, mean solar time is counted from the two branches of the meridian through 12 hours; the hours from the lower branch are marked a.m. (ante meridian), and those from the upper branch, p.m. (post meridian). In astronomical work, mean solar time is counted from the lower branch of the meridian through 24 hours. Naming the meridian of reference is essential to the complete identification of time. The Greenwich meridian is the reference for a worldwide standard of mean solar time called "Greenwich Mean Time" (GMT) or "Universal Time [Coordinated]" (UTC). [7]

measure of effectiveness (MOE). A qualitative or quantitative measure of the performance of a model or simulation or a characteristic that indicates the degree to which it performs the task or meets an operational objective or requirement under specified conditions. [14]

measure of outcome (MOO). A metric that defines how operational requirements contribute to end results at higher levels, such as campaign or national strategic outcomes. [10]

measure of performance (MOP). A measure of how the system/individual performs its functions in a given environment (e.g., number of targets detected, reaction time, number of targets nominated, susceptibility of deception, task completion time). It is closely related to inherent parameters (physical and structural) but measures attributes of system behavior. See measure of effectiveness. [1, 2]

message (event) delivery. Invocation of the corresponding service (Reflect Attribute Values, Receive Interaction, Instantiate Discovered Object, or Remove Object) by the Runtime Infrastructure to notify a federate of the occurrence of an event. [7]

message. A data unit transmitted between federates containing at most one event. Here, a message typically contains information concerning an event, and is used to notify another federate that the event has occurred. When containing such event information, the message's time stamp is defined as the time stamp of the event to which it corresponds. Here, a "message" corresponds to a single event, however the physical transport media may include several such messages in a single "physical message" that is transmitted through the network. [7]

metadata. Information describing the characteristics of data; data or information about meaning of the data; descriptive information about an organization's data, data activities, systems, and holdings. [6, 9, 15, 21, 22]

meta-knowledge. Knowledge about knowledge; knowledge about the use and control of domain knowledge in an expert or knowledge-based system or knowledge about how the system operates or reasons; wisdom. [13]

metamodel. A model of a model. Metamodels are abstractions of the M&S being developed which use functional decomposition to show relationships, paths of data and algorithms, ordering, and interactions between model components and subcomponents. Metamodels allow the software engineers who are developing the model to abstract details to a level that subject matter experts can validate. [13]

methodology. The system of principles, practices, and procedures, applied to a specific branch of knowledge. [14]

metric. 1. A measure of the extent or degree to which a product possesses and exhibits a certain quality, property, or attribute. [2] 2. A process or algorithm that may involve statistical sampling, mathematical computations, and rule-based inferencing. Metrics provide the capability to detect and report defects within a sample. [5]

mission space model. A model based primarily upon knowledge of the real world. Such a model, if based entirely upon expert opinion of the real world, is a preliminary to creating a mathematical or software model. A mission space model of an object should describe what that object does, at some level of fidelity, in the environment in which the mission is executed. See model, mathematical model, software model, mission space. [41]

mission space. 1. The battlespace in which a particular mission is performed. [41] 2. The environment of entities, actions, and interactions comprising the set of

interrelated processes used by individuals and/organizations to accomplish assigned tasks. [8]

mock-up. A full-sized structural, but not necessarily functional, model built accurately to scale, used chiefly for study, testing, or display. See physical model. [1, 2]

model. 1. A physical, mathematical, or otherwise logical abstract representation of a system, entity, phenomenon, or process with its own assumptions, limitations and approximations. See simulation, conceptual model, software model, mathematical model. [1, 8, 13, 40] 2. A geometry or feature assembly built in a relative coordinate system with the intent to multiply instances of the assembly at one or more world coordinate positions. [9] 3. A system that stands for or represents another typically more comprehensive system. [36]

modeling and simulation (M&S). The use of models, including emulators, prototypes, simulators, and stimulators, either statically or over time, to develop data as a basis for making managerial or technical decisions. The terms "modeling" and "simulation" are often used interchangeably. [13]

modeling. Application of a standard, rigorous, structured methodology to create and validate a physical, mathematical, or otherwise logical representation of a system, entity, phenomenon, or process. [10]

model-test-model. An integrated approach to using models and simulations in support of pre-test analysis and planning; conducting the actual test and collecting data; and post-test analysis of test results along with further validation of the models using the test data. [10]

modifier. A word that helps define and render a name unique within the database, which is not the prime or class word. [15]

Monte Carlo algorithm. A statistical procedure that determines the occurrence of probabilistic events or values of probabilistic variables for deterministic models (e.g., making a random draw). [10]

Monte Carlo method. In modeling and simulation, any method that employs Monte Carlo simulation to determine estimates for unknown values in a deterministic problem. [1, 2]

Monte Carlo simulation. A simulation in which random statistical sampling techniques are employed such that the result determines estimates for unknown values. [1]

multicast. A transmission mode in which a single message is sent to selected multiple (but not necessarily

all) network destinations (i.e., one-to-many). See broadcast, unicast. [1, 2]

multi-state objects. Mission space entities that express a changing state (in attribution and visual display) as the simulation progresses (e.g., damage to structures, changes in vegetation, damage system representations such as vehicles, tanks, etc). [8]

N

narrative model. A symbolic model the properties of which are expressed in words (e.g., a written specification for a computer system); verbal descriptive model. See graphical model, mathematical model, software model, tabular model. [1, 2]

natural environment. The Earth-based environment modeled by a synthetic environment. [9]

natural model. A model that represents a system by another system that already exists in the real world (e.g., a model that uses one body of water to represent another). [1, 2]

natural system. An abstract state space that couples with a set of finite observables that is a subset of real world systems, processes, or phenomena being modeled or simulated. [41]

network byte order. The Internet-standard ordering of the bytes corresponding to numeric values. [13]

network communication services. The capability provided to electronically transmit modeling and simulation data between networked computational nodes in a manner that meets requirements for transmission latency, multi-cast addressing and security needed to support the creation and operation of distributed time and space coherent synthetic environments. [23]

network filter. A system to selectively accept or reject data received from the network. [1]

network node. A specific network address; node. See processing node. [1]

network theory. The study of networks used to model processes such as communications, computer performance, routing problems, and project management. [1, 2]

node. 1. A general term denoting either a switching element in a network or a host computer attached to a network. See processing node, network node. [1, 2] 2. A zero-dimensional primitive used to store a significant location. [9]

non-absorbing state. In a Markov chain model, a state that can be left once it is entered. [1, 2]

non-standard cell. A cell that is not compliant with the Distributed Interactive Simulation message and data base standards. Non-standard cells require a Cell Adapter Unit in order to join a Distributed Interactive Simulation exercise. [1, 13]

non-standard data element. Any data element that exists in a system or application program and does not conform to the conventions, procedures, or guidelines established by the organization. [15]

normative model. A model that makes use of a familiar situation to represent a less familiar one (e.g., a model that depicts the human cardiovascular system by using a mechanical pump, rubber hoses, and water). [1, 2]

notional data. Speculative or theoretical data rather than actual data. [3]

numerical model. 1. A mathematical model in which a set of mathematical operations is reduced to a form suitable for solution by simpler methods such as numerical analysis or automation (e.g., a model in which a single equation representing a nation's economy is replaced by a large set of simple averages based on empirical observations of inflation rate, unemployment rate, gross national product, and other indicators). 2. A model whose properties are expressed by numbers. [1, 2]

O

object. A fundamental element of a conceptual representation for a federate that reflects the "real world" at levels of abstraction and resolution appropriate for federate interoperability. For any given value of time, the state of an object is defined as the enumeration of all its attribute values. [7]

object-based methodology. A software design methodology adhering to only some of the properties of object-oriented software methodologies (e.g., Ada does not support inheritance, a key property of object oriented systems, therefore Ada is often referred to as an object-based language). See object-oriented. [14]

object model. A specification of the objects intrinsic to a given system, including a description of the object characteristics, or attributes, and a description of the static and dynamic relationships that exist between objects. [7]

Object Model Framework. The rules and terminology used to describe High Level Architecture object models. [7]

object ownership. Ownership of the identification attribute of an object, initially established by use of the Instantiate Object interface service, and encompassing the privilege of deleting the object using the Delete Object service. Ownership can be transferred to another federate using the attribute ownership management services. [7]

object-oriented language. A computer programming language that best suits an object-oriented description of software and that provides the capability to implement classes and objects, to directly support data abstraction and classes, and to provide additional support for inheritance as a means of expressing hierarchies of classes. [10]

object-oriented methodology. A software design methodology that results in the battlefield being represented by objects, where objects encapsulate the methods or procedures associated with the object and where objects communicate with other objects by message passing. Examples of battlefield objects include platoons (unit level), tanks (platform level), main guns (component or module level), and gun barrels (part level). One of the main benefits of an object-oriented methodology is the inherent modularity (e.g., to change a tank model only the tank object must be changed). See object-based methodology. [30]

object-oriented programming. Use of a programming system that results in computer programs organized as cooperative collections of objects, each of which represents an instance of some class, and whose classes are members of class hierarchies as defined by the inheritance mechanism. [25]

observable. 1. Capable of being observed systematically or scientifically; discernible. 2. A physical property, such as temperature or weight, that can be observed or measured directly. [4] 3. A state variable, computable by a function or functions, or mathematical relation(s). 4. A function that maps a subset of the real world states into the set of real numbers. See natural system. [41]

oceanographic representation. Data describing the ocean, its objects, their attributes and the dependencies between the values of those attributes including descriptions of the ocean bottom (e.g., depth curves and bottom contours) as well as processes required to model the natural and man-made changing surface (e.g., sea state) and sub-surface (e.g., temperature, pressure, salinity gradients, acoustic phenomena) conditions. [9]

occlusion. 1. The effect of closer objects overlapping or obscuring more distant ones thus preventing the observation of parts or all of the distant objects and so providing clues to judge how close objects are from the

viewer. [31] 2. The process of preventing the passage of something; obstruction. [4]

octet. A sequence of eight bits, usually manipulated as a unit. [13]

off-line storage devices. Devices generally used for data backup and archival applications that employ media-like magnetic tapes or removable hard or floppy disks. [14]

on-line storage devices. Devices providing more immediate retrieval of data and usually including such devices as magnetic or optical hard disk drives. [14]

operational environment. A composite of the conditions, circumstances, and influences that affect the employment of military forces and the decisions of the unit commander. Frequently characterized as permissive, semi-permissive, or non-permissive. [1]

optimistic synchronization. A mechanism that uses a recovery mechanism to erase the effects of out-of-order event processing (e.g., the Time Warp protocol). Messages sent by an optimistic federate that could later be canceled. See conservative synchronization. [1]

original data. The source data used by a resource producer to construct their initial synthetic environment representation. [9]

orthogonal. 1. Pertaining to or composed of right angles. 2. Mutually independent mathematically (e.g., orthogonal variables). [14]

outcome-oriented simulation. A simulation in which the end result is considered more important than the process by which it is obtained (e.g., a simulation of a radar system that uses methods far different from those used by the actual radar, but whose output is the same). See process-oriented simulation. [1, 2]

output. 1. Any change produced in the surroundings by a system. 2. A variable at the boundary of an organism or machine through which information exits; the products, results or the observable parts of system behavior. [36] 3. The data produced by a computer from a specific input. See input, data. [4] 4. The aspects of the simulated system being modeled; calculated during each pass in response to inputs and time passing, normally output for external use; values providing a snap-shot of the current state of the simulated system, e.g., position, velocity, alive-or-dead. 5. An observables in a natural system that depends on some other observables. See observables, natural system. [41]

output validation. The process of determining the extent to which the output (outcome distributions for the models, simulations and/or sub-models) represent the significant and salient features of distributions or real world systems, events, and scenarios. [29]

owned attribute. An object attribute that is explicitly modeled by the owning federate. A federate that owns an attribute has the unique responsibility to provide values for that attribute to the federation, through the Runtime Infrastructure, as they are produced. [7]

