Simulation Fidelity Theory and Practice

A Unified Approach to Defining, Specifying
and Measuring the Realism of Simulations

Simulation Fidelity Theory and Practice

A Unified Approach to Defining, Specifying and Measuring the Realism of Simulations

PROEFSCHRIFT

ter verkrijging van de graad van doctor
aan de Technische Universiteit Delft,
op gezag van de Rector Magnificus, prof. dr. ir. J.T. Fokkema,
voorzitter van het College van Promoties,
in het openbaar te verdedigen

op dinsdag 18 januari 2005 om 13.00 uur

door

Zwerus Cornelis ROZA

ingenieur luchtvaart en ruimtevaart

geboren te Dussen

Dit proefschrift is goedgekeurd door de promotor:
Prof. dr. ir. J.A. Mulder

Samenstelling promotiecommissie:

Rector Magnificus,	Voorzitter
Prof. dr. ir. J.A. Mulder,	Technische Universiteit Delft, promotor
Prof. dr. ir. Th. van Holten,	Technische Universiteit Delft
Prof. dr. ir. P.A. Wieringa,	Technische Universiteit Delft
Prof. dr. D.K. Pace,	Johns Hopkins University, Laurel, MD USA
Prof. F.M. Cardullo,	State University of New-York, Binghamton, NY USA
Dr. G.J. Jense,	TNO-FEL, Den Haag, Nederland
Dr. R. Jacquart	ONERA, Toulouse, Frankrijk
Prof. dr. ir. M.J.L. van Tooren	Technische Universiteit Delft, reservelid

Dr. Jeroen Voogd van TNO-FEL heeft als begeleider in belangrijke mate aan de totstandkoming van het proefschrift bijgedragen.
Dr. Jeroen Voogd of TNO-FEL The Hague has provided substantial guidance and support in the preparation of this thesis

Een deel van dit promotieonderzoek is gefinancierd door en uitgevoerd in samenwerking met het TNO Fysisch en Elektronisch Laboratorium, Command & Control en Simulatie Divisie
A part of this research has been funded by and conducted in cooperation with the Netherlands Organization for Applied Scientific Research (TNO), Physics and Electronics Laboratory, Command & Control and Simulation Division.

Published and distributed by: DUP Science

DUP Science is an imprint of
Delft University Press
P.O.Box 98
2600 MG Delft
The Netherlands
Telephone: +31 15 27 85 678
Telefax: +31 15 27 85 706
E-mail: info@library.tudelft.nl

ISBN 90-407-2569-1

Keywords: 3

Printed in The Netherlands

To my parents

'Though the course may change sometimes, rivers always reach the sea'
(Robert Plant, Physical Graffiti)

Preface

This Ph.D. thesis lying in front of you is the final result of a cooperative research project between the Delft University of Technology (DUT), Faculty of Aerospace Engineering, Control and Simulation division and the Netherlands Organization for Applied Scientific Research (TNO), Physics and Electronics Laboratory, Command & Control and Simulation division on the topic of fidelity assessment of simulation systems. The author has conducted this research project in the period of January 1997 until August 2004 at both institutes. In that period the author served as a lead-team member of the Simulation Interoperability Standards Organization (SISO) Fidelity study group. Which is an international research group to the topic of simulation fidelity and related simulation system development and validation practices, particularly in the context of distributed simulation systems. Although many views and ideas in this thesis are the result of the active participation within this study group, the thesis should not be construed to represent views of the group or any of its members. The official view of this group has been published in two reports that were (co)-authored by this author. Furthermore, several other parts of this thesis have been published in various conference papers and internal TNO/TUD reports. The interested reader in these publications is referred to the reference section for more details.

This thesis is intended for modeling and simulation users, engineers, researchers and students who are looking for knowledge and guidance on fidelity theory and practice within the modeling and simulation enterprise. It provides a general applicable and unified fidelity theory and application framework for assessing simulation fidelity in a systematic and formal manner. The framework has been developed from a non-specific application or problem domain context in order to be of use for broad user community. As a result of this, however, the user may have to tailor some of the generic elements in the framework to better suite their own specific application and problem domain needs. The unified fidelity framework is presented in such a format that it can easily be translated by the reader into a set of top-level functional requirements for the development of an automated tool suite, which supports fidelity assessment activities during the simulation system life-cycle.

Some parts of this thesis, due to the chosen generic and non-specific application or problem domain approach, have a rather abstract and theoretical character. Nonetheless, I hope this thesis will provide the reader a better understanding of simulation fidelity and assists the reader in addressing simulation fidelity issues in a more formal and systematic manner. Particularly, since there is hardly any other literature or standard available on this hard but very important aspect of modeling and simulation. This thesis still leaves many fidelity issues open that need to be addressed and researched. Like the Chinese saying says it is better to light a candle than complaining about the darkness.

Z.C. (Manfred) Roza
December, 2004

Summary

Simulation fidelity is an intrinsic element of any simulation system, one that all its developers and users have to deal with one way or the other. It is commonly recognized by the modeling and simulation community that simulation fidelity is an essential vehicle in properly assessing the validity and credibility of simulation results. Furthermore, fidelity is one of the main cost-drives of any model or simulation development. Today simulation systems play an increasingly important role in our society, which are rapidly becoming the primary tool for crucial decision-making processes during engineering design, test and evaluation of new systems, even safety-critical systems, and in training of people operating these systems. With this increasing reliance on simulation results it is more than ever important to know how well a simulation corresponds to reality in order to ascertain that the risks involved in using the simulation results are within acceptable limits.

Despite these observations and the enormous advancements in simulation hardware and software, the ability to characterize, qualify and quantify the level of simulation fidelity is still a largely uncultivated area. An area in which there exist many incomplete, inconsistent and widely scattered views, concepts and approaches to fidelity. What is primarily lacking is the absence of a systematic and general applicable simulation fidelity assessment methodology, which is based on a sound unifying theory for fidelity and associated practices.

This thesis tries to fill this void by the analysis, extension and integration of existing simulation fidelity approaches into a single unified fidelity theory and practice. All this is done from a general simulation system life cycle perspective, not limited by any specific application or problem domain aspects.

In order to develop a unified fidelity theory a comparative analysis of existing fidelity theories and practices is required. This thesis identifies the major similarities, differences, problems and limitations of a representative portion of pioneering and contemporary fidelity research found in literature. The results from this comparative analysis serve as the first basis for the developed unified fidelity framework in this thesis. Without a contextual modeling and simulation framework it is hard to develop a fidelity theory and practice that can seamlessly be integrated within the development and validation process of simulation systems. Therefore, in this thesis a general modeling and simulation contextual is discussed that serves as the second basis for the unified fidelity framework.

The foundation of the developed unified fidelity framework comprises a precise mathematical formulation of the term fidelity and the fundamental concepts underlying its characterization and measurement. The real-world reference knowledge standard paradigm is the most fundamental element in this framework. This so-called fidelity referent formalizes the natural level of indirection of fidelity measurement i.e. one can actually never measure against reality itself but against an approximated interpretation of reality. By explicitly linking the real-world knowledge error and uncertainties to its structure, the fidelity referent transforms this insolvable problem of 'exact' fidelity assessment into a practical evidence-based assessment approach of simulation fidelity. The other key element in here is the simulation system knowledge specification concept.

Both concepts form the basis of the pragmatic measurement and specification of simulation fidelity. A possible practical implementation for each concept is proposed in terms of a generic knowledge-base architecture consisting of a set of well-structured specification templates. Furthermore, a set of associated mathematical formalisms is developed to support these both knowledge-base structures.
Having formally defined a fidelity referent and the simulation system knowledge base fidelity assessment becomes the assessment of the measured inverse differences between pairs of specified real-world and simulation knowledge. Since simulation fidelity has a multidimensional and multifaceted character, it is best qualified and quantified by an enumeration of various kind of metrics instead of a single measure. A taxonomy is presented in this thesis containing the most basic and common fidelity measurement methods and metrics, which can be used for this purpose. This taxonomy is a combination of a set of newly developed and existing methods and metrics available in literature.
The thesis introduces the concept of fidelity requirements, as a means for the formal and systematical specification of the level of fidelity that is required to meet the user needs. From this concept a fidelity-based simulation verification and validation process is developed. A multi-criteria analysis approach that evaluates alternatives based upon their fidelity performance and effectiveness scores is proposed to address fidelity issues in the comparisons of simulation systems, suitability and trade-off decision-making process.
The unified fidelity framework is completed with a fidelity management process model outlining a series of generic stages, activities and tasks, which together provide a structured but generic approach to properly integrate and apply all other unified fidelity framework elements in the simulation system development and validation process. Two aerospace simulation case studies have been conducted with the unified fidelity framework. The results of these case-studies haven been used to refine the unified fidelity framework and are also used to illustrate the major elements of this framework throughout the whole thesis.

Although, limited in scope these case studies demonstrated that the unified fidelity framework and underlying concepts and paradigms prove to be a promising and viable basis for a future standard fidelity theory and practice. Major benefits experienced in both case studies include a better definition of what, how and when fidelity assessment activities have to be performed and the specification of more clear simulation system requirements. It also has a positive effect on trade-off and priority decision-making during simulation system development. Further, it facilitates more efficient elicitation and organization of real-world and simulation data. The unified fidelity framework also enables an easier and systematic identification of sources causing large and unacceptable fidelity discrepancies, and defining suitable strategies and solutions to solve these issues.
According Murphy's Law every benefit comes with a disadvantage. The most important drawback of the unified fidelity framework lies in the inherent multidimensional and multifaceted nature of simulation fidelity. In practice this causes fidelity assessment to become a very complex, time consuming and hard to be handle activity by hand. Which is something also any other rigorous fidelity methodology will encounter. Therefore, the development and use of a general purpose or domain tailored automated tool-suit to assist simulation developers and validation agents is indispensable for a cost-effective application of a formal fidelity assessment processes within the model and simulation enterprise.

Rigorous assessment of fidelity is one of the most difficult and hard to grasp issues of the model and simulation enterprise. Substantial and exhaustive research endeavors in this area are very limited. Due to this, simulation fidelity still remains a hardly touched upon and rather uncultivated area. When considered from this perspective the major contribution of is thesis to the modeling and simulation community is the fact that it brings all aspects of simulation fidelity together within a single formal fidelity theory and application framework. The presented unified fidelity framework is thought to contain most of the essential elements for the development of a common standard and widely accepted fidelity theory and practice.

Contents

1 Introduction

1.1 Why and What is Fidelity?

Since the early development of simulation technology, the notion of fidelity has been an apparent and recognized issue by the modeling and simulation (M&S) community in properly assessing the validity and credibility of simulation results. Furthermore, fidelity is found to be one of the main cost-drivers of any model or simulation development. As a general rule, the higher the fidelity the more time and resource consuming the simulation development is. Thus, being able to state what level of fidelity is exactly required avoids unnecessary investments, superfluous simulation components and unusable simulation. Fidelity is therefore an intrinsic element of any simulation system, one that all simulation developers and users have to deal with one way or the other. Despite these observations and the enormous advancements in simulation hardware and software, the ability to characterize and quantify the level of simulation fidelity is still a largely uncultivated area [47] [105] [125]. From these publications it can be concluded that the current fidelity adjectives mainly express fidelity in rather subjective and qualitative terms within the purview of commercial reasons instead of technical utility.

This raises the question of what is meant with the term fidelity in order to be able to judge the value of such claims as "high fidelity" simulations. However, this is not an easy question to answer. A closer look of the usage of the term *fidelity* shows that fidelity is a nebulous term used for different types of concepts [125]. Even though the term fidelity can be loosely translated as simulator goodness or faithfulness to reality, it lacks a uniform definition and common accepted practice. To illustrate this, consider the following observation made by Gross et. al. [48]: *"Fidelity has always been the subject of heated discussion and is almost in the same category of what is said about politics and religion. When simulationists attempt to tackle fidelity, disparity of positions and the fervor with which they are held leads to either believers or disbelievers in the meaningfulness of fidelity. Like many terms applied to complex fields of study, fidelity assumed a complicated and contorted persona that appears as an elusive multi-headed monster poised ready to completely consume that dare to pass nearby"*. To deal with this complexity of fidelity the following two quick solutions are common practice in the M&S community [125]; first ignore fidelity issues as much as possible and second implement 'all you can afford' levels of fidelity in a simulation system. Obviously, the first solution isn't an answer to the fidelity problem at all. The second solution usually yields utilizing the most expensive and latest simulation hardware and software technology, which is not really a cost-effective solution to tackle the fidelity problem. Evidence from the simulator based training community proves that this approach to simulation fidelity can even degrade the training effectiveness [1] [57] [58] [77].
Currently, there is a renewed interest and call for more robust and widely accepted methods to assess simulation fidelity. Why is this? Until a decade ago there was no real need for such methods because simulations were used on a smaller scale and primarily

developed in-house to tackle some minor aspects of a specific problem. However, the M&S community starts to realize that the currently used subjective and ad-hoc fidelity practices can no longer fulfill the demanding requirements of today's simulation applications. The increasing dependency on simulation results, the greater complexity and more frequent reuse of simulations across different application domains have caused a desire for new more comprehensive fidelity methods. This increasingly important role of simulation in our society can be seen in the heavy usage of simulations for crucial decision-making processes during engineering design, test and evaluation of new systems, and in simulator training of people operating these systems. As an example, nowadays the concept of zero flight time training of new airline pilots through flight simulator training becomes widely accepted training method instead of the more expensive and less safe training in the real aircraft [118].
With the increasing reliance on simulation results, perhaps as the only available tool for a certain purpose, it is important to know how well a simulation relates to reality, i.e. the level of fidelity, in order to guarantee the validity and credibility of the simulation results. If there are no or limited resources spend to simulation validation in order to support the credibility of a simulation, decisions based on the results of that simulation become highly questionably and involve a high level of risks. Therefore, systematic and comprehensive fidelity methods should be seen as a critical part of simulation validation. These concerns have been expressed by a wide variety of user communities. However, this perspective doesn't always prevail in the presence of strong interests promoting M&S acceptance [108].