P

parallax. 1. The vision effect of having two eyes viewing the same scene from slightly different positions that creates a sense of depth. Computer-generated environments, one for each eye, can artificially create the parallax effect. [31] 2. An apparent change in the direction of an object, caused by a change in the observational position that provides a new line of sight. [4]

parallel processing. Multiple processes running on multiple processors simultaneously. [10]

parameter. 1. A constant or variable that distinguishes special cases of a general mathematical expression, e.g., the general form of the equation for a line, $y = mx + b$ contains the parameters m and b , representing the gradient and y -intercept of any specific line. [35] 2. A constant in a mathematical program, not subject to choice in the decision problem, but one that could vary outside the control of the decisions. [37] 3. That which determines the structure of a system. Parameters themselves can be changed by inputs, but usually the parameters determine how input will be transformed into outputs. [36] 4. Observables in a natural system that remain constant for every state. [41]

parametric model. A model using parametric equations that may be based on numerical model outputs or fits to semi-empirical data to succinctly describe a particular process, feature, or effect. [13]

perceived truth. That subset of ground truth acquired or distorted by sensors, human perception or judgement; the situation as perceived by an observer. See ground truth, perception, truth. [41]

perception. 1. An observer's awareness or appreciation of objects, processes or situations in his environment mediated through their sensory organs. 2. An observer's descriptions, hypotheses or constructs of the world of which they become thereby a part. [36] 3. To take notice of; observe. [4]

period. The time interval between successive events in a discrete simulation. [1, 2]

Petri net. An abstract, formal model of information flow, showing static and dynamic properties of a system (i.e., the Petri net is defined by its places, transitions, input function, and output function). [1, 2]

physical data model. A representation of the technologically independent information requirements in a physical environment of hardware, software, and network configurations representing them in the constraints of an existing physical environment. [6, 16]

physical model. A model whose physical characteristics resemble the physical characteristics of the system being modeled (e.g., a plastic or wooden replica of an airplane); a mock-up. See iconic model, scale model, symbolic model. [1, 2]

pixel. A "picture element," referring to the smallest visual unit in an image on a computer display. [31]

platform. A generic term describing a level of representation equating to vehicles, aircraft, missiles, ships, fixed sites, etc., in the hierarchy of representation possibilities. Other representation levels include units (made up of platforms) and components or modules (which make up platforms). [1, 13]

point feature. A geographic entity defining a zero-dimensional location (e.g., a well or a building). [9]

polygon. 1. A flat plane figure with multiple sides, the basic building block of virtual worlds. The more polygons a computer can display and manipulate per second, the more realistic the virtual world will appear. Humans perceive the equivalent of 80 million polygons at more than 30 frames per second in normal vision. [31] 2. Thematically homogenous areas composed of one or more faces. [9]

positional accuracy. The root mean square error (RMSE) of the coordinates relative to the position of a real world entity being modeled. Positional accuracy shall be specified without relation to scale and shall contain all errors introduced by source documents, data capture and data processing. [9]

precision. 1. The quality or state of being clearly depicted, definite, measured or calculated. 2. A quality associated with the spread of data obtained in repetitions

of an experiment as measured by variance; the lower the variance, the higher the precision. [35] 3. A measure of how meticulously or rigorously computational processes

are described or performed by a model or simulation. See resolution, sensitivity. [41]

predictive model. A model in which the values of future states can be predicted or are hypothesized (e.g., a model that predicts weather patterns based on the current value of temperature, humidity, wind speed, and so on at various locations). [1, 2]

prescriptive model. A model used to convey the required behavior or properties of a proposed system (e.g., a scale model or written specification used to convey to a computer supplier the physical and performance characteristics of a required computer). See descriptive model. [1, 2]

prime word. A word included in the name of a data entity that represents the logical data grouping (in the logical data model) to which it belongs. [15]

probabilistic model. See stochastic model. [1]

process improvement modeling. Defines and documents the current ("as is") and desired future ("to be") processes and information requirements of a functional activity. See activity model, data model. [14]

process model. A model of the processes performed by a system (e.g., a model that represents the software development process as a sequence of phases). See structural model. [1]

process. 1. Something that affects entities (e.g., attrition, communications, and movement). Processes have a level of detail by which they are described. [18] 2. A system of operations in producing something. 3. A series of actions, changes, or functions that achieve an end or result. [4]

processing node. The hardware and software processing resources devoted to one or more simulation entities. See node, network node. [1]

process-oriented simulation. A simulation in which the process is considered more important than the outcome (e.g., a model of a radar system in which the objective is to replicate exactly the radar's operation, and duplication of its results is a lesser concern). See outcome-oriented simulation. [1, 2]

projected coordinate system. An instantiation of a coordinate transformation; a planar, right-handed Cartesian coordinate set which, for a specific map projection, has a single and unambiguous transformation to a geodetic coordinate system. [9]

Protocol Data Unit (PDU). Distributed Interactive Simulation terminology for a unit of data that is passed on a network between simulation applications. [8]

protocol entity. An object that exchanges information with other protocol entities in a network via Protocol Data Units in accordance with an established protocol. A key attribute of a protocol entity is its state. State transitions occur in a given protocol entity in accordance with the established protocol as the result of: a. Protocol Data Units received from other protocol entities, and b. occurrence of an external event (e.g., expiration of a time-out counter.) See Protocol Data Unit. [1]

protocol suite. A defined set of complementary protocols within the communication architecture profile. [13]

protocol. A set of rules and formats (semantic and syntactic) that define the communication behavior of simulation applications. [1, 2]

prototype. A preliminary type, form, or instance of a system that serves as a model for later stages or for the final, complete version of the system. [1, 2]

pseudocode. A description of control and/or data structures in a natural language with no rigid rules of syntax. [29]

purpose. The objective for which a simulation or simulation exercise is intended; goal. See application. [41]

Q

qualitative data. A non-numeric description of a person, place, thing, event, activity, or concept. [15]

quantitative data. Numerical expressions that use Arabic numbers, upon which mathematical operations can be performed. [15]

queue. A set of zero or more entities waiting to be serviced by a service facility. [1, 2]

queuing model. A model consisting of service facilities and entities waiting in queues to be served (e.g., a model depicting teller windows and customers at a bank). See queuing theory. [1, 2]

queuing network model. A model in which a process is described as a network in which each node represents a service facility rendering a given type of service and a queue for holding entities waiting to be served (e.g., a model depicting a network of shipping routes and

docking facilities at which ships must form queues in order to unload their cargo). [1, 2]

queuing theory. The study of queues and the performance of systems that service entities that are organized into queues. See queuing model, queuing network model. [1, 2]

R

random. Pertaining to a process or variable whose outcome or value depends on chance or on a process that simulates chance, often with the implication that all possible outcomes or values have an equal probability of occurrence (e.g., the outcome of flipping a coin or executing a computer-programmed random number generator). [1, 2]

real battlefield. See real-world. [1]

reality. The quality or state of being actual or true. [4]

real-time service. A service that satisfies timing constraints imposed by the service user. The timing constraints are user specific and should be such that the user will not be adversely affected by delays within the constraints. [13]

real-time simulation. See constrained simulation. [7]

real-time system. A system that computes its results as quickly as they are needed by a real-world system. Such a system responds quickly enough that there is no perceptible delay to the human observer. In general use, the term is often perverted to mean within the patience and tolerance of a human user. [14]

real-time. In modeling and simulation, simulated time advances at the same rate as actual time (e.g., running the simulation for one second results in the model advancing time by one second). See fast time, slow time. [1]

real-world time. The actual time in Greenwich, Great Britain; sidereal time. [1, 2]

real-world. The set of real or hypothetical causes and effects that simulation technology attempts to replicate. See real battlefield. [1] The real world defines one standard against which fidelity is measured that includes both imagined reality and material reality in order to accommodate assessment of simulation fidelity when future concepts and systems are involved. See fidelity, imagined reality, material reality, perceived truth. [41]

referent. 1. A codified body of knowledge about a thing being simulated. [41] 2. Something referenced or

singled out for attention, a designated object, real or imaginary or any class of such objects. [4, 36]

reflected attribute. An object attribute that is represented but not explicitly modeled in a federate. The reflecting federate accepts new values of the reflected attribute as they are produced by some other federation member and provided to it by the Runtime Infrastructure. [7]

reflected object. An object that is represented but not explicitly modeled in a simulation. The reflecting simulation accepts changes in state of the reflected object as they are produced by some other federation member and provided to it by the Runtime Infrastructure. [14]

regime. The interaction domain of entities. [14]

reliability model. A model used to estimate, measure, or predict the reliability of a system (e.g., a model of a computer system, used to estimate the total down time that will be experienced). [1, 2]

reliable service. A communication service in which the received data is guaranteed to be exactly as transmitted. [1, 2, 13]

remote entity approximation (REA). The process of extrapolating and interpolating any state of an entity based on its last known state including dead reckoning and smoothing. See dead reckoning. [1]

repeatability. A measure of the ability to be done again and again. [4]

representation. 1. Something that stands in place of or is chosen to substitute for something else, e.g., representation of constituencies in government, linguistic representation of an event. [36] 2. Something that describes as an embodiment of a specified quality. [4] 3. The homomorphism of a group of abstract symbols into a group of more familiar objects. [35] 4. A model or simulation. [41]

representational polymorphism. Multiple representations of the same data to serve the needs of different users. [9]

resolution. 1. The degree of detail used to represent aspects of the real world or a specified standard or referent by a model or simulation. 2. Separation or reduction of something into its constituent parts; granularity. [4]

retraction. An action performed by a federate to un-schedule a previously scheduled event. Event retraction is visible to the federate. Unlike "cancellation"

that is only relevant to optimistic federates such as Time Warp, "retraction" is a facility provided to the federate. Retraction is widely used in classical event oriented discrete event simulations to model behaviors such as preemption and interrupts. [7]

Right-Hand Rule. Positive rotation is clockwise when viewed toward the positive direction along the axis of rotation. [1]

Runtime Infrastructure (RTI). The general purpose distributed operating system software which provides the common interface services during the runtime of an High Level Architecture federation. [14]

S

scalability. The ability of a distributed simulation to maintain time and spatial consistency as the number of entities and accompanying interactions increase. [8]

scale model. A physical model that resembles a given system, with only a change in scale (e.g., a replica of an airplane one tenth the size of the actual airplane). [1, 2]

scaled wall clock time. A quantity derived from a wall clock time defined as $\text{offset} + [\text{rate} * (\text{wall clock time} - \text{time of last exercise start or restart})]$. All scaled wall clock time values represent points on the federation time axis. If the "rate" factor is k, scaled wall clock time advances at a rate that is k time faster than wall clock time. [7]

scenario development. A phase of the development of a federation during which the federation developer(s) formulate a scenario whose execution and subsequent evaluation will lead toward achieving the study objectives set forth by the federation sponsor. The output of this phase is a functional-level scenario description, which is provided as input to the Conceptual Analysis phase. Certain key activities during Conceptual Analysis may also drive reiterations of the Scenario Development phase. [7]

scenario. 1. Description of an exercise. It is part of the session database that configures the units and platforms and places them in specific locations with specific missions. 2.. An initial set of conditions and time line of significant events imposed on trainees or systems to achieve exercise objectives. [1, 2] 3. An identification of the major entities that must be represented by the federation, a conceptual description of the capabilities, behavior, and relationships (interactions) between these major entities over time, and a specification of relevant environmental conditions (e.g., terrain, atmospheric). Initial and termination conditions are also provided. The style of format of the scenario documentation (e.g.,

graphics, tables, text) is entirely at the discretion of the federation developer. However, communities of use may wish to establish scenario documentation standards among themselves to facilitate reuse of scenario components. [7] 4. A part of the modeling and simulation database that contains the force structure, its mission and plans, and the terrain area in which the simulated engagement occurs. [32]

scheduling an event. Invocation of a primitive (Update Attribute Values, Send Interaction, Instantiate Object, or Delete Object) by a federate to notify the Runtime Infrastructure of the occurrence of an event. Scheduling an event normally results in the Runtime Infrastructure sending messages to other federates to notify them of the occurrence of the event. [7]

schema. Descriptive representation of data and/or data requirements that describe conceptual, internal, or external views of information/data needs. [14]

scope. The range of real or imagined world objects or conditions represented by a particular model, simulation or simulation exercise. See real-world, imagined reality, material reality. [41]

seamless. 1. Perfectly consistent. 2. Transparent. [3]

segment. A portion of a session that is contiguous in simulation time and in wall-clock time (sidereal time). [1, 2]

selector. A portion of an address identifying a particular entity at an address (e.g., a session selector identifies a user of the session service residing at a particular session address). [13]

semantics. 1. The implied meaning of data to define what entities mean with respect to their roles in a system. [9] 2. The study of relationships between signs and symbols and what they represent to their interpreters. [4]