With the advent of modern distributed simulation technology, such as the US Defense Modeling & Simulation Office (DMSO) High Level Architecture (HLA), a whole new range of fidelity concerns have been expressed that need to be addressed before the full potential of distributed simulations can be utilized [10] [14] [47] [51] [125]. Distributed simulation research has primarily focused on the technical interoperability of simulations, solving problems regarding the capability of simulations to physically interconnect via a certain communication infrastructure and effectively exchange data in accordance with a set of rules, data-formats and interface specification. HLA has resolved most of the problems concerning the technical capability to network together simulations and its concept is now proven for various kind of applications. There are, however, still many unresolved issues regarding the ability of simulations to inter-operate in a logically meaningful manner. One of the major concerns in development and validation of distributed simulations is the capability to qualify and quantify the 'overall' simulation fidelity. Demonstration of technical interoperability, however, is a necessary but not a sufficient condition to guarantee a credible and valid distributed simulation [119]. In his paper on fidelity and simulation interoperability Harmon formulates the problem as follows: *"The simulation community understands relatively little about the phenomena of interoperability beyond very basic communications and database issues. This community understands even less about the dependencies of simulation fidelity upon interoperability. Without an understanding of these most fundamental phenomena, constructing large scale distributed simulation exercises will continue to be complicated and unreliable with unpredictable results. Thus, risk will be hard to predict and manage and the benefits of distributed simulation will be difficult, if not impossible in some cases, to realize"* [51]. Besides interoperability, fidelity also plays an important role in enabling reuse. Reuse of one or more previously created simulation systems and components is a regular practice to reduce simulation development time and costs. In fact, the ability to reuse existing simulation systems or components is a key objective today within the M&S community. However, making a good decision about the reusability of a set of candidate simulations for the current application basically depends on two things. The first thing is the ability to specify the required fidelity levels of each simulation to fulfill the application purpose. The second thing is the ability to specify the level of fidelity of a reusable simulation system and its models.

A recent simulation development effort provides a real-life example of the mentioned fidelity problems regarding simulation interoperability and reuse [148]. This example describes a distributed simulation federating legacy simulations with the intent to evaluate marine combat systems. One of the scenarios to be represented by this environment includes collective operation of several ship platforms executing a co-operative defensive action against incoming air-threat. Despite the fact that a good model of the data-link between the platforms was available, overall fidelity problems were encountered because of the different fidelity levels of the radar models used by each platform. Several radar systems did not take into account all the environmental conditions in the scenario that limited target detection. Some radar systems could see the target while others did not, which resulted in severe conflicting radar readings communicated among the platforms, causing an unrealistic situation, not at all representative for the actual systems under study.
The problem in this example was caused by combining simulations, each of which having a fidelity perfectly acceptable for its original purpose, into a configuration to serve a new purpose. Which is the whole idea of simulation interoperability and reuse, but also stresses the importance of equipping simulation developers with the means to anticipate and correct these fidelity related problems of simulation interoperability and reuse in a structured manner. In order to properly apply distributed simulations to civil aviation and other safety-critical applications, such capabilities are mandatory.

These simulation system interoperability and reuse concerns have boosted the renewed interest in research to more comprehensive simulation fidelity assessment methods for the simulation development and validation process, in order to obtain a simulation that produces the desired and reliable (realistic) results against acceptable costs and development time. Not only in the context of HLA but also simulation in general. And has resulted in the establishment of special fidelity interest groups such as the Simulation Interoperability Standards Organization (SISO) fidelity study group [47] [126].

1.2 Fidelity Research Project: Origin, Method and Objectives

In the mid nineties the Delft University of Technology Aerospace Control and Simulation (DUT-C&S) division participated in a joint project of Dutch simulation industry and research institutes called SIMULTAAN [11]. SIMULTAAN was a two and half year project, which brought together knowledge and experience in the area of vehicle simulators and distributed simulation available in The Netherlands at that time. For DUT-C&S this was its first large-scale experience with the application of distributed simulation and HLA in specific. The purpose of this project was twofold. First, establishing a permanent intellectual infrastructure and strengthened working relationships between the participants. Second, the development of generic distributed

simulator architecture for the rapid development and interoperation of a wide range of simulators, including manned mock-ups of vehicles, full flight simulators and unmanned simulators.
The fundamental concept behind this SIMULTAAN Simulator Architecture (SSA) is the extension of the HLA interoperability and reuse principles from a simulation system interconnection level to the level of the simulation system components themselves. This is known as a component-based simulator architecture. In such architectures, the individual simulator is considered to be composed of various components (motion systems, visual systems, mock-up systems, vehicle dynamics models, etc.) interacting through a distributed data-exchange infrastructure similar to the simulator interconnection levels. Component-based simulator architectures are intended to maximize the reuse potential of component technology by using a standard interface for simulator components and standard component repositories. In this way the simulator development time and costs can be significantly reduced. The developed SSA facilitated interoperability between both simulator components as well as among other simulators in a fully HLA compatible manner. A successful proof of concept of the SSA has been presented in a large search & rescue scenario demonstration during the summer of 1999 [11].

Stimulated by this success both DUT-C&S and the Netherlands Organization for Applied Scientific Research, Physics and Electronics Laboratory (TNO-FEL), the SIMULTAAN project leader, continued the research and application of distributed simulation technology within their own laboratories to tailor the SSA to their specific needs [61] [111]. From this continued research and applications of distributed simulation technology, concerns regarding fidelity in relationship to simulation system credibility and interoperability emerged. Concerns similar to those expressed elsewhere in the M&S community (Section 1.1). These concerns were drivers for TNO-FEL to initiate and sponsor a fundamental research project on the subject of simulation fidelity in cooperation with the DUT-C&S.

In this thesis the results of this fundamental research project to simulation fidelity are presented. One of the objectives in this research was to develop a clear overview of the various perspectives and available knowledge of simulation fidelity. This very early research immediately showed that fidelity assessment of simulation systems in general is still in an embryonic state and at present there is no widely accepted methodology available [125]. Although this early research confirmed there always has been and still is a structural need for a robust simulation fidelity theory and practice, history demonstrates that fidelity research is characterized by a recurrent process of sudden revival followed by an equally rapid decline with the focus on an instantaneous and isolated solutions for a given simulation fidelity problem at hand. As a result the M&S community is left with an incomplete, inconsistent and scattered set of views and approaches to simulation fidelity. Therefore, what is needed most is a unified framework for understanding and applying fidelity. A framework which formally defines fidelity and its related aspects, practical fidelity concepts and measurement approach, and their application relationships with respect to the M&S enterprise [122]. The majority of this thesis is devoted to this subject with the emphasis on the unification and integration of simulation fidelity approaches, either existing or new ones under development, within a single and general applicable framework. To achieve this objective, this fidelity research project has been synchronized with the fidelity research activities of the Simulation

Interoperability Standards Organization (SISO) fidelity study groups. For a period of two years this fidelity research activity has been executed in a close cooperation with that of the SISO fidelity study groups. During that period many research results have been shared or were otherwise developed collectively. The results of this fidelity research have been published in two reports for the Simulation Interoperability Standards Organization [47] [126].

Since the aim of this thesis is the development of a general applicable and unified simulation fidelity theory and associated practices, this fidelity research project, unlike the mainstream of known fidelity research efforts, adopted a non-specific problem or application domain approach. As a result of this, the unified fidelity framework developed here has a fairly high abstraction level and a rather theoretical character. For that same reason the unified fidelity framework does not provide a ready-made fidelity answer for any specific simulation application or problem. Instead the unified fidelity framework has been developed such that it provides a generic basis for any simulation fidelity assessment process, which can be tailored or extended with any particular discipline or subject specific methods to suite the specific needs of the simulation application at hand. Working from such a single unified fidelity framework as a standard basis will improve the efficiency, repeatability, understandability and reusability of any simulation fidelity assessment process and its results. To demonstrate this actual practical usage and to provide a conceptual proof of this unified fidelity framework approach proposition, two practical simulation case studies with this unified fidelity framework have been conducted in the aerospace problem domain.

1.3 Overview of the Thesis

This thesis is structured in three main parts: simulation fidelity background and application context, the development of the unified fidelity framework itself, and the application of the unified fidelity framework application within two aerospace simulations. *Figure 1-1* presents a graphical overview of this structure and the logical relationships between each chapter.

The first part of this thesis starts in Chapter 2 with an overview of existing fidelity theories and practices found in literature. An effort is made to cite a representative portion of pioneering and contemporary fidelity research. This overview is followed by a comparative analysis of these fidelity theories and practices to identify their similarities, differences, problems and limitations. The results of this analysis serve as the bases for the development of the unified fidelity framework outlined in the second part of this thesis.

Since fidelity is an inherent element of any model and simulation, its research efforts must always be considered and rooted in a contextual modeling and simulation framework [126]. Without such a framework it is hard to develop a useful fidelity theory and practice that can seamlessly be integrated within the development and validation process of simulation systems. Chapter 3 introduces the fundamental modeling and simulation terminology, concepts, processes and mathematical formalisms, which together outline the modeling and simulation application context used for the fidelity research presented in this thesis.

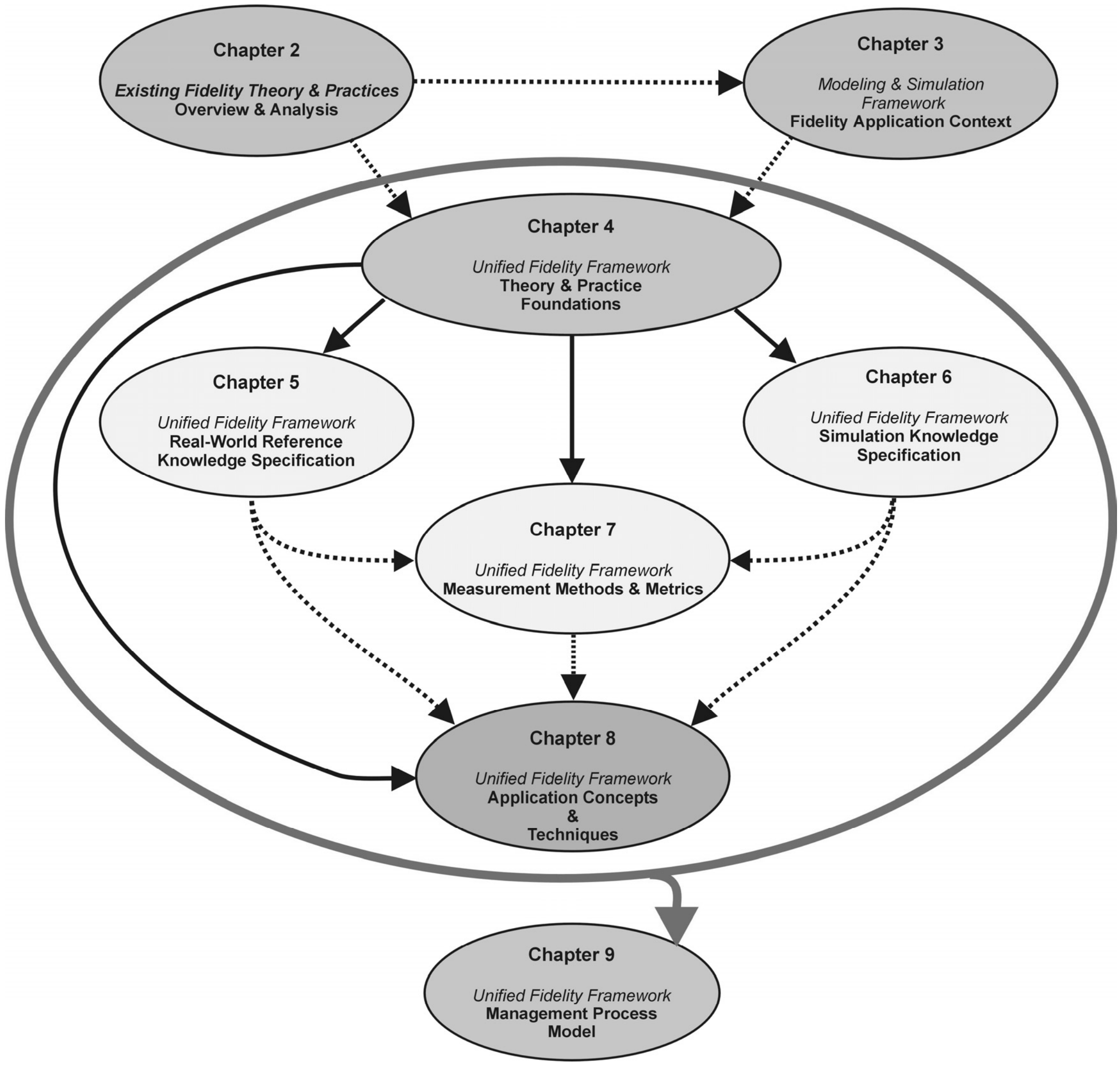


Figure 1-1 Graphical Overview of the Thesis Structure

The unified fidelity framework developed in part two of this thesis addresses the areas of fidelity definition, specification, measurement and application. In Chapter 4 the fundamental simulation fidelity principles and concepts are developed by means of synthesizing existing fidelity knowledge into a single consistent theory for simulation fidelity. This fidelity theory comprises a precise formulation for the term fidelity and the essential concepts underlying its characterization and measurement along with a set of mathematical formalisms. This formal definition of simulation fidelity is completed here with the key concepts and additional formalisms necessary to properly apply this fidelity theory within the modeling and simulation enterprise.

Chapters 5 and 6 are entirely devoted to the key elements of the unified fidelity framework, respectively the fidelity referent and simulation system knowledge-base concepts. Both concepts form the basis of pragmatic measurement and specification of simulation fidelity. In these chapters a possible implementation for each concept is proposed in terms of a generic knowledge-base architecture consisting of a set of well-structured specification templates. Furthermore, a set of associated mathematical formalisms is developed to support both these knowledge-base structures. Having formally defined a fidelity referent and the simulation system knowledge base it becomes possible to measure and specify the fidelity of a simulation system against this referent. Therefore, Chapter 7 continues with an overview of the most basic and common fidelity measurement methods and metrics, which are a combination of a set of newly developed and existing methods and metrics available in literature.

Chapter 8 combines the results of the previous four chapters to develop several major derived simulation fidelity application concepts. The first concept is the formal specification of the level of fidelity that is required to meet the user needs. These simulation fidelity requirements form the basis for the development of the fidelity-based simulation verification and validation concepts also outlined in this chapter. A multi-criteria analysis approach is adopted to construct simulation fidelity performance and effectiveness metrics, which can be utilized in the comparisons of simulation systems, suitability assessment and trade-off decision-making. The unified fidelity framework is in Chapter 9 completed with the fidelity management process model. This process model outlines a series of stages, activities and tasks, which together provide a structured but generic approach to properly integrate and apply all other unified fidelity framework elements in the simulation system development and validation process.

Finally in chapter 10 of this thesis the new contributions of this work to the area of simulation fidelity theory and practice are discussed together with conclusions that can be drawn from this fidelity research project. From this discussion a set of recommendation will be formulated to guide future theoretical and experimental research to simulation fidelity theory and practice.

The fidelity management process served as the basis for aerospace simulation case studies into the practical application of the unified fidelity framework within the model and simulation enterprise. The first case study has been the development of a HLA-based distributed simulation environment for future air-traffic control & management (ATC/ATM) concepts research purposes. A second case study that has been conducted comprises the development of an aircraft simulation model for a CN235 level D training flight simulator. Results from both these case studies have been used to test and refine the unified fidelity framework. Further, results and samples from these case studies are used throughout the thesis to illustrate the major elements of the unified fidelity framework. More details on these case studies can be found in Appendix C.

2 Analysis of Existing Fidelity Theories and Practices

2.1 Introduction

Concerns about simulation fidelity are as old as the modeling and simulation practice itself. Many publications touching on this topic have been published. This chapter presents a survey of these publications and an analysis of the common themes and problems of the fidelity theories and practices. A first analysis shows that the thoroughness of the reviewed publications varies significantly. Most of the available publications touch upon fidelity in an en passant manner. Therefore this chapter is confined to the discussion of those publications that are considered to provide some major insights on the topic of simulation fidelity. A more detailed discussion of this analysis is described in [125]. The results of this analysis presented in this chapter serves as the bases for the fidelity research, theory and practice outlined in the remaining chapters of this thesis. Most of the results have already been published elsewhere in bits and pieces over the years but are summarized and combined here to provide a consistent overall view of the existing fidelity practices [119] [121] [122] [125] [126].

This chapter is organized in four sections. Section 2.2 describes the early fidelity work and results from the training simulation domain. The next section focuses on the research on fidelity issues of distributed interactive simulations during late 1980's and early 1990's. An overview of contemporary research for both the unitary and the (HLA-based) distributed simulation[1] perspective is presented in Section 2.4. Finally, Section 2.5 gives a comparative analysis of the presented existing fidelity theories and practices to identify their similarities, differences, problems and limitations.

2.2 Early Fidelity Research from the Training Simulation World

The training application domain is the first major domain that employs simulation technology to a great extent. In a training system the major objective is to create training situations that have sufficient similarity to the real operational situation or devices in order to provide the most efficient training for the trainees. Therefore, it is logical that most of the early simulation fidelity research originates from this application domain, especially from pilot training flight simulator domain. This section describes the early fidelity research and achievements in the training simulation domain on the basis of three important publications.

[1] For the definitions of unitary and distributed simulation see Section 3.2 on basic terminology

2.2.1 AGARD Advisory Report on Fidelity of Simulation for Pilot Training

NATO's Advisory Group for Aerospace Research and Development report no. 159 "Fidelity of Simulation for Pilot Training" published in 1980 is the first comprehensive publication ever on the subject of simulation fidelity [1]. The origin of this report lies in the fact that the level of fidelity has significant impact on the development and operational cost of training simulators. Therefore the following issue must be addressed: 'What level of fidelity is required to provide appropriate pilot training against the lowest development and operational costs'. Addressing this last issue is the underlying objective of this report.

Simulator Fidelity Definition and Concept

The concept of fidelity outlined in the report is based on two defined classes of simulator cues. These are the equipment cues (duplication of appearance and feel of the operational equipment or aircraft) and environment cues (duplication of the environment, visual out-of-the window, sound and motion etc.). Fidelity in this report is characterized as the degree to which these two cue classes match those of the real aircraft with a distinction made between objective and human perceived cues. Based on this characterization the following types of fidelity are defined in the report:

- *Objective fidelity*: The degree according to an engineering viewpoint to which a simulator would be observed to reproduce its real-life counterpart, the aircraft, in terms of its appearance, substance and behavior as were sensed and recorded by an non-physiological instrumentation system onboard the simulator.
- *Perceptual fidelity:* The degree according to a psychological/physiological viewpoint to which the trainee subjectively perceives the simulator reproduction of its real-life counterpart, the aircraft, in the operational task situation.

Training Simulator Fidelity Specification Issues

A general observation made in the report is that simulators are usually developed under the concept that simulator training effectiveness equates to its realism. Thus the objective is to achieve the highest degree of realism possible for the represented aircraft appearance and behavior. The main reason for this approach is the fact that it is simple to state the design requirements and easy to obtain user acceptance. According to the report, effective training is doesn't necessarily imply the usage of high-fidelity simulators. Experience given is this report even shows that sometimes the opposite is true. Furthermore, there is no real justification for the high costs associated with such an approach.

Based on these observations the fidelity approach proposed in the report states that the training objectives should form the groundwork from which the actual simulator design requirements, including the fidelity requirements, have to be specified. Fidelity requirements for simulator parts cannot be determined strictly from the physical models of the aircraft and the environment in which it operates. It states that the intended role of the simulator in the training system and specific training objectives it should fulfill are of greater importance. Only after these training objectives have been defined the question of the required level of fidelity can be addressed properly. Therefore, the goal of the report is to provide information on the effect of simulator fidelity on training capabilities in such a way that developers can make informed choices on cost versus training effectiveness when specifying requirements for new simulators. This implies that there exists no unique answer to the question of how much fidelity is required and that the answer must be assessed for each specific training application at hand.

Three different viewpoints on simulator fidelity and requirements are presented in the report: training specialist's view, physiological specialist's view, and the simulator engineering view. The first view discusses how to develop training requirements and to translate these into simulator fidelity requirements. It also discusses how to assess the simulator effectiveness in the training system. The physiological view describes primarily how humans perceive motions and how motion cues can and should be generated to induce motion sensations that help to achieve the required perceptual fidelity. The simulator engineering view gives a whole range of simulator characteristics that determine the objective simulation fidelity and the maximum level of objective fidelity that could be achieved with the existing simulator technology in those days. Such simulator characteristics include motion system performance (bandwidth, acceleration limits etc.), visual system performance (field of view, detail, etc.), flight-deck mock-up look and feel.

Fidelity Requirements Specification Framework

Using the previous discussed fidelity concepts and specification issues, a high level framework for developing the proper perceptual and objective simulator fidelity requirements is presented in the AGARD report. This framework comprises the next two successive stages:

1. *Analyze training requirements and objectives.* Here, it is assumed that each mission can be broken down in several flight phases. Each phase is divided into the tasks that need to be performed in this phase. It is then determined which tasks need to be trained and what training techniques need to be used.
2. *Define methods and facilities to perform training.* First define objective cues that would be experienced in real aircraft while performing the task being trained. Next, define the perceptual cues experienced by the operator in the reality. Then determine the perceptual cues for the fidelity level required for each of the tasks to be trained. Finally, map the perceptual cues to the required objective cues for correct level of fidelity of the physical hardware and software characteristics of the simulator.

To perform the mapping at the end of stage two it is necessary to have a well-defined array of simulator parameters or characteristics on which to base these physical qualities. It also requires the knowledge of human physiology to determine how much objective fidelity is required to achieve a required level of perceptual fidelity in relationship to adequate training. Available research data should be consulted to help determining what cues are essential for training. Although, the focus of the report is on specifying the required level of fidelity for training flight simulators, it states that such requirements can significantly differ from requirements for flight simulators used for other purposes than training.

2.2.2 Hays' Training Simulator Fidelity Perspective

Robert Hays' early conceptual ideas on training simulator fidelity are described in the technical report for the US Army Research Institute for Behavioral and Social Sciences as published in 1980 [57]. These conceptual ideas are further developed and formalized in 1987 in his book, “Simulation Fidelity in Training System Design: Bridging the Gap between Reality and Training” [58]. The book contains a conceptual framework for considering fidelity in training simulator development as well as an extensive description and list of references on training system fidelity issues.

The Need for Training System Design Guidance

Hays notes that, currently, training system development focuses too much on the simulator design and technology at the expense of the actual goal of the training system, improving the job performance of the trainee or transfer of training. This often results in too expensive and unnecessarily high fidelity simulators, which do not fulfill all user training needs satisfactorily. Hays states that each application domain may have its own specific fidelity requirements, which serve their simulation purposes and objectives best. Therefore a general process is necessary to provide a detailed guidance for training simulator requirements and design, including fidelity, in order to be able to meet the training objectives effectively.

Simulation Fidelity Definition and Concept

A workable definition of fidelity, according to Hays, should be defined in terms of a domain of interest, relative to something else and in a measurable form. A training simulation domain oriented definition for fidelity is given by Hays: *“the degree of similarity between the training situation and the operational situation, which is simulated”*. Fidelity is presented as a two dimensional measurement in physical and functional characteristics. Physical characteristics address aspects like look and feel, while functional characteristics address aspects such as the informational, operational knowledge, and stimulus and response options. Fidelity is thus characterized as a summarizing descriptor of the overall training device characteristics and the trained scenarios.

Overview of the Fidelity Analysis Procedure

Based on the need for training design guidance a fidelity analysis procedure is proposed in Hays’ book. This procedure forms a conceptual bridge between the operational situation and the training situation to achieve an optimal and cost-effective training system. Furthermore it determines the fidelity requirements for the training device or simulator. From a high-level view this procedure consist of the following sequential steps:

1. Describing the operational situation in its functional and physical characteristics.
2. Performing a successive task and fidelity analysis to map the operational characteristics onto the required physical and functional aspects of each task.
3. Mapping the required physical and functional aspects onto the physical and functional fidelity requirements for the simulator components or equipment, and onto non-equipment centered physical aspects of the task.
4. Summarizing the requirements into the simulator’s physical and functional fidelity configuration.
5. Expand the fidelity configuration with instructional techniques.

Task and Fidelity Analysis

An operational situation description and task analysis are the first steps in the fidelity analysis procedure to determine the requirements for the entire training situation. The next step in this procedure is the fidelity analysis. Fidelity analysis uses the results of the task analysis as its input and determines the required physical and functional characteristics in order to provide the most cost-effective training. In order to do so, the fidelity analysis should be based on the best empirical data available on the relationships between fidelity of simulator configurations and training effectiveness. The information obtained from the task analysis is often difficult to extract and use in defining fidelity requirements. The fidelity analysis step should organize and document this information in a format, which is useful for simulator designers.