Semi-Automated Forces (SAFOR). Simulation of friendly, enemy and neutral platforms on the virtual battlefield in which the individual platform simulation are operated by computer simulation of the platform crew and command hierarchy. The term "semi-automated" implies that the automation is controlled and monitored by a human who injects command-level decision making into the automated command process. See also: computer-generated forces. [10]

semi-Markov model. A Markov chain model in which the length of time spent in each state is randomly distributed. [1, 2]

semi-Markov process. A Markov process in which the duration of each event is randomly distributed. [1, 2]

sensitivity. The ability of a component, model or simulation to respond to a low level stimulus. [47]

session. A portion of an exercise that is contiguous in wall-clock (sidereal) time and that is initialized per an exercise database. [1, 2]

sidereal time. 1. Time based upon the rotation of the Earth relative to the vernal equinox. 2. Time that is independent of simulation clocks, time zones, or measurement errors; the ground truth of time measurement. See real world time. [33]

simuland. The system being simulated by a simulation. See referent, model, simulation. [1]

simulated time. Time as represented within a simulation; virtual time. See fast time, real time, slow time. [2]

simulation application. 1. The executable software on a host computer that models all or part of the representation of one or more simulation entities, e.g., manned vehicle simulators, computer generated forces, environment simulators, and computer interfaces between a simulation network and actual equipment. More than one simulation application may execute simultaneously on a single host computer. See simulation, application, software model. 2. The application layer protocol entity that implements standard Distributed Interactive Simulation protocol. [1, 2]

simulation clock. A counter used to accumulate simulated time. [1, 2]

simulation entity. An element of the synthetic environment that is created and controlled by a simulation application, e.g., tanks, submarines, carriers, fighter aircraft, missiles, bridges. A simulation application may control more than one simulation entity. [1, 2]

simulation environment. 1. The operational environment surrounding the simulation entities including terrain, atmospheric, bathyspheric and cultural information. 2. All conditions, circumstances, and influences surrounding and affecting simulation entities including those stated in 1. [1] 3. An entire simulation framework including software, hardware, architecture, infrastructure and interfaces where models or simulations are developed and executed. [41]

simulation execution. The real-time execution of a simulation application. See exercise, simulation application. [41]

simulation game. A simulation in which the participants seek to achieve some agreed-upon objective within an established set of rules (e.g., a management game, a war game); gaming simulation. The objective may not be to compete, but to evaluate the participants, increase their knowledge concerning the simulated scenario, or achieve other goals. [1, 2]

simulation management. A mechanism that provides centralized control of the simulation exercise including start, restart, maintenance, shutdown of the exercise, and collection and distribution of certain types of data. [1, 2]

simulation model. A digital or physical realization of a conceptual model. A digital realization is a software implementation of a part or all of a conceptual model in a specific programming language based on some software design methodology; software model. A physical realization is a hardware implementation of part or all of a conceptual model, e.g., the layout of instrument panel in a mock-up or motion platform. [41]

Simulation Object Model (SOM). A specification of the intrinsic capabilities that an individual simulation offers to federations. The standard format in which SOMs are expressed provides a means for federation developers to quickly determine the suitability of simulation systems to assume specific roles within a federation. [7]

simulation process. The imitative representation of the actions of platform(s), munitions(s), and life form(s) by computer program(s) in accordance with a mathematical model and the generation of associated battlefield entities that may be fully automated or partially automated. [1]

simulation support entity. Processing modules used to support, control, or monitor the simulation environment, but which do not actually exist on the battlefield including battlefield viewing devices for controllers or exercise observers such as the stealth vehicle, the plan view display, after action review systems, and simulation control systems. [1, 13]

simulation time. 1. A simulation's internal representation of time which may accumulate faster, slower, or at the same pace as sidereal time. 2. The reference time (e.g., Universal Coordinated Time) within a simulation exercise, this time is established by the simulation management function before the start of the simulation and is common to all participants in a particular exercise. [1, 2]

simulation. 1. A method, software framework or system for implementing one or more models in the proper order to determine how key properties of the original may change over time. See model, representation. [41] 2. An unobtrusive scientific method of inquiry involving experiments with a model rather than with the portion of reality this model represents. [36]

simulator. A device or physical system that implements or performs simulation. See simulation, simuland, software model, mathematical model. [41]

six degrees of freedom (6 DOF). The number of simultaneous directions or inputs a sensor can measure typically used to describe the combination of spatial positions (X, Y, Z) and orientation (roll, pitch, yaw). [31]

slow time. The duration of activities within a simulation in which simulated time advances slower than actual time. [1]

smoothing. Interpolation of the previous state of an entity (location, velocity, etc.) to the current state, creating a smoothed transition between two successive entity state updates. [1]

software model. The actual compilable and linkable software source code that implements algorithms and data flow representing one or more mathematical models; simulation model. See model, conceptual model, mathematical model, simulation model. [41]

space representation. Representation of the regions beyond the upper boundary of the troposphere (including ionosphere) including data on neutral and charged atomic and molecular particles (including their optical properties). [9]

span. The scale of the domain that is global, theater, regional, local, individual. Description of the span is often subjective. [14]

sponsor. An individual, agency or business that pays for the development, modification or execution of a model, simulation or simulation exercise. See user. [41]

stability. 1. Constancy of purpose; steadfastness. 2. Reliability; dependability. [5] 3. Resistance to sudden change, dislodgment or overthrow. [4]

stabilized-variable model. A model in which some of the variables are held constant and the others are allowed to vary (e.g., a model of a controlled climate in which humidity is held constant and temperature is allowed to vary). [1, 2]

standard. 1. An accepted measure of comparison for quantitative or qualitative value; a criterion. [4] 2. Proposition of a norm or general pattern to be followed when constructing, operating or testing a (technical) device. A standard contains a set of reference criteria for functional, structural, performance or quality aspects of a device or for any combination of these. [36]

state transition. A change from one state to another in a system, component, or simulation. [1]

state variable. A variable that defines one of the characteristics of a system, component, or simulation where the values of all such variables define the state of the system, component, or simulation. [1]

state. 1. The internal status of a simulation entity (e.g. fuel level, number of rounds remaining, location of craters). 2. A condition or mode of existence in which a system, component, or simulation (e.g., the pre-flight state of an aircraft navigation program or the input state of given channel). 3. The values assumed at a given instant by the variables that define the characteristics of a system, component, or simulation; system state. See final state, initial state, steady state. [1]

static model. A model of a system in which there is no change (e.g., a scale model of a bridge, studied for its appearance rather than for its performance under varying loads). [1, 2]

steady state. A situation in which a model, process, or device exhibits stable behavior independent of time. [1, 2]

stimulate. To provide input to a system in order to observe or evaluate the system's response. [1, 2]

stimulation. The use of simulations to provide an external stimulus to a system or subsystem (e.g., using a simulation representing the radar return from a target to drive (stimulate) the radar of a missile system within a hardware/software-in-the-loop simulation). [10]

stimulator. 1. A hardware device that injects or radiates signals into the sensor system(s) of operational equipment to imitate the effects of platforms, munitions, and environment that are not physically present. 2. A battlefield entity consisting of hardware and/or software modules that injects signals directly into the sensor systems of an actual battlefield entity to simulate other battlefield entities in the virtual battlefield. [1]

stochastic model. A model in which the results are determined by using one or more random variables to represent uncertainty about a process or in which a given input will produce an output according to some statistical

distribution (e.g., a model that estimates the total dollars spent at each of the checkout stations in a supermarket, based on probable number of customers and probable purchase amount of each customer); probabilistic model. See Markov-chain model, deterministic model. [1]

stochastic process. Any process dealing with events that develop in time or cannot be described precisely, except in terms of probability theory. [10]

stochastic. Pertaining to a process, model, or variable whose outcome, result, or value depends on chance. See deterministic. [1, 2]

structural model. A representation of the physical or logical structure of a system (e.g., a representation of a computer network as a set of boxes connected by communication lines). See process model. [1, 2]

structural validation. The process of determining that the modeling and simulation assumptions, algorithms, and architecture provide an accurate representation of the composition of the real world as relevant to the intended use of the models and simulations. [29]

subject area. 1. A major, high-level classification of data. 2. A group of entity types that pertain directly to a function or major topic of interest to the enterprise. [6]

symbolic model. A model whose properties are expressed in symbols (e.g., graphical models, mathematical models, narrative models, software models, and tabular models). See physical model. [1, 2]

symbolology. A graphic representation of concepts or physical objects. [22]

synthetic battlefield. One type of synthetic environment. [8]

synthetic environment database. An integrated set of data elements, each describing some aspect of the same geographical region and the elements or events expected there. [9]

synthetic environments (SE). Internetted simulations that represent activities at a high level of realism from simulations of theaters of war to factories and manufacturing processes. These environments may be created within a single computer or a vast distributed network connected by local and wide area networks and augmented by super-realistic special effects and accurate behavioral models. They allow visualization of and immersion into the environment being simulated. [8, 23]

system. A collection of components organized to accomplish a specific function or set of functions. [2]

T

T-1. Data communications service that supports 1.544 megabits per second bandwidth. [28]

T-2. Data communications service that supports 45 megabits per second bandwidth. [28]

tabular model. A symbolic model whose properties are expressed in tabular form (e.g., a truth table that represents a Boolean logic "OR" function). See graphical model, mathematical model, narrative model, software model. [1, 2]

taxonomy. A classification system that provides the basis for classifying objects for identification, retrieval and research purposes. [18]

technical data. Scientific or technical information recorded in any form or medium (e.g., manuals and drawings). Computer programs, related software, financial data and other information related to contract administration are not technical data where documentation of computer programs and related software are. [14]

terrain representation. The configuration, composition, and representation of the surface of the earth, including its relief, natural features, permanent or semi-permanent man-made features, related processes, terrain coverage including seasonal and diurnal variation such as grasses and snow, foliage coverage, tree type, and shadow. The terrain surface includes inland waters, and the sea floor bottom to the 20 meter depth curve. [9]

terrain skin. The physical confirmation of the Earth's surface. [9]

textures. Application of surface detail to a polygon by mapping an image to the polygon (i.e., to show foliage on a polygon to represent a tree) [9]

three-dimensional (3-D). A visual display that exhibits breadth, height and thickness or depth. [31]

tightly coupled. A condition that exists when simulation entities are involved in very close interaction such that every action of an entity must be immediately accounted for by the other entities (e.g., several tanks in close formation involved rapid, complicated maneuvers over the terrain). [13]

tile. A spatial partition of a coverage that shares the same set of feature classes with the same definitions as the coverage. [9]

time flow mechanism. The approach used locally by an individual federate to perform time advancement (e.g., event driven (or event stepped), time driven, and independent time advance (real-time synchronization mechanisms). [7]

time management. A collection of mechanisms and services to control the advancement of time within each federate during an execution in a way that is consistent with federation requirements for message ordering and delivery. [7]

time stamp (of an event). A value representing a point on the federation time axis that is assigned to an event to indicate when that event is said to occur. Certain message ordering services are based on this time stamp value. In constrained simulations, the time stamp may be viewed as a deadline indicating the latest time at which the message notifying the federate of the event may be processed. [7]

time stamp order (TSO). A total ordering of messages based on the "temporally happens before" (\rightarrow_t) relationship. A message delivery service is said to be time stamp ordered if for any two messages M_1 and M_2 (containing notifications of events E_1 and E_2 , respectively) that are delivered to a single federate where $E_1 \rightarrow_t E_2$, then M_1 is delivered to the federate before M_2 . The Runtime Infrastructure ensures that any two time stamp ordered messages will be delivered to all federates receiving both messages in the same relative order. To ensure this, the Runtime Infrastructure uses a consistent tie-breaking mechanism to ensure that all federates perceive the same ordering of events containing the same time stamp. Further, the tie-breaking mechanism is deterministic, meaning repeated executions of the federation will yield the same relative ordering of these events if the same initial conditions and inputs are used, and all messages are transmitted using time stamp ordering. [7]

time step models. Dynamic models in which time is advanced by a fixed or independently determined amount to a new point in time, and the states or status of some or all resources are updated as of that new point in time. Typically these time steps are of constant size, but they need not be. [18]

time variable. A variable whose value represents simulated time or the state of the simulation clock. [1, 2]

time. The measurable aspect of duration. Time makes use of scales based upon the occurrence of periodic events. These are: the day, depending on the rotation of the Earth; the month, depending on the revolution of the