Formal Fidelity Metrics

Hays' developed a set of quantitative fidelity metrics that are used in the fidelity analysis procedure to predict how fidelity changes will effect the training outcome and to evaluate alternative system designs. According to Hay's fidelity has two major dimensions, physical and functional. Therefore the optimal training situation fidelity for a given task *x* can, according to Hays, be described as:

$$TSF_x = f\left[a\left(PhyF\right)_x + b\left(FuncF\right)_x\right] \tag{2.1}$$

where: TSF_x is the training situation fidelity for task *x*, $a(PhyF)_x$ is the weighted physical fidelity requirements function, $b(FuncF)_x$ is the weighted functional fidelity requirements function. These weight functions are themselves a function of weight functions of other fidelity requirements and can be written in a more mathematical fashion as follows:

$$b\left(FuncF\right)_x = f\left[b\left(inf\right)_x + b\left(equip\right)_x\right] \tag{2.2}$$

The weighted functional fidelity requirements function to train task *x* thus consists of both informational ($a(inf)_x$) and the equipment ($a(equip)_x$) functional requirements (*2.2*). These variables are weighted and could be further decomposed. However Hays doesn't show how these functions are precisely constructed.

$$\begin{aligned} a\left(PhyF\right)_x = f\Big[& a\left(task\ chars\right)_x + a\left(trainee\ chars\right)_x + \\ & a\left(instructor\ chars\right)_x + a\left(instructional\ strategies\right)_x + \\ & a\left(recources\right)_x + a\left(N\ other\ variables\right)_x \Big] \end{aligned} \tag{2.3}$$

The physical fidelity requirements function can be decomposed as in equation *(2-3)*. In here the $a(task\ chars)_x$ are the specific task characteristics to be trained, $a(trainee\ chars)_x$ are the specific characteristics of the trainee to be trained, $a(instructor\ chars)_x$ are the instructor characteristics which assist in the training, $a(instructional\ strategies)_x$ are the instructional strategies applied during training, $a(resources)_x$ are the available resources for the training system and the $a(N\ other\ variables)_x$ element represents all other aspects that contribute to a task specific physical fidelity requirement. The weight-factors in the equations depicted above have to be empirically determined and validated. Again Hays doesn't specify how these sub-weighting physical functions are composed in terms of specific characteristics and weight factors. Altogether, this makes the Hay's fidelity formulation rather vague since he doesn't provide useful definitions for each subterm.

2.2.3 Airplane Flight Simulator Requirement Standards

Many consider the "FAA Airplane Simulator Qualification" form 1993 and "JAA Simulator Requirements for Pilot Training" from 1997 as the fidelity standards from flight simulator qualification [35] [65]. However in both these documents, the term fidelity is used only sporadically and is even not formally defined. The FAA/JAA standards basically comprise three elements. The first element is the flight simulator qualification level, which specify the minimum flight simulator requirements for four classes (A, B, C and D) of pilot training purposes. This 'level' qualification is often used as a synonym to quantify the flight simulator fidelity. The second element is a listing, for each qualification level, of a set of required simulator component characteristics and tolerance levels for certain simulated aircraft dynamic characteristics with respect to the real aircraft behavior. The last element comprises a standard for how to obtain real world reference data plus an evaluation process standard outlining how the simulator representation compares to this real world data and meet the specified tolerance levels (proof-of –match). Three evaluation types are used:

1. *Objective evaluation or validation testing*: A quantitative assessment based on comparison with actual aircraft data, preferably flight test data.
2. *Functional testing*: A quantitative assessment or verification of the operation and performance of the flight simulator (controls, instruments, airframe systems etc.) by a suitable qualified evaluator.
3. *Subjective testing*: A qualitative assessment of the aircraft behavioral and representational characteristics of the simulator based on established standards as interpreted by a suitable qualified evaluator.

FAA/JAA requirements are the most well defined methods for assessing simulator fidelity when compared to all other methods discussed in this chapter. However, their direct application is tailored and thus limited to civil aircrew training purposes. The approach is a front-end fidelity analysis, which means the level of fidelity is assessed when the simulator is completed and not during its development. Furthermore, the method is heavily based on the assumption that a rich set of real-world data is always available through flight-testing.

2.3 Distributed Interactive Simulation Area Perspective

Already during the Apollo program NASA used simulator networks for training and mission rehearsal [153]. However, it wasn't until the late 1980's when affordable network capabilities, like the Internet, came available to the general modeling and simulation community that usage of networked or distributed simulation became regular practice. This resulted in one of the first industrial standards for interoperation of simulators, the Distributed Interactive Simulation (DIS) protocol, in the early 1990's. With the rise of DIS many new fidelity concerns and problems emerged adding to the unitary simulator fidelity issues of the past. Simulator fidelity effects now transcended its own boundary to all other simulators in the simulation network and introduced additional complexity in assessing the overall fidelity of the complete simulation system [26]. In this section, the two most significant research publications on the specific fidelity issues of DIS based simulation applications are discussed.

2.3.1 Lane's Perspective on Fidelity in Distributed Interactive Simulation

In his 1992 publication, "Fidelity and Validity in Distributed Interactive Simulation: Questions and Answers", Lane uses dialog approach of seventeen questions and answers to address simulator fidelity, application-based fidelity requirements and validity issues in the context of DIS based simulations [76]. The set of Q&A outline Lane's concept of fidelity and method for making decisions about fidelity requirements in DIS based simulations, called Fidelity Anchoring.

Fidelity Definition, Concepts and Quantification

Lane uses the term fidelity as a shorthand measure for the overall agreement between the simulation and the perception of reality. When it is possible to define this conception of reality precisely, fidelity can be defined as the degree of correspondence between the simulated situation and the reference situation. The reference situation is that situation in which the real system is operating. Based on this, he formally defines fidelity as an engineering concept referring to the physical correspondence of the simulator's hardware and software to that of the actual equipment being simulated. Realism is defined as a separate concept referring to the perceptions and subjective judgments of the users whether the simulated system performs and appears sufficiently close to the real system.

Lane also remarks that implementing 'all you can afford' levels of fidelity in a simulation is not always the most correct and cost-effective approach to address the problem of how much fidelity is required for a specific purpose. To deal with this issue the Fidelity Anchoring method is proposed, which is a process for forcing a systematic justification of simulation fidelity requirements and ensuring that fidelity decisions and resource investments are based on the identified uses of the simulation. On fidelity quantification he states that fidelity must be seen not as indivisible, all-or-none concepts, but rather as a term with many dimensions and subdivisions, far too many to be represented by any single metric. Therefore, fidelity requirements must be examined for specific parts of a simulation and not for the simulation as a whole. Then these fidelity judgments could, according to Lane, be converted into numerical indices, but they only have a meaning when looking at the parts but not for the simulation as a whole.

Fidelity Anchoring: Major Concepts

To be able to answer the question of how much fidelity is required for a simulation, it is necessary to take a close look at what exactly the simulation is intended to accomplish. The fidelity anchoring method is based on matching the characteristic of each component and sub-component of a simulation to a defined purpose or intended use of the simulation. Each fidelity decision must be based on a specific requirement. The premise of this method is that it is possible to specify what the simulation is intended to accomplish and the probable range of applications for which it will be used. Furthermore, each decision about the appearance and operation of the simulation must be justified or *'anchored'* by a systematic examination of requirements. Fidelity anchoring uses three criteria to anchoring the level of fidelity. These criteria are training effectiveness, user acceptance and affordability constraints. The major objective of fidelity anchoring is to ensure that every component of a simulation should have the exact degree of fidelity required by its intended application no more and no less. Therefore, the process uses a detailed examination of simulation requirements on four key dimensions or fidelity drivers. Simulation requirements are then systematically analyzed and cross-compared to derive fidelity requirements. The four fidelity drivers are the following:

1. *Mission Segments.* This dictates the specific tasks to be performed by the operator, the system components, and the simulation components on which fidelity should be focused.
2. *Simulation Objectives.* The fidelity needed to meet the objectives should be based on the extent to which each of the identified activities must be supported.
3. *Fidelity Dimensions.* The dimensions on which fidelity can be examined and evaluated are: characteristics of simulator as a stand-alone entity, the operator or team tasks to be performed and external processes or events generally arising from the dynamics of system participation in the distributed interactive environment.
4. *Simulation Components.* A simulation can consist of one or more simulators, live equipment, etc. interacting as required by the evolving scenario. Simulations can thus be decomposed into several components, which can be divided into local and global sets. Local components are parts of the simulator (motion, visual etc.) itself while the external processes and environments with which the local components interact define the global components (network, communication protocols, etc.). Since fidelity decisions must be made for each individual component it is important to break down the whole simulation into its building blocks.

Fidelity Anchoring: Process Steps

The essential idea behind the fidelity anchoring process is that decisions on both configuring a simulation system and investing the resources for it should be based on a systematic rational examination of how that specific simulation is to be used. To be able to make these decisions the previously described three criteria and four fidelity drivers are combined in the single analytic process of fidelity anchoring. The criteria effectiveness and user acceptance are used in the earlier stages of this process, whereas the third criteria, affordability constraints becomes relevant only when alternative simulator configurations have been identified. The ultimate goal is to determine, for each component, on each fidelity dimension, the degree of fidelity required to support the intended uses of the simulation. Fidelity anchoring consists of four successive stages:

1. Determine which fidelity dimensions is relevant to each simulation component. To make this decision a two-way matrix structure is formed by the intersections of components and dimensions. For each cell of this matrix a relevance judgment needs to be made.
2. Determine for each relevant matrix entry the highest fidelity required to attain the simulation objectives in any mission segment to be simulated on either the criteria training effectiveness and user acceptance.
3. Use for each relevant matrix element combination determined in stage two the highest level of fidelity to design that component.
4. Now use affordability criteria to determine whether all required ‘ideal’ levels of fidelity could be afforded. If not, then reverse the process to see where fidelity requirements can be loosened with least loss.

2.3.2 IEEE-1278.5 Distributed Interactive Simulation Fidelity Standard

The IEEE-1278.5 Standard for Distributed Interactive Simulation: Fidelity Description Requirements, is a standard developed in 1995 on a government and industry initiative as part of other IEEE DIS standard and recommended practice efforts [63]. This standard defines a method to describe the fidelity of DIS-compliant model and simulation components for real-time simulation applications. This method consists of two building blocks: a fidelity definition taxonomy and a fidelity assessment process (*Figure 2-1*).

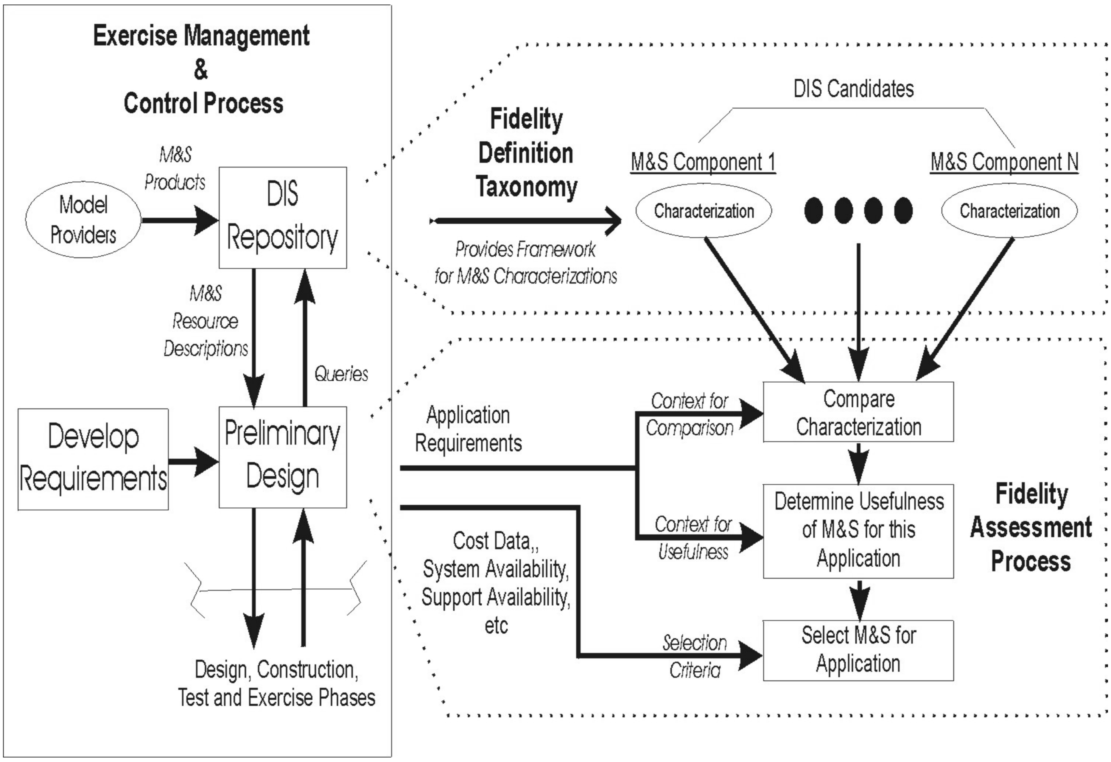


Figure 2-1 DIS Fidelity Assessment Process Model

According to this standard it is the user who must make this decision based upon the specific application. However, it does not prescribe any application specific required minimum levels of fidelity for simulators to participate in DIS applications. Instead the first building block of this DIS fidelity method, the fidelity description taxonomy, is proposed as the tool to support this decision process and to assess which simulator components are appropriate candidates for a given application. Only a very high-level and abstract figure of the second building block, the fidelity assessment process (*Figure 2-1*), is given without detail explanation or other guidelines, is given on how to use the fidelity taxonomy in practice. Even a formal definition of the fidelity concept is missing. The remaining part of this IEEE standard, about five hundred pages, presents the developed fidelity taxonomy.