Moon around the Earth; and the year, depending upon the revolution of the Earth around the Sun. Time is expressed as a length on a duration scale measured from an index on that scale (e.g., 4 p.m.). Local mean solar time means that 4 mean solar hours have elapsed since the mean Sun was on the meridian of the observer. [7]

time-dependent event. An event that occurs at a predetermined point in time or after a predetermined period of time has elapsed. See conditional event. [1, 2]

time-slice simulation. 1. A discrete simulation that is terminated after a specific amount of time has elapsed (e.g., a model depicting the year-by-year forces affecting a volcanic eruption over a period of 100,000 years); time-interval simulation. See critical event simulation. 2. A discrete simulation of continuous events in which time advances by intervals chosen independent of the simulated events (e.g., a model of a time multiplexed communication system with multiple channels transmitting signals over a single transmission line in very rapid succession). [1]

tolerance. 1. The maximum permissible error or the difference between the maximum and minimum allowable values in the properties of any component, device, model, simulation or system relative to a standard or referent. Tolerance may be expressed as a percent of nominal value, plus and minus so many units of a measurement, or parts per million. [45, 46, 47] 2. The character, state or quality of not interfering with some thing or action. [44, 45].

topology. 1. time-tested technique for storing a variety of relationship information amongst features that allows you to quickly answer certain types of questions. 2. Any relationship between connected geometric primitives that is invariant under transformation by continuous mappings. [9]

transmit management. The control of the transmission rate to match the transmission media. The transmission rate is selected to reduce total network traffic. [1]

transportation service. A Runtime Infrastructure provided service for transmitting messages between federates. Different categories of service are defined with different characteristics regarding reliability of delivery and message ordering. [7]

true global time. A federation-standard representation of time synchronized to Greenwich Mean Time or Universal Time [Coordinated] with or without some offset (positive or negative) applied. [7]

truth. 1. Conformity to fact or actuality. 2. Faithful to an original or standard. 3. Reality; actuality. 4. A

statement proven to be or accepted as true. [4] 5. A property implicitly ascribed to a proposition by belief in or assertion of it; the denial is "falsity". 6. In the verification theory of truth, a correspondence between the proposition and the events, properties or objects to which it refers linguistically or operationally. 7. In the logical theory of truth, the coherence between that proposition and other propositions. 8. In the constructivist theory of truth, constructability implying the absence of paradox and contradiction. [36]

two-dimensional (2-D). A visual display that exhibits only height and breadth (e.g., computer images and television). [31]

typing. The enforcement of the class of an object, such that objects of different types may not be interchanged, or may be interchanged only in restricted ways. [25]

U

unbundling. The process of unpacking a bundled Protocol Data Unit into multiple separate Protocol Data Units. See bundling. [1]

unconstrained simulation. A simulation where there is no explicit relationship between wall clock time and the rate of time advancements, sometimes called "as-fast-as-possible" simulations. Analytic simulation models and many constructive "war game" simulations are often unconstrained simulations. [7]

unicast. A transmission mode in which a single message is sent to a single network destination (i.e., one-to-one). [1, 13]

unit. 1. An aggregation of entities. 2. A basis of measurement. [1, 2]

unit conversion. A system of converting measurement from one basis to another (e.g., English/metric, knots/feet per second). [1]

Universal Transverse Mercator projection. An ellipsoidal Transverse Mercator Projection to which specific parameters, such as central meridians, have been applied. The Earth, between latitudes 84.0 degrees North and 80.0 degrees South, is divided into 60 zones each generally 6 degrees wide in longitude. [9]

Universal Time [Coordinated] (UTC). Greenwich Mean Time. A nonuniform time based on the rotation of the Earth, which is not constant; Coordinated Universal Time. [7]

Universal Space Rectangular (USR) Coordinate System. A right-handed orthogonal coordinate system

with its origin at the center of the Earth, positive x-axis in the equatorial plane and passing through the zero degree meridian, positive y-axis in the equatorial plane and passing through the ninety degree east meridian, and positive z-axis passing through the North Pole. [33]

user. Persons or organizations that are or will be the recipients of simulation products or services, and who, as a result of this position, may be involved in the evolution of such products or services. [41]

V

validation. The process of determining the degree to which a model or simulation is an accurate representation of the real-world, or some other meaningful referent, from the perspective of the intended uses of the model or simulation. [27, 40]

validity. 1. The quality of being inferred, deduced or calculated correctly enough to suit a specific application. 2. The quality of maintained data that is found on an adequate system of classification (e.g., data model) and is rigorous enough to compel acceptance for a specific use. [41] 3. The logical truth of a derivation or statement, based on a given set of propositions. [47]

variable. A quantity or data item whose value can change. See dependent variable, independent variable, state variable, constant. [1, 2]

verification. The process of determining that a model or simulation implementation accurately represents the developer's conceptual description and specification. Verification also evaluates the extent to which the model or simulation has been developed using sound and established software engineering techniques. [8]

vignette. A self-contained portion of a scenario. [1]

virtual. 1. The essence or effect of something, not the fact. [31] 2. Existing or resulting in effect or essence though not in actual fact, form, or name. [4]

virtual battlespace. The illusion resulting from simulating the actual battlespace. [1]

virtual prototype. A model or simulation of a system placed in a synthetic environment, and used to investigate and evaluate requirements, concepts, system design, testing, production, and sustainment of the system throughout its life cycle. [8]

virtual simulation. A simulation involving real people operating simulated systems. Virtual simulations inject human-in-the-loop in a central role by exercising motor control skills (e.g., flying an airplane), decision skills

(e.g., committing fire control resources to action), or communication skills (e.g., as members of a C4I team). See live simulation, virtual simulation, constructive simulation. [8]

virtual time. See simulated time. [1]

virtual world. See synthetic environment. [1]

visualization. The formation of an artificial image that cannot be seen otherwise, typically, abstract data that would normally appear as text and numbers is graphically displayed as an image. The image can be animated to display time varying data. [31]

W

wall clock time. A federate's measurement of true global time, where the measurement is typically output from a hardware clock. The error in this measurement can be expressed as an algebraic residual between wall clock time and true global time or as an amount of estimation uncertainty associated with the wall clock time measurement software and the hardware clock errors. [7]

warfare simulation. A model of warfare or any part of warfare for any purpose (such as analysis or training). [1, 18]

war game. A simulation game in which participants seek to achieve a specified military objective given pre-established resources and constraints (e.g., a simulation in which participants make battlefield decisions and a computer determines the results of those decisions); constructive simulation; higher order model. See management game. [1, 2]

well formed formula. In logic, a sequence of symbols from a formal language constructed according to the formation rules of the language. [35]

white box model. See glass box model. [1]

wide area network (WAN). A communications network designed for large geographic areas. [1, 2]

World Coordinate System. The right-handed geocentric Cartesian system. The shape of the world is described by the World Geodetic System 1984 standard. The origin of the world coordinate system is the centroid of the Earth. The axes of this system are labeled X, Y, and Z, with: the positive X-axis passing through the Prime Meridian at the Equator; the positive Y-axis passing through 90 degrees East longitude at the Equator; and the positive Z-axis passing through the North Pole. [1, 2]

World Geodetic System 1984 (WGS 84). A geocentric coordinate system which describes a basic frame of reference and geometric figure for the Earth, and which models the Earth from a geometric, geodetic, and gravitational standpoint. The WGS 84 coordinate system origin and axes also serve as the x, y, and z axes of the WGS 84 ellipsoid, the z axis being the rotational axis. [34]

world view. The view each simulation entity maintains of the simulated world from its own vantage point, based on the results of its own simulation and its processing of event messages received from all external entities. For Computer Generated Forces and for manned simulators or real vehicles, the world view is the perceptions of the participating humans. [1, 13]

X, Y & Z

yoked variable. One of two or more variables that are dependent on each other in such a manner that a change in one automatically causes a change in the others. [1, 2]

REFERENCES

These references relate only to the glossary, i.e., Appendix A.

- [1] "A Glossary of Modeling and Simulation Terms for Distributed Interactive Simulation (DIS)," August 1995.
- [2] Institute of Electrical and Electronics Engineers (IEEE), "IEEE Standard Glossary of Modeling and Simulation Terminology," IEEE Std 610.3-1989, nd.
- [3] Simulation Interoperability Standards Organization, RDE Forum, "Glossary of Terms Applied to Fidelity," nd.
- [4] Houghton Mifflin Co., Webster's II, New College Dictionary, 1995.
- [5] Department of Defense, "Data Quality Assurance Procedures," DoD 8320.1-M-3, February 1994
- [6] Department of Defense, "Data Administration Procedures," DoD 8320.1-M, 29 March 1994, authorized by DoD Directive 8320.1, 26 September 1991.
- [7] Defense Modeling and Simulation Office, High Level Architecture Glossary, <http://hla.dmsomil>, nd.
- [8] Department of Defense, "Modeling and Simulation Master Plan," DoD 5000.59-P, October 1995.
- [9] SEDRIS Glossary, 29 June 1998.
- [10] Defense Systems Management College (DSMC), "Systems Acquisition Manager's Guide for the Use of Models and Simulation," September 1994.
- [11] Federal Information Processing Standard (FIPS) Publication (PUB) 184 "Integration Language for Information Modeling (IDEFIX), nd.
- [12] Department of Defense, "Military Handbook for Joint Data Base Elements for Modeling and Simulation (M&S)," 5 August 1993.
- [13] Navy Air Weapons Center, "M&S Educational Training Tool (MSETT), Navy Air Weapons Center Training Systems Division Glossary," 28 April 1994.
- [14] Defense Modeling and Simulation Office, "Modeling and Simulation Glossary," January 1998.
- [15] Department of Defense, "Data Element Standardization Procedures," DoD 8320.1-M-1, 15 January 1993.
- [16] Federal Information Processing Standard (FIPS) Publication (PUB) 11-3 "American National Dictionary for Information Systems," (adopted in entirety from American National Standards Institute (ANSI) X3.172-1990), February 1991.
- [17] Department of Defense, "Mandatory Procedures for Major Defense Acquisition Programs (MDAPs) and Major Automated Information System (MAIS) Acquisition Programs," DoD Directive 5000.2-R, 15 March 1996.
- [18] Military Operations Research Society (MORS), "A Taxonomy for Warfare Simulation (SIMTAX)," 27 October 1989.
- [19] D. B. Fink & D. Christiansen, eds. Electronics Engineers' Handbook, 3rd Edition, McGraw-Hill Book Co., New York, NY, 1989.
- [20] Department of Defense, "Software Development and Documentation," DoD Std 498, 5 December 1994.
- [21] National Bureau of Standards (NBS), "Guide to Information Resource Dictionary System Applications: General Concepts and Strategic Systems Planning," NBS Special Pub 500-152, April 1988.
- [22] Department of Defense, "DoD Data Administration," DoD Directive 8320.1, 26 September 1991.