Fidelity Description Taxonomy

The IEEE Fidelity Taxonomy is primarily an exhaustive enumeration of fidelity definitions or descriptions, which are structured according to an object-oriented decomposition tree approach. The hierarchy is developed solely from a military problem domain perspective and is composed of the following six levels:

1. *DIS Resource Level*. This is the top level and defines a combination of hardware and software components i.e. a simulator that can be connected to and interoperated within a DIS application.
2. *Fidelity Domain Level*. The total fidelity domain level is composed of a set of mutually exclusive objects, which define the physical configuration of the DIS resource and its physical location (host/site), its virtual configuration (action space entity), and its virtual location (environment or world in which the action space entity operates).
3. *Capability Level*. Decomposition of objects, which are the mutually exclusive subsystems of the fidelity domain level objects. These are properties that describe the type of actions or functions that a DIS resource can perform or represent.
4. *Implementation Level*. The means by which capability layer objects are realized i.e. a specific instantiations of a capability object. For instance, a radio system is an instantiation of the communication capability of an aircraft action space entity.
5. *Characteristics Level*. The characteristics level objects are objects, which represent individual qualities or aspects of an implementation. For example, an antenna is a characteristic of a radio system.
6. *Descriptor Level*. The set of the most primitive objects of a DIS resource and are a measurable feature of a characteristic level object, including units and definition of measurement. For example, a radio antenna orientation or gain patterns.

2.4 High Level Architecture Area and Other Contemporary Fidelity Research

Simulation fidelity issues and problems gained renewed interest of the modeling and simulation community with the advent of the High Level Architecture (HLA) as the successor of the DIS standard in the late 1990s. Most of this new fidelity research has been conducted within the Simulation Interoperability Standards Organization (SISO) in the context of the development and validation of HLA based simulations. Therefore, the focus of this section is on the most substantial contemporary fidelity ideas and trends from the SISO community research efforts. But contemporary fidelity perspectives originating from other modeling and simulation communities will also be discussed in this section.

2.4.1 Pace's Perspective on Simulation Fidelity

Pace is one of the most notable authorities on simulation fidelity and validation issues, with significant contributions in both these fields. Pace's perspective on simulation fidelity as presented here is based on three of his fidelity publications [104] [105] [106].

Fidelity Definition and Concepts

Pace remarks that simulation validation processes lack specific guidance about hard validation issues, such as describing, measuring and estimating the fidelity of a simulation. Fidelity in simulations is important because if it is not addressed quantitatively, then design and development decisions based upon simulation results have significant potential for performance risks. Furthermore, Pace states that there still are no commonly used fidelity definitions and concepts. To develop these, fidelity should be addressed from a multi-disciplinary view, combining ideas from experienced simulation developers/engineers, statistical experts, analysis specialist/modelers, theoreticians concerned with fidelity and validation, and simulator users. At least the following fidelity related terms should be defined by the modeling and simulation community: accuracy, error, fidelity, granularity, precision, resolution and validation. The definition for fidelity proposed by Pace is: *"The degree of exactness of a model or simulation representation when compared to the real world"*. According to Pace, simulation fidelity is an absolute concept while simulation validity is a relative concept, dependent upon the simulation application. In other words the fidelity of a simulation is always the same, independent whether the simulation is valid or not for a specific application.

Pace states that compared to unitary simulations the fidelity assessment of distributed simulations is more complicated because implementation limitations imposed by hardware and the impact of the distributed simulation environment need to be considered. In distributed systems a cascading effect of combined errors of simulators can reduce the overall fidelity of the complete simulation. Communication delays may produce causality errors, synchronization errors and uncontrollable variations. The simulation fidelity result is thus a function of the complete set of simulators involved in the simulation and the way they interact.

Fidelity Quantification and Measurement

Pace takes a system engineering approach to fidelity quantification. The approach doesn't focus on developing a singular fidelity metric since Pace believes that developing and using such a singular fidelity metric is neither very meaningful nor possible. Rationale for this is that they lack the information content for proper technical decisions about simulation appropriateness for a particular application. To quantify simulation fidelity, Pace decomposes fidelity into dimensions and attributes.

Dimensions of simulation fidelity are concerned with the portion of pertinent entities, factors, and relationships represented within the simulation. The first dimension involves the enumeration of entities. This dimension is addressed in scope and depth. Scope in this regard means the spectrum of entities represented by the simulation and depth is the (de-)composition level of each entity in smaller elements. The second fidelity dimension, factors, identifies those internal and external processes that influence, impact, or describe entity states and behavior. Finally, the relationship dimension specifies the represented relationships among entities involved in the simulation. This concept of fidelity dimensions requires the existence of an authoritative description of the characteristics of reality, which the simulation represents. The closeness of the measurement between this description and the simulation is the essence of fidelity. Besides the fidelity dimensions the suggested approach addresses the following seven attributes of fidelity that are concerned with the quality of the dimensions:

1. *Order*. The order of the used behavioral descriptions for each factor in the simulation. For instance, a second order algorithm for aircraft dynamics.
2. *Accuracy.* A measure for how good a parameter or set of parameters represent the reality simulated.
3. *Precision*. The level of granularity with which a parameter can be determined.

4. *Timeliness*. The manifestation of timeliness impact on simulation fidelity
5. *Error sources:* The simulation errors that must be considered in simulation fidelity assessment.
6. *Consistency*. The bias and stability of simulation results in terms of the dispersion of results induced by simulation processes.
7. *Repeatability*. The quality of reproducing the same results/responses given the same stimuli (inputs, decisions, operator actions, etc.) by a simulation.

2.4.2 Gross' Perspective on Fidelity Differentials of HLA Simulations

Gross' perspectives on fidelity assessment of HLA based simulations are described in two publications [45] [46]. His ideas are an extension of the initial fidelity differential concept as described in an early DIS area paper and later further refined in two other publications [153] [132] [133].

Fidelity Definition and Concepts

Fidelity is defined by Gross as the extent to which a model reproduces the referent along one or more of its interests. In here a referent is an authoritative description of reality in the context of HLA. To characterize fidelity Gross uses three classifications: Existence (which object of the referent exists in the model), Attributes (which object attributes of the referent exist in the model) and Behavior (what object behavior of the referent is included in the model). The next four theorems for fidelity are proposed by Gross:

1. The range of the fidelity of a model A is between zero and one or mathematically $0 \le F(A) \le 1$. The higher the value the higher the fidelity.
2. If the model A is perfect the model is no longer a model but is the referent R or mathematically, $F(A) = 1$ then $A \equiv R$.
3. The fidelity of a model B that incorporates another model A is limited by what is incorporated or mathematically, $F(A) \ge F(B(A))$.
4. The fidelity of two models A and B working in corporation is limited by the lowest fidelity one or mathematically, $F(A) + F(B) = min\,(F(A), F(B))$.

Fidelity Quantification and Measurement

In his first publication, Gross identifies the following four fidelity measurement methods that should be combined within a fruitful fidelity framework and practice [46]:

1. *Counting/Checking*. Checking and counting if the required attributes are present in the simulation.
2. *Tolerance bands*. Comparing the numerical simulation output and the referent output as follows:

$$F = 1 - \left(\frac{|M_i - R_i|}{A} \right) \tag{2.4}$$

   where F is the fidelity, M_i is Model data point, R_i is Referent data point, A is required accuracy. It only considers the behavior of a single attribute.
3. *Expert opinion*. Expert opinion is useful in considering integrate behavior of entities but has limitations as to subjective and misinformation.

4. *Soft computing*. Using artificial intelligence techniques to measure fidelity, such as expert systems, fuzzy logic, and neural networks.

In a later publication, Gross contradicts this proposed measurement method number two when he states that fidelity should be separated from accuracy [45]. Fidelity is a measure of detail and accuracy a measure of correctness of the represented detail. On fidelity measurement Gross adds here that fidelity is inherently a static analysis of the model i.e. it is not necessary to execute the model. Accuracy is inherently a dynamical analysis and can only be measured by collecting data from executions. Obviously this is very strange since simulations are usually mimicking systems and real-life processes that are dynamic by nature. Therefore, to assess the fidelity of a simulation the consideration of execution results is indispensable.

2.4.3 Foster's Perspective on Model and Simulation Fidelity

Foster's fidelity perspective focuses primarily on a cascaded error estimation technique as a construct for fidelity assessment during the conceptual modeling stage of simulation development [41] [42]. Fidelity is defined by Foster as how well the simulation agrees with the real world measured data. Fidelity is used by Foster as a synonym for accuracy and is inverse proportional to the error of the simulation components and their interactions. According to Foster, fidelity requirements are determined by the simulation objectives. Fidelity requirements are represented by the maximum allowable error bounds in a simulation. This error is a function of the simulation components and their interactions, it includes object models, database support, data exchanged between models and the effect of the simulation infrastructure.

Fidelity Measurement Process of Simulation Abstractions

The first abstraction encountered in the simulation development process represents reality and is represented by Foster as the letter *F*. *F* consists of measurement data gathered from the real world. To ensure a valid simulation the maximum allowable errors should be defined at this stage. A modeler will take into account all-important aspects of the reality to achieve the simulation objectives and develops the conceptual model *G*. The models are developed using the observations of states *F* and identification techniques. *G* should include performance variations due to the implementation of the design in a computer environment. Finally the implementation and simulation execution model or *H* will deviate from *G* due to computational irregularities etc. Foster defines the following errors:

$$\begin{aligned} E_{gf} &= G - F \\ E_{hg} &= H - G \end{aligned} \quad (2.5)$$

where *F, G* and *H* are multidimensional functions and states. Therefore a suitable norm can be chosen as a measure for the error. Furthermore, the model *G* is expanded as follows:

$$G = A + E_{ms} \quad (2.6)$$

where A is the algorithm to describe the dynamic behavior, E_{ms} is defined below. This algorithm has an error, $E_{af} = A - F$, relative to the real world measurements due to numerical integration, linearity approximations etc. Assuming one has perfect data the algorithmic error describes the variations from ground truth of reality one expects as the results of using this algorithm. If desired the accuracy can be assessed from the norm of the error. E_{ms} is the error associated with the sensitivity of A with respect to data errors and is defined as follows:

$$E_{ms} = \left. \frac{\partial\ A}{\partial\ \overline{\mathrm{x}}} \right|_{\bar{x}=\bar{x}_o} \cdot \Delta\, \overline{\mathrm{x}} \qquad (2.7)$$

where $\Delta\bar{x}$ is the vector of data errors due to internal and external model variables/parameters/states errors. Foster visualizes a simulation as a composition of a number of interconnected model layers and associated sub-models. These layers and models exchange data according certain input/output data flow paths. Each model will compound the error of another model as data flows along a path from one model to another. Thus, all data flow paths should be checked, using the previously described formula (*2.5* until *2.7*), to determine if the system design meets the required system fidelity. Both a bottom-up and a top-down approach can be used to assign accuracy requirements to each model. It requires an iterative process to improve the models to achieve the required accuracy when needed. Overall system level fidelity accounts for all component and interactions. Therefore, to achieve the desired level of fidelity for the complete simulation, fidelity must be maintained along each data path among and between models during simulation execution. If the top layer is always outputting data to the user within the error limits, the required fidelity is achieved.

2.4.4 McDonald's Perspective on Fidelity Requirements Definition

The fidelity work of McDonald focuses on an approach for operationally defining the fidelity of candidate simulators available for a distributed simulation exercise and the required overall simulation fidelity level in order to achieve the simulation objectives [83]. The rationale offered by McDonald for having such an approach is the fact that it is too expensive or technically infeasible to develop a simulation that provides a 100% accurate representation of all aspects of the real world. In order to meet affordability and real-time performance goals fidelity concessions have to be made by the simulation developers.

Fidelity Definition and Characterization

McDonald defines fidelity as the accuracy of representation when compared to the real world. This fidelity has two major parts: the extent to which the simulation models each aspect of the real world and the agreement between the performance of each modeled aspect and real world performance. According to McDonald, simulation analysts use an abstraction process to abstract synthetic entities, actions, characteristics, and behaviors. For aspects near the simulation interest this abstraction contains more fine-grained decompositions, extensive detail and higher accuracy, while for those aspects outside the simulation interest the abstractions use far more coarse-grained decompositions, with limited detail and lower accuracy. This is the reason why McDonald states that any attempt to quantify the fidelity of an overall simulation by a single metric is doomed to failure.