- [23] Department of the Army, "Army Model and Simulation Master Plan," 18 May 1995.
- [24] Department of Defense, "Department of Defense Dictionary of Military and Associated Terms," Joint Pub 1-02, 23 March 1994.
- [25] Defense Modeling and Simulation Office (DMSO), "Survey of Semi-Automated Forces," 30 July 1993.
- [26] Department of Defense, "Defense Information Management Program," DoD Directive 8000.1, 27 October 1992.
- [27] Department of Defense, "DoD Modeling and Simulation (M&S) Verification, Validation, Accreditation (VVA)," DoD Instruction 5000.61, 29 April 1996.
- [28] Department of the Navy, "Marine Corps Modeling and Simulation Master Plan," 29 July 1994.
- [29] Department of the Army, "Verification, Validation, and Accreditation of Army Models and Simulations," Pamphlet (DA PAM) 5-11, 15 October 1993.
- [30] Naval Sea Systems Command, SC-21 Program Office, "21st Century Surface Combatant Modeling and Simulation Master Plan," nd.
- [31] Defense Systems Management College (DSMC), "Virtual Prototyping: Concept to Production," March 1994.
- [32] U.S. Army Training and Doctrine Command, "TRADOC Modeling and Simulation Management Plan," 31 July 1991.
- [33] Department of Defense, "Glossary of Mapping, Charting, and Geodetic Terms," MIL-HDBK-850, 21 January 1994.
- [34] Defense Mapping Agency (DMA), "Department of Defense World Geodetic System 1984," DMA Technical Report 8350.2, 2nd Edition, 1 September 1991.
- [35] J. Daintith & R. D. Nelson, "Dictionary of Mathematics," Penguin Books USA, Inc., New York, NY, 1989.
- [36] F. Heylighen, Web Dictionary of Cybernetics and Systems, <http://pespmcl.vub.ac.be/ASC/indexASC.html>, nd.
- [37] H. J. Greenberg, Mathematical Programming Glossary, <http://www-math.cudenver.edu/~hgreenbe/glossary/glossary.html>, 11 October 1998.
- [38] General Chemistry Glossary, <http://genchem.chem.wisc.edu/labdocs/glossary/glossary.htm>, nd.
- [39] E. W. Weisstein, CRC Concise Encyclopedia of Mathematics, CRC Press, LLC, Boca Raton, FL, 1998.
- [40] Defense Modeling and Simulation Office VV&A Technical Support Team, Verification, Validation, and Accreditation (VV&A) Recommended Practices Guide, Defense Modeling and Simulation Office, Alexandria, VA, 1997.
- [41] The result of Fidelity ISG discussions and comments from March 1998 to December 1998.
- [42] R. Prieto-Diaz & G. Arango, Domain Analysis and Software Systems Modelling, IEEE Computer Society Press, Los Alamitos CA, 1991.
- [43] Encyclopedia of Operations Research and Management Science," eds Saul I. Gass & Carl M. Harris, Kluwer Academic Publishers, 1996. [DH 11/10]
- [44] Funk & Wagnalls Standard Desk Dictionary, Volumes 1 & 2, Harper & Row Publishers, Inc., 1984.
- [45] Webster's New World Dictionary of American English, 3rd College Edition, V. Neufeld & D.B. Gurlink, eds., Simon & Schuster, Inc., Cleveland, OH, 1994.
- [46] The Penguin Dictionary of Electronics, 2nd Edition, V. Illingworth, Penguin Books, New York, NY, 1988.
- [47] The Illustrated Dictionary of Electronics, 5th Edition, R. P. Turner & S. Gibilisco, eds., TAB Professional & Reference Books, Blue Summit, PA., 1991.
- [48] The International Dictionary of Applied Mathematics, D. Van Nostrand Co., Inc., Princeton, NJ, 1960

Appendix B. A Mathematical Foundation for Fidelity Standards

Many concepts related to modeling and simulation (M&S) fidelity exist, e.g., reality, resolution, accuracy and validity. However, as with fidelity, the specific definitions of these related concepts have been fuzzy so these terms have been used quite broadly. The poorly defined nature of these related concepts complicates the appearance of their relationships to fidelity and to each other. This article attempts to demystify these relationships by proposing a formal structure within which the fidelity-related concepts can be defined accurately. This formal structure itself contains nothing unique, beyond a particular notation. Many authors have proposed similar structures for various purposes. This discussion takes advantage of these past forages into fidelity to explain the related terms. This discussion also contrasts the formal definitions of these related concepts to their definitions in the M&S Fidelity Glossary, provided in the Appendix.

B.1 Simulands, Referents, and Reality

The world to be modeled or simulated can be defined by the set S such that

$$S = \{S_1, S_2, \dots, S_n\} \quad (1)$$

where

$$S_1 = \text{the } i\text{th object in the actual world to be modeled or simulated.}$$

The state and behavior manifested by S defines the simuland. The body of knowledge about the world S is a subset of that world. This codified body of knowledge can also be organized as a set of objects, R , such that

$$R = \{R_1, R_2, \dots, R_n\} \quad (2)$$

where

$$R_1 = \text{the knowledge about the } i\text{th object in the world to be modeled or simulated.}$$

The knowledge about the state and behavior of the objects in S created by R defines the referent. Models and simulations of S can be abstracted from this knowledge. Further, R also provides the knowledge against which to compare the results produced by models and simulations of S .

A set, R_i , describes the nature of each object in R where

$$R_i = \{P_i, G_i\} \quad (3)$$

and

$P_i =$ the set of properties whose values describe the state of R_i and

$G_i =$ the set of dependencies that couple the object states in R .

A vector \underline{P}_i represents the state of any object R_i in R where

$$\underline{P}_i = \sum p_{ij} p_{ij} \quad (4)$$

and

$p_{ij} =$ the value of j th property of object O_i

$p_{ij} =$ the unit vector representing the j th property of object O_i .

The set P , where

$$P = \{P_1, P_2, \dots, P_n\}, \quad (5)$$

represents all of the properties of all of the objects in R . The number of members in P defines the dimensionality of vector space \underline{P} , the property space.

The tensor \underline{P} , where

$$\underline{P} = \sum \underline{P}_i r_i \quad (6)$$

and

$r_i =$ the unit vector representing the i th object in R .

Each property P_{ij} has associated with it a range of possible values given by the set W_{ij} where

$$p_{ij} \in W_{ij}. \quad (7)$$

When all properties have real numbered values, there exist two members, w_{lij} and w_{uij} , of W_{ij} where

$$w_{lij} \quad p_{ij} \quad w_{uij} \quad (8)$$

and

$w_{lij} =$ the lower limit value of the acceptable range of property P_{ij} and

$w_{uij} =$ the upper limit value of the acceptable range of property P_{ij}

and where

$$w_{lij} < w_{uij} \quad (9)$$

for the value, p_{ij} , of any property, P_{ij} , of any object R_i in R .

This restriction to real numbered values may seem a significant limitation but the extension of these arguments to other classes of values is straightforward. By extending this assumption, the number of members in each set W_{ij} can be reduced to just the lower and upper limits of the acceptable range, as defined above, or

$$W_{ij} = \{w_{lij}, w_{uij}\}. \quad (10)$$

The set G_{ij} describes a dependency of object R_i in R where

$$G_{ij} = \{U_{ij}, V_{ij}, g_{ij}\} \quad (11)$$

and

U_{ij} = the set of object properties representing the independent variables of the dependency G_{ij} ,

V_{ij} = the set of object properties representing the dependent variables of the dependency G_{ij} , and

g_{ij} = the dependency function that determines the values of the dependent variables from the values of the independent variables;

and where

$$U_{ij} \subseteq P \text{ and } V_{ij} \subseteq P. \quad (12)$$

The dependency function, g_{ij} , relates the values of the properties in V_{ij} to the values of the properties in U_{ij} such that

$$\underline{V}_{ij} = g_{ij}(\underline{U}_{ij}) \quad (13)$$

where

\underline{V}_{ij} = the vector in property space representing the values of the properties in the set V_{ij} and

\underline{U}_{ij} = the vector in property space representing the values of the properties in the set U_{ij} .

When the set R' represents the entire body of knowledge about all of the objects in the entire known universe; the set P' represents all of the properties describing the state of the entire known universe, the set G' represents all of the dependencies between all of the object properties in the entire known universe, and R , P and G represent the sets of objects, properties and dependencies,

respectively, sufficient to describe the world for some application, model or simulation then

$$R \subset R', P \subset P' \text{ and } G \subset G'. \quad (14)$$

Definition 1: The sets R' , P' and G' define the known reality of the entire world.

The sets R , P and G abstract those aspects of reality that are necessary and sufficient for a particular application. Reality may be a world that has been manifested or could be manifested. However, reality generally has the quality of being actual or true, as given in the M&S Fidelity Glossary. Definition 1 further specifies the nature of actual or true.

B.2 Models and Simulations as Approximations

The set O defines that part of the world R that a modeled or simulation represents, where

$$O = \{O_1, O_2, \dots, O_m\} \quad (15)$$

and

O_i = the object O_i in O that models or simulates one or more objects in R .

As in R , the set O_i describes each object O_i in the modeled or simulated world, O , where

$$O_i = \{Q_i, F_i\} \quad (16)$$

and

Q_i = the set of properties whose values describe the state of O_i and

F_i = the set of dependencies that couple the object states in O .

Also similarly, the vector \underline{Q}_i describes the state of any object O_i in O where

$$\underline{Q}_i = \sum q_{ij} q_{ij} \quad (17)$$

and

q_{ij} = the value of the j th property of the object O_i represented by the model or simulation

q_{ij} = the unit vector representing the j th property of the object O_i represented by the model or simulation.

The set Q , where

$$Q = \{Q_1, Q_2, \dots, Q_n\}, \quad (18)$$

represents all of the modeled or simulated properties of all of the objects in O.

The tensor, $\underline{\mathbf{Q}}$, where

$$\underline{\mathbf{Q}} = \Sigma \underline{\mathbf{Q}}_i o_i \quad (19)$$

and

$$o_i = \text{the unit vector representing the object } O_i \text{ in } O.$$

$\underline{\mathbf{Q}}$ represents the state of O at any instant in time. Each property Q_{ij} has associated with it a range of possible values contained by the set Z_{ij} where

$$q_{ij} \in Z_{ij}. \quad (20)$$

As in R, all of the values of all of the properties in O are assumed real numbers. Thus for any property Q_{ij} of any object O_i in O there exists two members of Z_{ij} , z_{lij} and z_{uij} , where

$$z_{lij} \leq q_{ij} \leq z_{uij} \quad (21)$$

and

$$z_{lij} = \text{the lower limit of the acceptable range of values of the property } Q_{ij} \text{ and}$$

$$z_{uij} = \text{the upper limit of the acceptable range of values for the property } Q_{ij}$$

and where

$$z_{lij} < z_{uij}. \quad (22)$$

As mentioned earlier, the restriction to real numbered values may seem a significant limitation, however, the extension of these arguments to other classes of values is straightforward. If the set Z_{ij} contains a continuous range of values then the number of members in each set Z_{ij} can be reduced so that

$$Z_{ij} = \{z_{lij}, z_{uij}\}. \quad (23)$$

The vectors, $\underline{\mathbf{Z}}_{li}$ and $\underline{\mathbf{Z}}_{ui}$, defined by

$$\underline{\mathbf{Z}}_{li} = \Sigma z_{lij} q_{ij} \text{ and } \underline{\mathbf{Z}}_{ui} = \Sigma z_{uij} q_{ij}, \quad (24)$$

represent the values of the lower and upper limits describing the valid ranges of all of the properties depicting the state of the object O_i .

These vectors can be combined into the tensors in object \times property space, defined by

$$\underline{\mathbf{Z}}_l = \Sigma \underline{\mathbf{Z}}_{li} o_i \text{ and } \underline{\mathbf{Z}}_u = \Sigma \underline{\mathbf{Z}}_{ui} o_i, \quad (25)$$

that represent all of the lower and upper limits of the all of the properties in Q. For every property P_{ij} in P,

$$w_{lij} \leq z_{lij} < z_{uij} \leq w_{uij} \quad (26)$$

If a finite set of values, W_{ij} , sufficiently describes the range of values that the actual property P_{ij} can assume and a finite set of values, Z_{ij} , sufficiently describes the range of values that the modeled or simulated property Q_{ij} can assume then

$$Z_{ij} \subseteq W_{ij} \quad (27)$$

for every property Q_{ij} for every object $O_i \in O$.

The set F_{ij} describes each dependency in F where

$$F_{ij} = \{X_{ij}, Y_{ij}, f_{ij}\} \quad (28)$$

and

$$X_{ij} = \text{the set of modeled or simulated object properties representing the independent variables of the dependency } F_{ij},$$

$$Y_{ij} = \text{the set of modeled or simulated object properties representing the dependent variables of the dependency } F_{ij}, \text{ and}$$

$$f_{ij} = \text{the modeled or simulated dependency function that determines the values of the dependent variables from the values of the independent variables;}$$

and where:

$$X_{ij} \subseteq Q \text{ and } Y_{ij} \subseteq Q. \quad (29)$$

The function f_{ij} relates the values of the properties in the set Y_{ij} to the values of the properties in the set X_{ij} such that

$$\underline{\mathbf{Y}}_{ij} = f_{ij}(\underline{\mathbf{X}}_{ij}) \quad (30)$$

Where:

$$\underline{\mathbf{Y}}_{ij} = \text{the vector in property space representing the values of the properties in the set } Y_{ij} \text{ and}$$

$$\underline{\mathbf{X}}_{ij} = \text{the vector in property space representing the values of the properties in the set } X_{ij}.$$

B.3 Abstraction, Resolution, and Detail

Definition 2: The process of choosing the sets for a model or simulation so that

$$O \subset R, Q \subset P, \text{ and } F \subset G; \quad (31)$$

and so that

$$w_{lij} \leq z_{lij} < z_{uij} \leq w_{uij} \quad (32)$$

for all of the properties associated with all of the objects in O represents the process of **abstraction**.

The choices of the sets O, Q and F identify the essential aspects of R that the model or simulation must represent. The sets R – O, P – Q, and G – F and the tensors \underline{Z} – \underline{W} and \underline{W}_u – \underline{Z}_u identify what was ignored because their members do not contribute to the purpose. Thus, this formal definition corresponds with the first definition proposed in the M&S Fidelity Glossary.

The sets O, Q, F and Z lead to the following definition of the resolution of a model or simulation.

Definition 3: The sets O, Q, F and Z define the **resolution** or level of **detail** of a model or simulation.

Definition 3 equates the definitions of resolution and level of detail. Each object, property or dependency represents a separate detail of the modeled or simulated world. The collections of these objects, properties and dependencies define the degree of detail of the model or simulation. The definitions of the modeled or simulated elements as subsets of the referent world relate resolution to the level of abstraction of the model or simulation. Consequently, Definition 3 is consistent with the first definitions of both resolution and detail in the M&S Fidelity Glossary.

B.4 Error and Accuracy

Only the state of modeled, simulated or real worlds can be observed. This means that the behaviors manifested by the dependencies from the sets G and F can only be observed through the changes of state in the properties contained by the sets P and Q. This suggests that the characteristics of the dependencies in G and F can be completely described by the characteristics of their independent and dependent variables.

In general, any modeled or simulated dependency function, f_{ij} in F, will only approximate a corresponding dependency function, g_{ij} in G, where

$$\underline{Y}_{ij} = f_{ij}(\underline{X}_{ij}) \approx \underline{V}_{ij} = g_{ij}(\underline{U}_{ij}). \quad (33)$$

Sometimes, a single modeled or simulated dependency function will approximate the combined behavior of

several interacting dependencies in G. This aggregation of dependencies occurs during the abstraction process. In this case, the dependent variable vector of the referent represents all of those properties whose state depends upon the group of functions being modeled or simulated. The interacting group of dependencies is treated as a single function g_{ij} . The difference between the results from a modeled or simulated approximation and the one or more referent functions being approximated is the error of the approximation. This is captured below in Definition 4.

Definition 4: The vector in property space $\underline{\epsilon}_{ij}$, where

$$\underline{\epsilon}_{ij} = \sum |y_{ijk} - v_{ijk}|q_{ijk} \quad (34)$$

and

y_{ijk} = the value of the kth dependent variable representing a property in Y_{ij} , and

v_{ijk} = the value of the kth dependent variable representing a property in V_{ij} .

defines the **error** of the approximate dependency function f_{ij} .

The vector defined in Definition 4 characterizes the errors associated with each of the properties representing the dependent variables of a dependency function f_{ij} . All of the components of $\underline{\epsilon}_{ij}$ are positive values so

$$\underline{e}_i \leq 0 \quad (35)$$

In effect, f_{ij} computes the values that are compared against the values produced by the corresponding dependency function g_{ij} in R. The components of $\underline{\epsilon}_{ij}$ are the differences between what is computed by f_{ij} , the model or simulation, and the correct values or what is computed by g_{ij} , the referent. This formal definition of error is therefore consistent with that presented in the M&S Fidelity Glossary.

In general, for any approximation of error

$$\underline{s}_{ij} = \underline{\epsilon}_{ij}(\underline{X}_{ij}). \quad (36)$$

The behavior of the error function, $\underline{\epsilon}_{ij}(\underline{X}_{ij})$, depends upon the characteristics of the approximation f_{ij} . In some cases, the values of $\underline{\epsilon}_{ij}$ may not be well characterized over the entire range of possible values in the set W_{ij} for the referent dependency g_{ij} . However, the process of abstraction is assumed to choose a subset, Z_{ij} , of the range W_{ij} over which the error is well known. It is further assumed that there exists a constant vector, $\underline{\epsilon}_{mij}$ over Z_{ij} , where

$$\mathbf{e}_{mij} \geq \mathbf{e}_{ij}(\underline{X}_{ij}) \quad (37)$$

for all components x_{ijk} such that

$$z_{lij} \leq x_{ijk} \leq z_{uij} \quad (38)$$

The constant vector $\underline{\mathbf{e}}_{mij}$ represents the maximum possible error between the dependent variable values produced by a model or simulation and those values produced by its referent over the approximate dependency function's acceptable range. This constant vector together with the acceptable range capture two of the characteristics of the dependency, \underline{f}_{ij} .

As the maximum error of a dependency, $\underline{\mathbf{e}}_{mij}$, decreases then the accuracy of that modeled or simulated approximation increases. This correlation suggests the following related definition for accuracy.

Definition 5: The vector in property space, $\underline{\alpha}_{ij}$, where

$$\underline{\alpha}_{ij} = \mathbf{k}_{ij}(\underline{\mathbf{e}}_{mij})^{-1} \quad (39)$$

and

$\underline{\mathbf{e}}_{mij}$ = the vector representing the maximum error associated with each component of the vector \underline{Y}_{ij} and

\mathbf{k}_{ij} = a constant value that scales a dependency function's error to its accuracy,

defines the **accuracy** of the dependency function \underline{f}_{ij} .

From Definition 5, when the error of a model or simulation is zero then its accuracy is maximized. The accuracy, as defined above, represents the degree to which a dependent variable corresponds with the reality defined by R. This notion is consistent with the informal definition of accuracy given in the M&S Fidelity Glossary. This definition also clearly distinguishes between the terms error and accuracy. They are not synonyms and probably should not be used interchangeably, although they often are.

B.5 Sensitivity and Precision

Every approximation of some dependency in reality has, at least, two other characteristics. Definition 6, below, defines first of these characteristics.

Definition 6: The vector in property space $\underline{\Delta x}_{ij}$, associated with the dependency function \underline{f}_{ij} and defined by the relationships

when $\underline{X}_{ij} < \underline{\Delta x}_{ij}$ for every component of \underline{X}_{ij}

$$\text{then } \underline{Y}_{ij} = \underline{f}_{ij}(\underline{X}_{ij}) = 0 \quad (40)$$

and

when $\underline{X}_{ij} \geq \underline{\Delta x}_{ij}$ for every component of \underline{X}_{ij}

$$\text{then } \underline{Y}_{ij} = f(\underline{X}_{ij}) \neq 0, \quad (41)$$

represents the **sensitivity** of \underline{f}_{ij} .

Not all dependency functions necessarily have nonzero sensitivities. The existence of a sensitivity vector, as defined above, depends entirely upon the nature of the approximation and how the computation of that approximation is implemented upon computational devices.

As with a dependency's error, for any dependency function \underline{f}_{ij}

$$\underline{\Delta x}_{ij} = \underline{\Delta x}_{ij}(\underline{X}_{ij}). \quad (42)$$

The behavior of $\underline{\Delta x}_{ij}(\underline{X}_{ij})$ depends upon the character of \underline{f}_{ij} . As with the error, it is assumed that there exists a constant vector, $\underline{\Delta x}_{mij}$, where

$$\underline{\Delta x}_{ij} \leq \underline{\Delta x}_{mij} \quad (43)$$

for all components of \underline{X}_{ij} and for all values of the property Q_{ik} such that $z_{lik} \leq x_{ijk} \leq z_{uik}$ for each component, x_{ijk} , of the vector \underline{X}_{ij} .

The constant vector $\underline{\Delta x}_{mij}$ defines the maximum possible sensitivity limitation over the acceptable range of \underline{f}_{ij} . The components of this vector define the lower limits of input changes to which \underline{f}_{ij} can respond. Therefore, Definition 6 is consistent with the information definition for sensitivity presented in the M&S Fidelity Glossary.

The second additional characteristic describes the limitations of a dependency function approximation in terms of its dependent variables.

Definition 7: The vector in property space $\underline{\Delta y}_{ij}$, associated with the dependency function \underline{f}_{ij} and defined by the relationship

$$\underline{Y}_{ij} = (\underline{N}_{ij}\underline{\Delta y}_{ij}) = \underline{f}_{ij}(\underline{X}_{ij}) \quad (44)$$

where

\underline{N}_{ij} = a vector in property space whose components are all integers

for all values of \underline{X}_{ij} and for all values of the property Q_{ik} such that

$$z_{lij} \leq x_{ijk} \leq z_{uij} \quad (45)$$

for each component, x_{ijk} , of the vector \underline{X}_{ij} , represents the *precision* of f_{ij} .

Again, not all dependency functions necessarily have nonzero precisions for all of their dependent properties. The existence of a precision vector, as defined above, depends upon the nature of the approximation and how the computation of that approximation is implemented.

As with a dependency's sensitivity, for any dependency function f_{ij}

$$\underline{\Delta y}_{ij} = \underline{\Delta y}_{ij}(\underline{Y}_{ij}). \quad (46)$$

The behavior of $\underline{\Delta y}_{ij}(\underline{Y}_{ij})$ depends upon the character of f_{ij} . As with the sensitivity, it is assumed that there exists a constant vector $\underline{\Delta y}_{mij}$ where

$$\underline{\Delta y}_{ij} \leq \underline{\Delta y}_{mij} \quad (47)$$

for all components of \underline{Y}_{ij} and for all values of the property Q_{ik} such that

$$z_{lik} \leq y_{ijk} \leq z_{uik} \quad (48)$$

for each component, y_{ijk} , of the vector \underline{Y}_{ij} .

The constant vector $\underline{\Delta y}_{mij}$ defines the maximum possible precision limitation over the acceptable range of f_{ij} . Definition 7 correlates with the first definition of precision given in the M&S Fidelity Glossary. An extension of this definition is required to accurately represent the second possible definition.

Definition 8: If the dependency f_{ij} behaves such that when the its values are computed two separate times with the independent variable vectors \underline{X}_{1ij} and \underline{X}_{2ij} where

$$\underline{Y}_{1ij} = f_{ij}(\underline{X}_{1ij}) \text{ and} \quad (49)$$

$$\underline{Y}_{2ij} = f_{ij}(\underline{X}_{2ij}) \text{ and} \quad (50)$$

and where

$$\underline{X}_{1ij} = \underline{X}_{2ij} \quad (51)$$

but

$$\underline{Y}_{1ij} \neq \underline{Y}_{2ij} \quad \underline{Y}_{1ij} \quad \underline{Y}_{2ij} \quad (52)$$

then the vector in property space, $\underline{\Delta y}_{ij}$, defined by

$$\underline{\Delta y}_{ij} = \Sigma |y_{2ijk} - y_{1ijk}| q_{ijk} \quad (53)$$

represents the precision of f_{ij} .

As with Definition 7, it is assumed that there exists a constant vector, $\underline{\Delta y}_{mij}$, where

$$\underline{\Delta y}_{ij} \leq \underline{\Delta y}_{mij} \quad (54)$$

for all components of \underline{Y}_{ij} and for all values of the property Q_{ik} such that

$$z_{lij} \leq y_{ijk} \leq z_{uij} \quad z_{lik} \quad y_{ijk} \quad z_{uik} \quad (55)$$

for each component, y_{ijk} , of the vector \underline{Y}_{ij} .

Definition 8 captures the notion of precision for stochastic functions whose computation produce dependent variable values with variance. In fact, variance, in this definition, is associated directly with precision. This direct association contradicts the inverse relationship suggested by the informal definition of precision presented in the M&S Fidelity Glossary. The same argument is also true for sensitivity, although in a lesser sense. While adjustments could be made to relieve these slight contradictions, as was done with error and accuracy, these changes would have only complicated the Definitions 7 and 8, perhaps unnecessarily, without adding information.

The precision and error of any dependency function are interrelated since limits in precision may contribute to the error of a model or simulation. Ideally,

$$\underline{\Delta y}_{mij} \ll \underline{\epsilon}_{mij}. \quad (56)$$

for all components of the vector \underline{Y}_{ij} .