Available and Required Fidelity Specification Process

To determine and to document the fidelity of existing simulations in operational terms McDonald suggests a structured and interactive questionnaire approach. This would yield a two-staged quantitative index. The first stage is to determine whether a given aspect of the real world is represent by the simulation or not. How far above zero the fidelity of the simulation resides is the second stage and would be indicated by the accuracy data of each aspect.

McDonald's proposed process for determining the required level of fidelity of simulations for an exercise is based on the following premise: The focus and goals of an exercise is operationally defined by the measures of performance (MOPs), which serve as surrogates for the real world measures of mission success. The components and sub-components of a simulation serve as models of the matching aspects of the real world. The aspects of the real world that have the greatest impact on the MOPs require to be modeled in the simulation with the highest resolution and accuracy. Aspects of the real world that have less impact on the MOPs can be simulated with lower resolution and less accuracy models. The simplest way for the simulation developers to determine this required fidelity of each simulation component is to define the extent to which each of the matching aspects of the real world impact the MOPs in the real world. Interviews with subject matter experts (SMEs) are required in this process to properly rate the impact of each aspect of the real world on the MOP. A very course grained five-point rating scale is proposed for this purpose.

Using McDonalds required fidelity and available fidelity definition approach it is possible to make a proper selection of available simulations based on their available fidelity description that meet the fidelity requirements of a specific exercise. The simulation developer makes a first selection of candidate simulations by searching for matching simulation details and than comparing the accuracy of each matching simulation detail to the required accuracy. The final selection of candidate simulations is performed by SMEs through a comparative analysis between the exercise fidelity requirements and the available fidelity capabilities of each candidate simulation.

2.4.5 Meyer's Perspective on Fidelity Quantification

Meyer's perspective lays out a set of definitions for describing simulation goodness, which try to reflect the notions that the object-oriented paradigm has brought to the attempts to illustrate the interplay of these definitions in several different application domains [87]. He further attempts to outline how the modeling and simulation community can adapt or extend these definitions to bring some quantitative measures for simulation goodness.

Definition and Discussion of Simulation Goodness Terms

Meyer identifies four simulation goodness terms: detail, accuracy, resolution and fidelity. *Detail* describes how the dimensionality of the model with respect to the physical entity, which it supposed to represent, is captured. Formally defined as a measure of the completeness/complexity of the model with respect to the observable characteristics of the physical entity. *Accuracy* of each dimensional element of a model is formally defined as follows: *"The exactness of the model with respect to the observable characteristics and behaviors of the physical entity"*. For example, detail would relate to how complete an aircraft model is with respect to the identifiable characteristics of the real aircraft, such as engines, gear, controls etc. On the other hand, accuracy would relate to how precise any of these represented aircraft features are with respect to its real aircraft counterpart. Thus, for example how well corresponds the aerodynamic model to the real aircraft. *Resolution* is defined by Meyer as a measure of the minimum degree to which the accuracy and detail of the constituent models must coincide with the required level of fidelity of the simulation. In here fidelity is defined as the agreement of a simulation with perceived reality.

Quantification of Detail and Accuracy

According to Meyer it is a difficult and arbitrary process to determine what and how many characteristics there are in an entity. He provides two approaches to solve this problem. Firstly, by formation of so-called entity domain expertise consortia whose role would be to identify and develop those characteristics, which define their classes of systems. Secondly, by using a template approach that functionally decomposes the problem domain into its constituent models. Each of these models is further decomposed into systems and subsystems, until a level of detail is reached which would support virtually any question which might be asked concerning each constituent model. Once this model's detail has been measured, one can begin to measure the accuracy of each element of detail. This can be done in a variety of ways depending on the nature of the detail elements. Meyer does not mention any specific accuracy measurement methods.

2.4.6 Computer Science, Operation Research and Economics Perspectives

Modeling and simulation is a tool often used by computer scientists, operational researchers and economists for various system analysis purposes. This large simulation user domain is often characterized as the discrete event simulation community in contrast to engineering disciplines and physics, which are characterized as the continuous time simulation community [2]. Compared to the continuous time simulation this discrete event simulation community does not really use the term fidelity to describe the conformance of simulated reality to reality. Literature on simulation validation from this domain shows that the term accuracy is used instead [2] [8] [66] [81] [115] [129].

This difference between both the two can easily explained by the nature of the problem and applications of discrete event simulations. Most discrete event simulations are of the type of implementation of some sort of stochastic mathematical model of system, which is simulated on a single computer to study its behavior and performance. These simulations do usually not involve human-machine interaction, do not run in real-time and the output data are a set of numerical values or graphics of the simulated system behavior. In this sense the term accuracy is solely used to quantify the correspondence of numerical outputs compared to numerical output trajectories from the real system. Since the systems of interest of computer scientists, operational researchers and economists are of stochastic or random nature, statistical and probability analysis are the mainstream techniques used [2] [8] [66] [81] [115] [129]. On the other hand, human player involvement does play an important role in the continuous simulation community, especially the training domain. It seems that usage of the term fidelity in this application area finds its origin in using it to specify how well the human player perception of the simulated system corresponds to what is perceived by a human player in the real system. Due to its inevitable subjective character, this is an issue hard to grasp and to describe in terms of numerical accuracy alone.

2.5 Fidelity Practices in Retrospective: Potentials, Limitations and Unresolved Issues

This section provides an overall comparative analysis and evaluation of the existing fidelity theories and practices discussed in the previous sections as well as other work related to simulation fidelity, which are not previously referenced. Primary purpose of this comparative analysis is to identify the similarities, differences, relationships, problems and limitations. The evaluation of the analysis results discusses the meaningfulness, usefulness and reusability of (parts of) existing fidelity approaches within today's simulation development and applications. This knowledge serves as the basis for the simulation fidelity theory and practice developed in the remaining part of this publication.

2.5.1 General Observations

In modeling and simulation literature, it is commonly recognized that fidelity is an intrinsic element of any simulation, one which all simulation developers and users have to deal with one way or the other. It is also commonly agreed that the concept of fidelity is an essential vehicle in properly assessing the validity and credibility of simulation results. If there is no way to determine the credibility or validity of a simulation, decisions based on results of that simulation become questionable [119]. This is of vital importance when applying simulations to safety-critical applications, such as civil aviation applications. Furthermore, fidelity is found to be one of the main cost-drives of any model or simulation development [49] [57] [76] [83] [121]. As a rule of thumb, the higher the fidelity the more time and resource consuming the simulation development is. Thus, being able to state what level of fidelity is required and available avoids unnecessary investments in superfluous simulation components and unusable simulations. Despite these observations the ability to characterize and quantify the level of simulation fidelity still remains a largely uncultivated area when compared to the enormous advancements in simulation hardware and software.
On the other hand the modeling and simulation community starts to realize that the current fidelity practices can no longer fulfill the demanding requirements of today's simulation applications such as simulation-based design of new systems or zero flight time training of airline pilots [47] [144]. This increasing dependency on simulation results, the greater complexity and more frequent reuse of simulations across different applications calls for more formal, robust and comprehensive fidelity methods.

Traditionally, fidelity research has been a practice well cultivated in the training simulation domain, especially in the pilot training area [1] [65]. Most of this research has been, and still is, focused on the human perception of the entity behavior and representation and how to stimulate the human sensory systems in order to give the human operator the impression that he or she is in the real entity [57] [58]. This type of research always considers fidelity solely in the context of unitary simulators. For unitary simulators, which involve humans, these human-machine interface aspects of fidelity are probably the most important ones for judging the simulator suitability for a certain training purpose but are certainly the only ones. With the advent of modern distributed simulation technology, such as the HLA, a whole range of fidelity concerns have been identified that need to be addressed before the full potential of distributed simulations can be utilized for various application purposes [46] [83] [105] [119] [144]. These are fidelity issues that are not really touched upon and cannot be addressed by existing fidelity practices. Solutions to these fidelity aspects are of vital importance to achieve substantial inter-operation of simulators.

Currently, there seems to be no widely accepted methodologies available for the fidelity characterization and measurement of simulations. A closer look at available modeling and simulation literature shows that there do exist many approaches and ideas on the topic of fidelity. However, most of the publications only touch upon fidelity superfluously and in ad-hoc manner from a specific simulation application or problem domain perspective. Only in some publications it is attempted to assess fidelity in a more rigorously and formal manner [58] [76] [104]. Furthermore, almost all simulation fidelity studies known today start their research or assessment from scratch without really reusing the results of previous fidelity efforts. Therefore, the available fidelity knowledge is widely scattered over the whole modeling and simulation community and mostly unrelated to each other. This results in an incomplete overall picture of the fidelity issue and a lack of generic and consistent fidelity assessment theory and practices.

2.5.2 Lack of Common Agreed Fidelity Definition and Related Terminology

Literature reveals a wide variety of definitions for the term fidelity. Almost every paper provides its own definition for several or all of the following fidelity related terms: *accuracy, error, granularity, resolution, precision, tolerance, validity, model, simulation*. Often these terms are used confusingly, as a synonym for fidelity or do not have an unique defined meaning. Additionally, many connotations for the term fidelity are encountered in literature, such as functional fidelity, physical fidelity, attribute fidelity, abstract fidelity and concrete fidelity. This variety of definitions illustrates that there still does not exist a clear practical and common agreed terminology for fidelity but also for other modeling and simulation terms in general.

A careful examination of the existing definitions of fidelity indicates that the simulation community agrees that fidelity measurement is somehow based on the comparison between reality, or some abstraction of reality, and the simulated representation of this reality. However, the different and often seemingly contradicting definitions of what the fidelity concept exactly comprises causes serious confusion among simulation developers and users. This confusion hampers communication between researchers and therefore limits the synthesis of various existing viewpoints into a single common fidelity approach. A problem also recognized and stressed by Pace [106]. Meticulous consideration of modeling and simulation terminology is thus a necessary precondition in understanding the problems and tasks at hand in a fidelity assessment process. Implicit in the used terminology lie the limitations of its associated fidelity assessment methods and practices. Therefore, carefully developing fidelity related terminology is an important task to produce fidelity concepts and practices workable for the whole modeling and simulation community.

2.5.3 Reality Perception, Description and Data Issues

A major, still unsolved, difficulty encountered in the fidelity assessments is the creation or definition of an authoritative real-world description. This is a very important topic, since fidelity is mostly characterized in literature as some measure of the degree to which the simulation represent reality (Section 2.5.2). In this regard it is necessary to have a formal description of reality to quantify fidelity. Up to this date there is no real common answer to what constitutes a real-world description, what it should look like, and how it should be created and used. Absence of such an authoritative description may also have significant effect on the consistency of fidelity ratings for simulations originating from different application domains, hampering the comparison and interoperability of these simulations. Ideally, there should be one such reality description because there is just one world. However, in practice this is an utopia due to the enormous and largely unknown complexity of the reality, differences inherent to the problem scope addressed by each simulation problem/application domain, differences in cultural background of those involved in the modeling and simulation business and resource constraints. Independently of what real-world knowledge is contained within a reality description, the exact structural format of it depends on the way fidelity metrics are defined and measured and vice versa. This is illustrated by the various proposed reality descriptions, which all differ in structural format, development process and real world contents [46] [49] [63] [65] [87].

In general the opinions on real world data contents of reality descriptions are divided two camps. Firstly, there are those who state that the comparison of simulation results with real-world data is conceptually the most robust and only approach for fidelity quantification [42] [66] [104] [106]. This is certainly true when the simulation is already capable of producing data results and a rich real-world data set is available. However, this is not a valid standpoint in case such data is not present. For example, when it is tried to assess fidelity of a simulation during its design stages or when creating a simulation of a system that does not yet exist, which is often the case for engineering applications. Secondly, it is stated that one should focus on a reality description that is more of theoretical nature, in which actual measured real-world data doesn't play the most important role. Instead, a first abstraction of the real world serves as the reality description by capturing the basic information about important entities involved in any mission space and their key actions and interactions. Examples, of these are the Conceptual Models of Mission Space (CMMS) and IEEE DIS fidelity taxonomy [45] [63]. It is generally acknowledged by the other camp that such descriptions of reality do have some kind of fidelity level of their own due to its theoretical nature and therefore fidelity measurements based on these descriptions do provide erroneous and biased values [42]. There are even those that try to provide methods to describe this fidelity of such theoretical reality descriptions [49].

2.5.4 Fidelity and the Fitness for Purpose Confusion

Since the level of fidelity is a major simulation cost-driver (see Section 2.5.1), the question of 'how much realism or fidelity is good enough for our needs' has been historically the driving force of most fidelity research. This question is the prime reason why some parts of the modeling and simulation community have been wrongly using the term fidelity as a synonym for fitness for purpose. Since simulations are abstract representations of some part of reality, the answer to this question states an important set of requirements for simulation systems, which should guide the simulation development. This has resulted in many fidelity definitions and concepts, which incorporate application specific objectives, tasks and other aspects [12] [46] [58] [63] [93]. Clearly, such fidelity constructs are not workable for the whole simulation community and tend to assign different levels of realism for the same simulation when used in another application. Obviously, the simulation doesn't change in such case, nor its level of realism. It is therefore better to separate a formal fidelity definition from application specific aspects. In this way fidelity becomes an absolute measure of simulation realism instead of the often-used relative fitness-for-purpose measure, which is causing so many pointless fidelity discussion and confusion

Of course, a robust fidelity assessment framework should include methods for selecting the most appropriate existing simulation (trade-of-decision process) during design and methods to determine whether the actual achieved simulation fidelity is sufficient for a specific application (validation process). However, these methods should be based on a comparison of an absolute-fidelity measure in terms of the required fidelity and available fidelity for a simulation. The quantification of the required fidelity from the simulation objectives is mostly based on informal and subjective methods, which assign some sort of importance rating to different fidelity aspects of the simulation [1] [45] [58] [76] [83] [93] [147]. Such assignments should be carried out very carefully, using well-determined rating scales to obtain consistent, unbiased and representative ratings [149]. In this regard the modeling simulation literature on fidelity requirements specification methods demonstrates that not enough effort is put in (re)using or learning from well established requirements engineering processes from other domains such as in system or software engineering. Again, this characterizes today's ad-hoc and informal fidelity assessment methods.

2.5.5 Comparison of Fidelity Quantification Strategies and Approaches

Various types of metrics for quantifying simulation fidelity are found in literature. Existing fidelity metrics can be classified in qualitative versus quantitative metrics, singular and dimensionless versus complex multi-dimensional metrics, and unitary versus distributed simulation metrics. Each of these has its pros and cons, which will be discussed next.

Qualitative versus Quantitative Metrics

Basically, all currently used fidelity metrics have a rather qualitative and subjective character. Analysis of the available literature shows that there is a community wide recognition that subjective adjectives (low, medium, high), which express fidelity in qualitative terms can no longer fulfill the current simulation requirements. Complexity, interoperation and reuse of today's simulations require new more comprehensive methods to assess fidelity both in the development process and in the validation process of simulations, in order to obtain a simulation that produces the desired and reliable (realistic) results against acceptable costs in both time and money. Many publications considering simulation fidelity have therefore stressed the importance of the development of quantitative fidelity metrics, concepts and methods to measure the fidelity of simulations in a more objective manner [22] [26] [58] [76] [105].

Lane identifies the most important problems with describing fidelity requirements in terms of high, low and medium. First of all, they are too subjective and therefore different persons come-up with different interpretations [76]. Secondly, he states that this approach is no different from the educated guesses that must be frequently made in simulation design. Pace goes even further by stating that these subjective fidelity adjectives only serve the purpose of an advertising blurb of simulation developers, with no real meaningful value [106]. Since methods for a more technical engineering approach are simply not available yet, Lane suggests to extend this judgment by adding a more precise statement what is meant by the terms low, high and medium or to use several different and more refined rating scales.

Singular versus Complex Metrics

One of the most frequently encountered methods to quantify fidelity is the attempt of combining al kinds of simulation fidelity factors, which characterize the realism of a simulation, into a single dimensionless metric or total simulation fidelity [1] [12] [46] [65] [93] [133]. This is often accomplished by using specific simulation-task dependent weighting factors for each simulation fidelity factor. These fidelity factors itself are dimensionless. Mostly, however, it not clearly demonstrated how to determine these weights in practice. Some even takes the cost aspects into account and stresses the importance of human-related fidelity [93]. This human-related fidelity is an important and difficult issue, which needs special attention. According to Knepell, a human element in a simulation increases the uncertainty of the simulation dramatically [66]. Cost aspects on the other hand are import in the development process of a simulation but should not be part of a fidelity metric, because they do not provide any information regarding the real world representation provided by the simulation.

The major drawback of singular overall or summarizing fidelity indices is most clearly expressed by Haddix [49]. He states that the use of an overall index is not a substitute for a careful review of a simulation in the context of its development and application since it hides the source of deviations from reality necessary to support proper technical decisions. In other words, a number of relatively insignificant deviations or one substantial element may cause the same overall deviation. However, a singular overall fidelity metric may have some utility in simulation development when a simulation must be selected from equivalent simulations designed for the same purpose and application domain in which they will be used. It provides a tool for a quick first selection. But for making final selection additional fidelity knowledge is required.

Examples of such summarizing metrics are the FAA and JAA flight simulator classification levels A, B, C and D [35] [65]. When a flight simulator needs to be selected for a given type of pilot training, these levels should be sufficient to make a first selection. On the other hand, when the same simulator is used for a purpose other than pilot training these level ratings may not be sufficient and additional, more detailed, knowledge about the fidelity of the simulator is then required. A B747 training simulator (level D) for instance, may not provide the required fidelity to act as a vehicle for aircraft handling qualities research.

The use of complex metrics is the other often seen approach for quantifying simulation fidelity. Foster for instance has been using a three level simulation abstraction and a six-layer federation data flow model, he identified various error sources in simulations and developed an accuracy estimation method [40] [42]. With this method it is possible to estimate how deviations from reality cascade through different coupled models during simulation design stage and how they finally effect the fidelity of the resulting simulation. Pace describes a different but similar method that accounts for different sources of uncertainties that affect the accuracy or fidelity of a simulation [105]. Furthermore, he observes that when errors are not independent it is not possible to

estimate the overall simulation fidelity simply from the accuracy of the simulation elements alone. In that case the effects of interactions between the simulation elements also need to be considered. Fidelity is specified and measured by Pace in terms of various dimensions and attributes (Section 2.4.1). The proposed fidelity quantification of McDonald is based on a two-staged quantitative fidelity index [83]. First it is determined which aspects of the real world are present, then the accuracy data of the present aspects are addressed. Meyer also follows the approach of decomposition of models in an object-oriented fashion [87]. He decomposes a simulation in details and uses the term accuracy to describe how precise a detail represents its real-world counterpart.
All discussed methods illustrate that describing the simulation realism is a complex issue involving many simulation characteristics that are hard, if not impossible, to capture in a meaningful single dimensionless fidelity metric. The advantage of these comprehensive fidelity descriptions is that they have the potential of turning simulation fidelity into an useful development and validation tool, a feature desired by many in the modeling and simulation community. A major drawback of these complex fidelity descriptions is the large amount of data that must be processed, making it more difficult to use, and development more time consuming and expensive. The development of tools to automate such fidelity assessment methods may fully or partially overcome these objections.

Unitary versus Distributed Simulation Metrics
Some believe that the FAA and JAA simulator requirements for pilot training fully addresses the whole spectrum of simulation fidelity issues [35] [65]. Indeed, a literature study reveals that this standard comprises the most mature, practical and proven approach to fidelity requirements and quantification developed yet (Section 2.2.3). However, these requirements are specifically tailored for unitary pilot training simulators. Due to the distributed character (real world data and representational correlation issues etc.) and differences in underlying simulation architecture (network delays, reliability etc.), unitary simulation fidelity metrics are not directly applicable to distributed simulations. These require unique fidelity quantification issues that never played a role in the fidelity literature on unitary simulations [1] [58] [65].

The available publications on fidelity assessment of distributed simulation do provide an enumeration of simulation properties and physical hardware components, which need to be considered in quantifying the fidelity of simulations [63] [76] [153]. Despite the incompleteness, the DIS orientation and high-level descriptive character of this enumeration, it clearly demonstrates that there is a difference between fidelity assessment of stand-alone or unitary simulators and interacting simulators through a network. More importantly, it can be concluded from this work that distributed simulation fidelity assessment comprises all those quantification properties and activities of unitary simulation fidelity assessment and added to this are the fidelity quantification issues related to achieving proper interoperation of geographically distributed simulations through dedicated computer network technology.

2.5.6 Limited Scopes and Different Contexts for Considering Fidelity

Existing simulation practices demonstrate that fidelity characterization and measurement are highly interwoven with all facets of simulation development and validation [22] [103] [144] [147] [153]. It can be concluded from this observation that a mutual relationship exists between how one perceives and abstracts reality and how to animate real-world behavior over time, and how one should measure the difference between the simulation and reality. Therefore, a sensible fidelity theory can only be fully understood and utilized in the context of such a wider modeling and simulation framework.

However, current modeling and simulation practice is still an art-form and at best an inexact science lacking serious fundamental and common accepted theory. The way in which models and simulations are currently build depend upon the real-world system to be represented, its purpose and the background of the simulation developers performing this development process. Therefore, almost every organization uses its own unique modeling and simulation approach. The lack of an uniform and formal modeling and simulation theoretical framework combined with the notion that a fidelity theory should be considered in such a larger context, makes it difficult to develop a fidelity theory that can easily be applied and provide significant improvements to each individual simulation process. This observation is a plausible explanation for the fact that the current fidelity practices are solely considered from a limited scope and specific application contexts. The limited scope is meant in the sense that fidelity methods are developed for a dedicated model or simulation type, or as a separate method for each specific simulation development stage problem such as simulation requirements, discrete front-end analysis activity of certain simulation generated variables etc. As a result of all this, there are practically no unifying and reusable simulation fidelity theories and practices available for the modeling and simulation community. With the advent and extensive usage of new simulation technologies, such as HLA-based distributed simulations, this nowadays poses more and harder problems in addressing fidelity issues during simulation development and validation than in the past. Especially, if one considers the multi-disciplinary simulation development teams with various educational backgrounds and experiences, which separately develop simulators or simulator components that have to interoperate with those of others within a single distributed simulation environment.

2.6 Summary

This chapter has discussed a selection of the many existing fidelity theories and practices. In the previous sections many problems, limitations and unresolved issues of these existing fidelity theories and practices have been outlined that need to be addressed in the development of a formal and unifying simulation fidelity theory and practice. However, for reasons given in the previous section such a fidelity theory and practice should be integrated in and build on a set of principle modeling and simulation engineering concepts that identify how reality is perceived and how this observed real-word knowledge is expressed by humans. These principle concepts should also describe how reality can be or is artificially abstracted by a model or simulation. If fidelity is considered in such a general modeling and simulation framework, a fidelity theory and practice, which measures the difference between ‘true’ reality and ‘virtual’ reality, is then a logical outcome and will fit naturally in any simulation development and validation process applying the same underlying principles. Therefore the next chapter will outline the general modeling and simulation framework, which serves as the foundation of the unifying fidelity theory and practice developed in the subsequent chapters of this thesis.

3 Modeling and Simulation Contextual Framework for Fidelity

3.1 Introduction

Several authors characterize current modeling and simulation practices as more of a black art than a science, lacking common agreed definitions, concepts and methods [2] [47] [58] [81] [105] [156]. This is the prime reason for the in section 2.5.2 discussed issue of the inconsistent definition and usage of fidelity and related terms within the modeling and simulation community. Therefore, this chapter introduces the fundamental modeling and simulation terminology, concepts and processes that are utilized to develop and embed the fidelity theory and practices presented in the remainder of this thesis. In other words it outlines the contextual modeling and simulation framework in which fidelity should be considered and applied. As discussed in section 2.5.6, this framework is necessary in order to develop a useful fidelity theory and practice that can seamlessly be integrated with other modeling and simulation activities. It comprises a synthesis of modeling and simulation theory and practices found in literature complemented with results from own fidelity research of which has already partially been published in [47] [122] [125].

This chapter is organized in three sections. Section 3.2 describes the basic modeling and simulation terminology and concepts. Next the modeling and simulation enterprise is presented by means of a discussion of the general development process for models and simulations and a verification, validation and accreditation process. Both processes serve as the application context in which fidelity theory is utilized to develop valid models and simulations. These two sections are the foundation for the fidelity management process model developed in Chapter 9. The remaining section 3.4 focuses on an abstract mathematical systems theory description for various knowledge specifications encountered in model and simulation enterprise. This theory provides the fundamental conceptual and mathematical formalisms for the formal definition of fidelity, its associated concepts and measurement methods in the subsequent chapters.

3.2 Basic Modeling and Simulation Terminology

The words *model* and *simulation* are often used interchangeably within literature along with their verbs. Despite their close relationship they do have a different meaning. Furthermore, there are several other important terms and concepts related to models and simulations. For the proper understanding and development of a fidelity methodology, unambiguous definitions of elementary modeling and simulation terminology and concepts is a necessary precondition. In the following paragraphs the terminology as used in this dissertation will be defined.

3.2.1 Abstraction, Reality, Real-World and Simuland

The term *model* has many connotations and derivatives such as aerodynamic model, object model, discrete model etc. [47] [55] [56]. However, comparison of all definitions yields a common underlying concept: an abstracted representation or specification of a system, entity, phenomenon or process. *Abstraction* in this context is the process in which a relative sparse set of entities, relationships and their inherent qualities are extracted or separated from a complex reality. Within the modeling and simulation enterprise a model in general is considered to be a placeholder for some part of our universe and is the product of abstraction.