This condition prevents limits in precision from influencing the limits in error Thus enabling these two characteristics to be treated as if they were independent. In addition, precision may be dependent upon sensitivity. In some cases, precision limitations may result completely from dependency sensitivity limitations. In other cases, precision limitations may result from other contributions due solely to the dependency function's executable representation and that representation's actual execution. If the condition

$$\underline{\Delta x}_{mij} \ll \underline{\Delta y}_{mij}, \quad (57)$$

for all components of \underline{X}_{ij} and \underline{Y}_{ij} , is true then the limits in sensitivity and precision may be treated as independent quantities.

B.6 Fidelity

The fidelity of a model or simulation describes how faithfully it represents the referent. The preceding discussion suggests the following formal definition of fidelity.

Definition 9: The *fidelity* of a model or simulation can be described by the set

$$E = \{O, Q, F, Z, \underline{\epsilon}_m, \underline{\Delta x}_m, \underline{\Delta y}_m\} \quad (58)$$

where

$O =$ the set of all objects represented by the model or simulation,

$Q =$ the set of all object properties represented by the model or simulation,

$F =$ the set of all object dependencies represented by the model or simulation,

$Z =$ the set of all object property ranges over which the model or simulation is well behaved,

$\underline{\epsilon}_m =$ the tensor in object \times property spaces representing the maximum possible values of error associated with all of the dependent variables over the acceptable ranges of all of the dependencies in F ,

$\underline{\Delta x}_m =$ the tensor in object \times property spaces representing the maximum possible values of sensitivity associated with all of the independent variables over the acceptable ranges of all of the dependencies in F , and

$\underline{\Delta y}_m =$ the tensor in object \times property spaces representing the maximum possible values of precision associated with all of the dependent variables over the acceptable ranges of all of the dependencies in F .

The set E very specifically describes the components of fidelity. However, this definition may not be complete in that additional components may be necessary to better characterize some modeled or simulated approximations of reality. This definition is presented as a starting point from which more specific and accurate definitions can be derived. Definition 9 suggests, unequivocally, that the components of fidelity for any approximation of reality can be specifically characterized and, where appropriate, actually measured. These measurements enable the capabilities of different approximations of the same phenomena to be compared. These measurements also

permit the capabilities of a model or simulation to be compared against the requirements of an application.

The informal definitions of fidelity presented in the M&S Fidelity Glossary essentially agree with that given in Definition 9. However, the M&S Fidelity Glossary definition goes, as one might reasonably expect, far beyond even the formal definition by identifying several constraints. Definition 9 fairly captures the notions of measuring fidelity against a referent and the need to carefully describe that referent in order for fidelity measurements to be meaningful. However, Definition 9 does not approach the issue of how the definitions of fidelity differ when applied to models, simulations, data and exercises. While this work is left to others, this discussion does provide a notational framework within which to differentiate these definitional differences.

B.7 Tolerance

For the sake of this discussion, it is assumed that the components of E represent a sufficient set of parameters to characterize the capabilities and limitations of any model or simulation over its acceptable range. These components contribute to the mechanism necessary to specify the validity of a model or simulation for a particular application. For any application A there exists a set of objects, O_A , properties, Q_A , and dependencies, F_A , such that

$$O_A \subset R \quad (59)$$

$$Q_A \subset P \quad (60)$$

$$F_A \subset G \quad (61)$$

that is necessary and sufficient to achieve that application.

In effect, the different sets between O_A and R , Q_A and P , and F_A and G define the range of sets that can provide the functionality necessary to achieve A . However, any functionality greater than that defined by O_A , Q_A and F_A may impact the application in other extra-functional ways (e.g., cost, maintenance, manning). As a result, a model or simulation that essentially duplicates reality may be unacceptable for an application although it provides more than enough functionality for that application. Thus, the required functionality for any application A can be described by the sets O_{AI} , O_{Au} , Q_{AI} , Q_{Au} , F_{AI} and F_{Au} such that

$$O_{AI} \subseteq O_{Au} \subset R, \quad (62)$$

$$Q_{AI} \subseteq Q_{Au} \subset P, \text{ and} \quad (63)$$

$$F_{AI} \subseteq O_{Au} \subset R. \quad (64)$$

An application may also impose restrictions upon the limits of the acceptable ranges of the properties it requires to be represented. Four tensors exist that describe these limit requirements, \underline{Z}_{lA} , \underline{Z}_{lAu} , \underline{Z}_{uA} and \underline{Z}_{uAu} , such that

$$\underline{W}_l \leq \underline{Z}_{lA} < \underline{Z}_{lAu} < \underline{W}_u, \quad (65)$$

$$\underline{W}_l < \underline{Z}_{uA} < \underline{Z}_{uAu} \leq \underline{W}_u, \text{ and} \quad (66)$$

$$\underline{Z}_{lA} < \underline{Z}_{lAu} < \underline{Z}_{uA} < \underline{Z}_{uAu}. \quad (67)$$

for all of the components of Z .

Applications may require the modeling or simulation of several instances of the objects in the set O_A . This capacity, required to represent a certain number of objects, can be described by the vector in object space \underline{C}_A such that

$$\underline{C}_A = \Sigma c_{Ai} o_{Ai} \quad (68)$$

where

c_{Ai} = the number of instances of the i th object required to be modeled or simulated to meet the requirements of application A.

Like the functionality, the capacity required for instances of object O_i for an application may need to be described as an acceptable range of values c_{Ali} and c_{Aui} such that

$$c_{Ali} \leq c_{Aui} \quad (69)$$

for all objects O_{Ai} in O_A .

The application A may also impose requirements upon the accuracy, sensitivity and precision. Each dependency, \underline{f}_{Aij} in F_A , is represented by the vectors $\underline{\epsilon}_{mAij}$, $\underline{\Delta x}_{mAij}$ and $\underline{\Delta y}_{mAij}$ in the property space Q_A where

$$\underline{\epsilon}_{mAij} = \Sigma \epsilon_{mAijk} q_{Aijk}, \quad (70)$$

$$\underline{\Delta x}_{mAij} = \Sigma \Delta x_{mAijk} q_{Aijk}, \text{ and} \quad (71)$$

$$\underline{\Delta y}_{mAij} = \Sigma \Delta y_{mAijk} q_{Aijk}. \quad (72)$$

Thus, for any application, three tensors in object \times property spaces, $\underline{\epsilon}_{m\Delta}$, $\underline{\Delta x}_{m\Delta}$, $\underline{\Delta y}_{m\Delta}$, can be constructed to represent the maximum error, sensitivity and precision limit requirements for the dependencies that must be represented to achieve the application's objectives. However, any application may have ranges of values of maximum error, sensitivity and precision that are

acceptable. In this case, these ranges are represented by the tensors $\underline{\epsilon}_{m\Delta}$, $\underline{\epsilon}_{mAu}$, $\underline{\Delta x}_{m\Delta}$, $\underline{\Delta x}_{mAu}$, $\underline{\Delta y}_{m\Delta}$, and $\underline{\Delta y}_{mAu}$ where

$$\underline{\epsilon}_{mA} \leq \underline{\epsilon}_{mAu}, \quad (73)$$

$$\underline{\Delta x}_{mA} \leq \underline{\Delta x}_{mAu}, \text{ and} \quad (74)$$

$$\underline{\Delta y}_{mA} \leq \underline{\Delta y}_{mAu}. \quad (75)$$

Definition 10: The set T_A where

$$T_A = \{O_{Al}, O_{Au}, Q_{Al}, Q_{Au}, F_{Al}, F_{Au}, \underline{Z}_{lA}, \underline{Z}_{lAu}, \underline{Z}_{uA}, \underline{Z}_{uAu}, \underline{C}_A, \underline{C}_{Au}, \underline{\epsilon}_{mA}, \underline{\epsilon}_{mAu}, \underline{\Delta x}_{mA}, \underline{\Delta x}_{mAu}, \underline{\Delta y}_{mA}, \underline{\Delta y}_{mAu}\} \quad (76)$$

defines the **tolerances** within which the functionality and performance requirements for the application A can be met by some model or simulation.

The subset

$$T_{Al} = \{O_{Al}, Q_{Al}, F_{Al}, \underline{Z}_{lA}, \underline{Z}_{uA}, \underline{C}_A, \underline{\epsilon}_{mA}, \underline{\Delta x}_{mA}, \underline{\Delta y}_{mA}\} \quad (77)$$

defines the lower limits of the tolerances for the application A and the subset

$$T_{Au} = \{O_{Au}, Q_{Au}, F_{Au}, \underline{Z}_{lAu}, \underline{Z}_{uAu}, \underline{C}_{Au}, \underline{\epsilon}_{mAu}, \underline{\Delta x}_{mAu}, \underline{\Delta y}_{mAu}\} \quad (78)$$

defines the upper limits of the tolerances for the application A.

These tolerances define the envelope of functionality and performance necessary to achieve an application's objectives. The maximum and minimum allowable values defined in T_A are all measured in terms of the referent that is appropriate for that application. These aspects of the formal definition make it completely consistent with the first definition for tolerance provided in the M&S Fidelity Glossary.

B.8 Fitness and Validity

Defining the capabilities of a model or simulation and the requirements for a modeling or simulation application in equivalent terms enables the definition of a model's or simulation's fitness for an application.

Definition 11: If a model or simulation, S, exists for which all of the following statements are true

$$O_{Al} \subseteq O_S \subseteq O_{Au}, \quad (79)$$

$$Q_{A1} \subseteq Q_S \subseteq Q_{Au}, \quad (80)$$

$$F_{A1} \subseteq F_S \subseteq F_{Au}, \quad (81)$$

$$\underline{Z}_{1A1} \leq \underline{Z}_{1S} \leq \underline{Z}_{1Au}, \quad (82)$$

$$\underline{Z}_{uA1} \leq \underline{Z}_{uS} \leq \underline{Z}_{uAu}, \quad (83)$$

$$\underline{C}_{A1} \leq \underline{C}_S \leq \underline{C}_{Au}, \quad (84)$$

$$\underline{e}_{mA1} \leq \underline{e}_{mS} \leq \underline{e}_{m1Au}, \quad (85)$$

$$\underline{\Delta x}_{mA1} \leq \underline{\Delta x}_{mS} \leq \underline{\Delta x}_{mAu}, \text{ and} \quad (86)$$

$$\underline{\Delta y}_{mA1} \leq \underline{\Delta y}_{mS} \leq \underline{\Delta y}_{mAu}. \quad (87)$$

specific property or object (depending upon the space). Subscripts have been used in a myriad of wonderful, and probably confusing, ways. The letters i, j and k in subscript always represent the indices into a set, a vector or a tensor. Further, these indices always trail other subscripts. The subscript letter m denotes a maximum value for which it is assumed that no values exceed over the acceptable range of possible values. Similarly, the subscript letters l and u refer to the lower and upper limits of a value. Finally, subscript A denotes some property that represents a requirement for an application and subscript S denotes some property characterizing a capability of a model or simulation.

then that model or simulation has the necessary and sufficient *fitness* for the application A.

This definition defines the capabilities needed for an application and then directly compares the capabilities provided by a model or simulation against those requirements. As a result, this definition of fitness agrees perfectly with the informal definition given in the M&S Fidelity Glossary. Further, these fitness conditions also establish the criteria to determine the validity of a model or simulation for an application. If a model or simulation meets the fitness conditions for an application then it is also valid for that application. As a result, the definition of validity can be constructed from the definition of fitness.

Definition 12: If a model or simulation has the necessary and sufficient fitness for a particular application then that model or simulation has sufficient *validity* for that application.

Definition 12 broadens the notion of validity considerably from any of those presented in the M&S Fidelity Glossary. However, this broadening does not come at the expense of a loss of precision. It just expands the definition of correctly enough.

B.9 Definitions of Symbols

This discussion uses many symbols. These have been arranged alphabetically in Table 1 with their definitions to aid the reader. Some conventions have been applied in the use of these symbols. A capital letter denotes a set of items. An underscore denotes a vector. Bolding denotes a tensor in multiple vector spaces. A lower case letter identifies the value of a property and an italicized lower case letter identifies a unit vector representing a

Table 1. Definition of Symbols Used in Describing Terms Related to Model and Simulation Fidelity.

<i>Symbol</i>	<i>Definition</i>
$\underline{\epsilon}$	the tensor representing the values of error associated with all of dependent variables of all of the dependencies in the set F
$\underline{\epsilon}_i$	the tensor representing the values of error associated with all of the dependent variables of all of the dependencies associated with the modeled or simulated object O_i
$\underline{\epsilon}_{ij}$	the vector in property space representing the values of error associated with all of the dependent variables of the jth dependency representing the behavior of the modeled or simulated object O_i
$\underline{\alpha}_{ij}$	the vector representing the minimum possible values of accuracy associated with all of the dependent variables of the dependency F_{ij}
ϵ_{ijk}	the value of error associated with the kth dependent variable of the jth dependency representing the behavior of the modeled or simulated object O_i
$\underline{\epsilon}_m$	the tensor representing the values of maximum error associated with all of the dependent variables over the acceptable ranges of all of the dependencies in the set F
$\underline{\epsilon}_{m\Delta l}$	the tensor representing the lower limits of the maximum error for all dependent variables in F_A that are acceptable in order to meet an application's requirements
$\underline{\epsilon}_{m\Delta u}$	the tensor representing the upper limits of the maximum error for all dependent variables in F_A that are acceptable in order to meet an application's requirements
$\underline{\epsilon}_{mij}$	the vector representing the maximum possible values of error associated with all of the dependent variables of the dependency F_{ij}
$\underline{\epsilon}_{mS}$	the tensor representing the maximum error associated with every dependent variable of every dependency in F_S that a model or

	simulation can produce
$\underline{\Delta x}$	the tensor representing the values of sensitivity associated with the independent variables of all of the dependencies in the set F
$\underline{\Delta x}_i$	the tensor representing the values of sensitivity associated with all of the independent variables of all of the dependencies associated with the modeled or simulated object O_i
$\underline{\Delta x}_{ij}$	the vector in property space representing the values of sensitivity associated with all of the independent variables of the jth dependency representing the behavior of the modeled or simulated object O_i
Δx_{ijk}	the value of sensitivity associated with the kth independent variable of the jth dependency representing the behavior of the modeled or simulated object O_i
$\underline{\Delta x}_m$	the tensor representing the maximum possible values of sensitivity associated with all of the independent variables over the acceptable ranges of all of the dependencies in the set F
$\underline{\Delta x}_{m\Delta l}$	the tensor representing the lower limits of the maximum sensitivity for all dependent variables in F_A that are acceptable in order to meet an application's requirements
$\underline{\Delta x}_{m\Delta u}$	the tensor representing the upper limits of the maximum sensitivity for all dependent variables in F_A that are acceptable in order to meet an application's requirements
$\underline{\Delta x}_{mij}$	the vector representing the maximum possible values of sensitivity associated with all of the independent variables of the dependency F_{ij}
$\underline{\Delta x}_{mS}$	the tensor representing the maximum sensitivity associated with every dependent variable of every dependency in F_S that a model or simulation can produce
$\underline{\Delta y}$	the tensor representing the values of precision associated with the dependent variables of all of the dependencies in the set F
$\underline{\Delta y}_i$	the tensor representing the values of precision associated with all of the

	dependent variables of all of the dependencies associated with the modeled or simulated object O_i
$\underline{\Delta y}_{ij}$	the vector in property space representing the values of precision associated with all of the dependent variables of the j th dependency representing the behavior of the modeled or simulated object O_i
Δy_{ijk}	the value of precision associated with the k th dependent variable of the j th dependency representing the behavior of the modeled or simulated object O_i
$\underline{\Delta y}_m$	the tensor representing the maximum possible values of precision associated with all of the dependent variables over the acceptable ranges of all of the dependencies in the set F
$\underline{\Delta y}_{mAl}$	the tensor representing the lower limits of the maximum precision for all dependent variables in F_A that are acceptable in order to meet an application's requirements
$\underline{\Delta y}_{mAu}$	the tensor representing the upper limits of the maximum precision for all dependent variables in F_A that are acceptable in order to meet an application's requirements
$\underline{\Delta y}_{mij}$	the vector representing the maximum possible values of precision associated with all of the dependent variables of the dependency F_{ij}
$\underline{\Delta y}_{mS}$	the tensor representing the maximum precision associated with every dependent variable of every dependency in F_S that a model or simulation can produce
\underline{C}_A	the vector of integer values representing the total number of unique instances of the objects in the set O_A required to meet an application's requirements
c_{Ai}	the integer number representing the total number of unique instances of the object O_i required to meet an application's requirements
C_{Ai}	the vector representing the minimum total number of unique instances of the objects in the set O_A required to meet an application's requirements

c_{Ali}	the minimum number of unique instances of object O_i required to meet an application's requirements
C_{Au}	the vector representing the maximum total number of unique instances of the objects in the set O_A required to meet an application's requirements
c_{Aui}	the maximum number of unique instances of object O_i required to meet an application's requirements
E	the set of characteristics describing the fidelity of a model or simulation of some aspect of reality
F	the set of modeled or simulated dependencies describing the coupling between the properties in Q
F_{Al}	the set of object dependencies representing the minimum set required to meet an application's requirements
F_{Au}	the set of object dependencies representing the maximum set required to meet an application's requirements
F_i	the set of modeled or simulated dependencies associated with object O_i in G
\underline{f}_{ij}	the actual dependency function of the dependency represented by the set F_{ij}
F_{ij}	the j th dependency in the set of dependencies F_i representing the behavior of the modeled or simulated object O_i
F_S	the set of object dependencies that a model or simulation can represent
G	the knowledge about the set of dependencies describing the couplings (and therefore the behavior) between the objects in R through their properties
G'	the knowledge about the set of all of the object dependencies in the real world
G_i	the knowledge about the set of dependencies describing the behavior of the i th object in R , the dependencies representing the behavior about a single object

\underline{g}_{ij}	the actual dependency function of the dependency represented by the set G_{ij}
G_{ij}	the knowledge about the j th dependency related to the i th object in R
\underline{N}_{ij}	an integer vector in property space representing the number of precision increments needed to represent the value of the \underline{Y}_{ij} .
k_{ij}	a constant value, associated with each dependency function f_{ij} , that scales that function's error to its accuracy
O	the set of objects represented by the models or simulations of the objects in R
O_{Al}	the set of objects representing the minimum set required to meet an application's requirements
O_{Au}	the set of objects representing the maximum set required to meet an application's requirements
O_i	the i th modeled or simulated object in the set O
o_i	the unit vector representing the modeled or simulated object O_i in O
O_S	the set of objects that a model or simulation can represent
P	the knowledge about the set of properties describing the state of all of the objects in R
\underline{P}	the tensor in object \times property space describing the state of all of the objects in R
P'	the knowledge about the set of all of the object properties in the real world
\underline{P}_i	a vector in object property space describing the state of the i th object, $P_i = \sum p_{ij} p_{ij}$
P_i	the knowledge about the set of properties of the i th object in R , the properties describing the state of a single object
p_{ij}	a unit vector representing the j th property of the i th object, a unique identifier representing a single property of a single object

P_{ij}	the knowledge about the j th property of the i th object in R
p_{ij}	the value of the j th property of the i th object in R , a single property of a single object
Q	the set of modeled or simulated properties describing the state of the objects in G
\underline{Q}	the tensor representing the state of the world O
Q_{Al}	the set of object properties representing the minimum set required to meet an application's requirements
Q_{Au}	the set of object properties representing the maximum set required to meet an application's requirements
Q_i	the set of modeled or simulated properties describing the state of the object O_i in O
q_{ij}	the value of the j th property of object O_i in O
q_{ij}	a unit vector representing the j th property of object O_i in O
Q_S	the set of object properties that a model or simulation can represent
R	the set of knowledge about real world objects, the referent
R'	the knowledge about the set of all of the objects in the real world
r_i	a unit vector representing the i th object in R , a unique identifier representing a single object
R_i	the knowledge about the i th object in R , the knowledge about a single object
S	the set of real world objects, the simuland
S_i	the i th object in the set S , a single object
T_A	the set of elements representing the tolerances of the requirements of an application
T_{Al}	the set of elements representing the lower limits of the tolerances of the requirements of an application

T_{Au}	the set of elements representing the upper limits of the tolerances of the requirements of an application
U	the set of properties in P representing all of the independent variables of all of the dependencies in G
U_i	the set of properties in P representing all of the independent variables of all of the dependencies associated with the object R_i
U_{ij}	the set of properties in P representing the independent variables of the j th dependency representing the behavior of object R_i
\underline{U}_{ij}	the vector in property space representing the values of the properties in the set U_{ij}
u_{ijk}	the value of the k th component of the vector \underline{U}_{ij}
V	the set of properties in P representing all of the dependent variables of all of the dependencies in G
V_i	the set of properties in P representing all of the dependent variables of all of the dependencies associated with the object R_i
V_{ij}	the set of properties in P representing the dependent variables of the j th dependency representing the behavior of object R_i
\underline{V}_{ij}	the vector in property space representing the values of the properties in the set V_{ij}
v_{ijk}	the value of the k th component of the vector \underline{V}_{ij}
W	the set of values describing the ranges of all possible values of all of the properties in the set P
W_i	the set of values describing the ranges of all possible values of all of the properties of the i th object in R
W_{ij}	the set of values describing the ranges of all possible values of the j th property of the i th object in R
w_{ij}	the lower limit value of the range of the j th property of object O_i
w_{uij}	the upper limit value of the range of the j th

	property of object O_i
X	the set of properties in Q representing all of the independent variables of all of the dependencies in F
X_i	the set of properties in Q representing all of the independent variables of all of the dependencies associated with the modeled or simulated object O_i
X_{ij}	the set of properties in Q representing the independent variables of the j th dependency representing the behavior of modeled or simulated object O_i
\underline{X}_{ij}	the vector in property space representing the values of the properties in the set X_{ij}
x_{ijk}	the value of the k th component of the vector \underline{X}_{ij}
Y	the set of properties in Q representing all of the dependent variables of all of the dependencies in F
Y_i	the set of properties in Q representing all of the dependent variables of all of the dependencies associated with the modeled or simulated object O_i
Y_{ij}	the set of properties in Q representing the dependent variables of the j th dependency representing the behavior of modeled or simulated object O_i
\underline{Y}_{ij}	the vector in property space representing the values of the properties in the set Y_{ij}
y_{ijk}	the value of the k th component of the vector \underline{Y}_{ij}
Z	the set of values describing the ranges of possible values of the properties in all of the objects in Q
Z_i	the set of values describing the ranges of possible values of the properties in the set Q_i
Z_{ij}	the set of values describing the ranges of possible values of the property Q_{ij} of the object O_i in O
\underline{Z}_i	the tensor that represents the values of the lower limits of the acceptable ranges of all of the properties describing the state of all

	of the objects in O
$\underline{Z}_{l\Delta l}$	the tensor that represents the values of the lower limits of the lower limits of the acceptable ranges of all of the properties describing the state of all of the objects in O suitable for a particular application
$\underline{Z}_{l\Delta u}$	the tensor that represents the values of the upper limits of the lower limits of the acceptable ranges of all of the properties describing the state of all of the objects in O suitable for a particular application
\underline{Z}_{li}	the vector that represents the values of the lower limits of the acceptable ranges of all of the properties describing the state of the object O_i
z_{lij}	the lower limit of the acceptable range of property Q_{ij} and
\underline{Z}_{lS}	the tensor that represents the values of the lower limits of the acceptable ranges of all of the properties describing the state of all of the objects in O as represented by a particular model or simulation
\underline{Z}_u	the tensor that represents the values of the upper limits of the acceptable ranges of all of the properties describing the state of all of the objects in O
$\underline{Z}_{u\Delta l}$	the tensor that represents the values of the lower limits of the upper limits of the acceptable ranges of all of the properties describing the state of all of the objects in O suitable for a particular application
$\underline{Z}_{u\Delta u}$	the tensor that represents the values of the upper limits of the upper limits of the acceptable ranges of all of the properties describing the state of all of the objects in O suitable for a particular application
\underline{Z}_{ui}	the vector that represents the values of the upper limits of the acceptable ranges of all of the properties describing the state of the object O_i
z_{uij}	the upper limit of the acceptable range of property Q_{ij}
\underline{Z}_{uS}	the tensor that represents the values of the upper limits of the acceptable ranges of all of the properties describing the state of all of the objects in O as represented by a

	particular model or simulation
--	--------------------------------