Since models are an abstract representation of reality, it is necessary to define what is understood to be reality or its often-used synonym real-world. *Reality* is commonly defined as the quality or state of being real [85]. Although this definition for reality seems straightforward it is not directly suitable for usage within in a modeling and simulation context. This problem becomes eminent when one considers the SISO Fidelity ISG definition for *real-world*: 'The set of real or hypothetical causes and effects that simulation technology attempts to replicate' [47]. One of the major application areas of models and simulations is research, development and engineering (RD&E). Here systems and situations are studied using models, which are fully or partially non-existent in an observable material form. Most of these models are merely an imaginary or non-observable representation of future systems and situations. Examples of these form the new avionics system concepts and aircraft operational procedures to be used within the civil airspace to reduce air-traffic delays. Strictly speaking, in such a case one cannot speak of modeling reality in terms of being materially real or existent like in the case of a Boeing 747-400 pilot training simulation. To address this modeling and simulation issue in an uniform way the terms *material* and *imaginary reality* are introduced. These terms originate from the SISO Fidelity ISG research activities but are further refined here in order to better fit in the contextual modeling and simulation framework for fidelity as outlined in this thesis [47]. *Material reality* is defined as the material universe or parts of it that have been or can currently be observed. *Imaginary reality* is defined as a possible imaginary universe or parts of it that have no exact counterpart in the material universe. Using these two concepts, *reality* in the context of modeling and simulation is now defined as being either material reality or imaginary reality or the union of both (*Figure 3-1*).

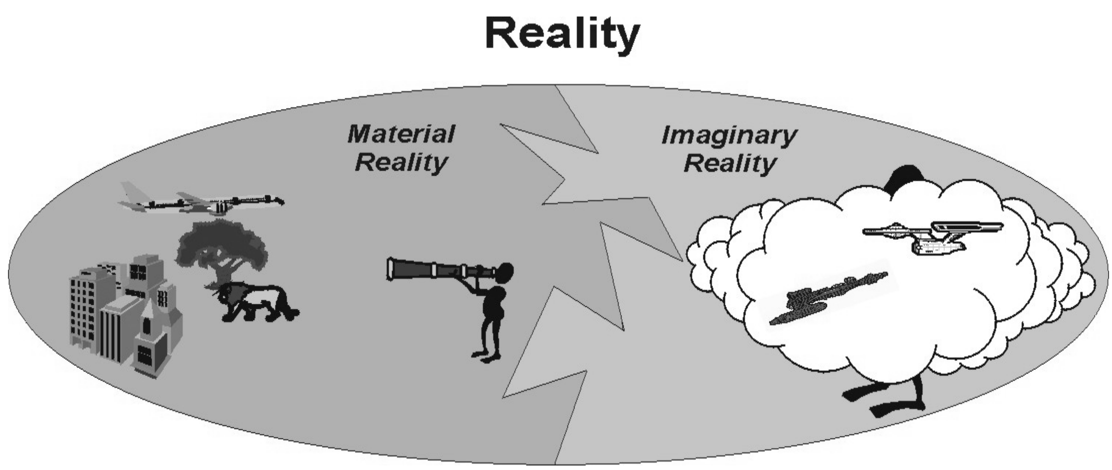


Figure 3-1 Division of Reality in Material and Imaginary Parts

One of the elementary characteristics of abstraction is the omission of those parts of reality that are not of interest for the problem or application at hand. This means that when building models and simulations it is never attempted by its developers to address reality in its entirety. Instead, developers confine their scope to only a limited part of reality. For instance a model of a communication satellite will obviously omit, say, any sea-life representations. To designate this limited part of reality the previously mentioned term *real-world* is often used. A synonym for real-world that has been developed within SISO Fidelity ISG is the term *simuland*. Simuland is defined as that part of reality being simulated (*Figure 3-2*). A simuland can be virtually everything ranging from a system, entity and phenomenon to a process or concept.

3.2.2 Modeling and Models

The current SISO Fidelity ISG glossary contains about sixty different identified terms referring to models in one form or another. Therefore it is hard to provide a common definition for the term model. However, Harmon has classified these terms in three model dimensions and four model categories, which provides a useful basis for creating a common definition of the term model and handling its connotations [56]. The first of three dimensions is the *resolution approach* that captures the difference between black box and glass or white box models. The *representation domain* is the second model dimension, which differentiates between continuous time, discrete time and discrete event models and between stochastic and deterministic models. The last model dimension is the *temporal representation* specifying the difference between dynamic models that describe time dependent behavior and static models that do not exhibit time dependencies. All three model dimensions are orthogonal to the four model categories and apply to any of the categories.
The first model category is *development approach*, describing development approaches such as the waterfall model often used in software engineering. The second model category is the *representation topic* model, which specifies what the model represents such as an aircraft model, tank model etc. The *purpose* of a model is the third category of models, for instance an engineering model. The last and most important category is the *manifestation* category or the medium through which the model is expressed. Harmon identifies three categories of modeling media: *physical*, *computer* and *symbolic* [56]. Physical models represent the portrayed system with some other physical object resembling that system in the desired way, such as wind-tunnel models, aircraft structural test models or a simulator flight-deck mock-up representing the appearance of the real aircraft flight-deck. A computer model renders the modeled reality within a computer through the states of its logic gates and electromagnetic domains, such as a compiled executable, which numerically solves the aircraft aerodynamics and equations of motion. Symbolic models express the modeled reality in symbols. The symbol types include diagrams, words, tables, mathematical relations and formulations.

Combining the discussed interpretation of reality and the classification of models, the common definition for *model* as will be used in here is defined as: Any physical, symbolic or computer representation or combinations thereof that abstracts some parts of reality (*Figure 3-2*). In this regard the term *modeling* refers to the activity of creating a model. The explicit usage of ‘combinations thereof’ refers to another important and general applicable characteristic element of abstraction; the composition of a larger model from a set of smaller models coupled through their input and output interfaces. A clear example of this is the already mentioned flight-simulator that can be composed of a physical model, the flight-deck mock-up, and a computer model, the compiled executable of the aircraft aerodynamics and equations of motion, which can be used to drive the displays and instruments in the flight-deck mock-up.

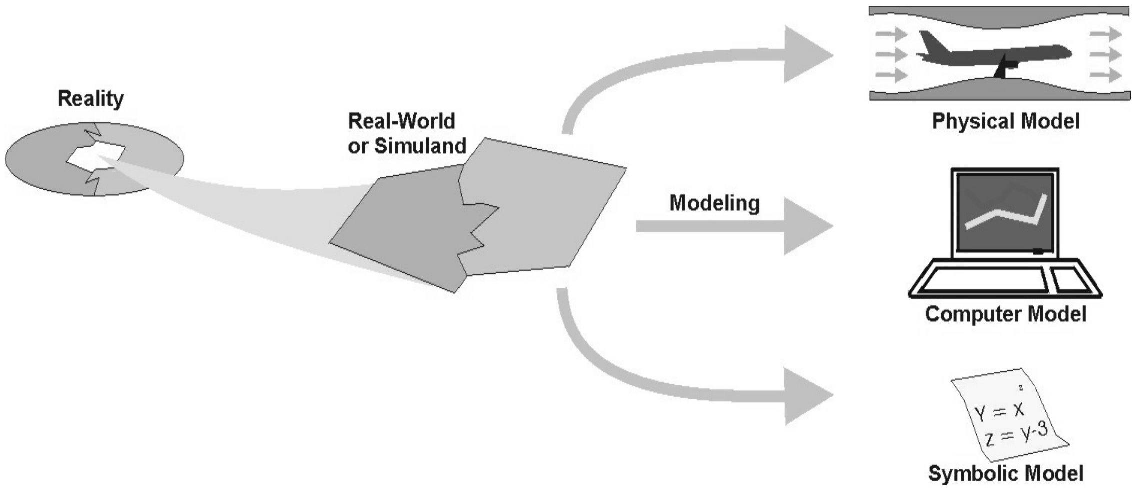


Figure 3-2 Relationships between Reality, Real-World or Simuland and Model

3.2.3 Simulation and Simulation Systems

Like the term model the term simulation also has many connotations and derivatives. Such derivatives are often closely related to the model categories and dimensions previously discussed, for instance a discrete model usually maps to a discrete simulation. Comparison of various definitions for simulation yields two common interpretations [47] [55] [125]. First, a simulation is interpreted as a method, software framework or system to implement and evaluate a model or various types of models over-time i.e. a system in which models are animated. Similar terms like *simulator*, *simulation model, simulation application* and *simulation environment* are also used in this regard. The second interpretation focuses on the act of producing some set of desired outcomes by exercising a model. Especially in the operations-research area and also in the control engineering area such interpretation of simulation is equated to numerically or computationally solving as a set of complex mathematical equations as a function of time for which no analytical solution [81]. Other similar terms used to indicate this interpretation include *simulation process*, *simulation exercise, simulation execution* and the verb *to simulate*.

The common denominator found in all definitions is that simulations, unlike models, are always dynamic processes that represent parts of reality whose state and characteristics change overtime. The notion of time is thus a fundamental characteristic of simulation. As Meyer mentions, models tend to be static in the context that they just exist and do or produce nothing when they are not animated by an engine [87]. On the other hand Harmon states that since a simulation is also an abstraction of reality a simulation can be considered as a possible class of dynamic models [56]. A physical model like a flight-deck mock-up placed on a motion-base with its aircraft dynamics generated by a computer model connected to this motion-base is a simulation since it is capable of providing dynamic behavior as its output. However, such a system is also a dynamic model since it is an abstraction of the motion cues experienced by the human-pilot in the real aircraft. Therefore both of them are right when viewing a simulation in the context of its first interpretation. Often models that are used and combined for creating a simulation are of symbolic type that require a transformation to an integrated hard and software system implementation before they are capable to reproduce some real dynamic behavior. These sets of models are often called conceptual models and are discussed in more detail in section *3.3*.

Using the definition of model and the background knowledge regarding the term simulation, the general term *simulation* is defined here as: The execution of a physical, symbolic or computer model or combinations thereof that results in a reproduction of how some part of the simuland evolves over time. To *simulate*, is the act of executing models. Similar to model categories simulations can also be categorized according to the realizing media: physical simulation, computer simulation and symbolic simulation. An example of a pure *physical simulation* is for instance a hydrologic simulation of coastlines by geographic models made in sand and water. *Symbolic simulation* is for instance a simulation by manually solving a set of mathematical functions over time, which often comes down to the construction of an analytical solution. A *computer simulation* is the actual execution of a computer model. The combination of physical, symbolic or computer models capable of animating how the real-world evolves over time will be designated here by using the term *simulation model*.

When developing and using simulation models for e.g. simulation-based training not only a simulation model is created but also a whole infrastructure of supporting systems is developed to configure, operate and execute the simulation model. Such systems provide a series of additional operational and functional capabilities necessary for the proper usage of the simulation system. Some of these have direct effect on the represented reality during simulation such as scenario preparation and parameter-configuration systems and run-time simulation management systems. Others do not or indirectly effect the reproduced reality such as learning systems, observer systems, safety systems etc. This combination of simulation model(s) and support systems is what is called here a *simulation system* or *simulation environment*. A simulation system is thus the tool capable to provide simulations as requested by its users.

In the rest of this dissertation the usage of the term simulation system is limited to either systems completely build upon computer models or a mixture of computer models and physical models (mock-ups, hardware-in-the-loop, etc.). Another term often used is *simulator*. However, most of the time this term is used for a specific simulation system instantiation, a single stand-alone simulation system that usually involves human-player interaction, like for instance a full flight-simulator.

An important characteristic of simulation systems is that within their own system boundaries simulation systems describe another system. In systems theory terms such systems are called a meta-systems [21]. Models in this regard can be considered as a system specification of the simuland contained within a simulation system. Therefore, like for any system, the (de)composition characteristic is present in simulation systems. This means that a simulation system can be an aggregation of smaller simulation systems that are interacting through their input and output interfaces. For instance a coupled set of flight simulators forming a lager distributed simulation system for the simulation of complex air-traffic dynamics.

The main output of a simulation system is a subset of the *endogenous variables* of the simulation model. This type of variables is either the data of interest or the data to be used by a connected simulation system. The input interface of a simulation system comprises two major types of *exogenous inputs*: simulation model configuration input and independent simulation model input. Often simulation systems have to provide not just a single simulation instance but a range of similar simulation instances to properly fit the user's purpose. In such cases the user is offered the possibility to configure a part of the simulation model before starting a simulation. An important observation to make is that the reproduced real-world is thus only fully determined after these configuration input settings are made. This implies for example that the fidelity of the simulation model and simulation is not uniquely known before a configuration setting is made as will be discussed later on in this thesis. Configuration settings include but are not limited to parameter-settings, scenario, entity instantiation, simulation model component selection, initial states, fixed or predefined input for (random) variables such as recorded data. Independent inputs are those inputs that are fed by endogenous simulation model variables of other connected simulation systems.
An implication of the input-output interface and (de)compostion rule is that the following condition must hold for a simulation system: a simulation system is only capable of providing simulation when, considered at its highest aggregation level, the simulation system forms an autonomous system with respect to its simulation model(s) input-output. Thus its simulation model configuration input must be set and there are no independent simulation model inputs allowed at that level (See also paragraph 3.4.4). Again this for example implies that the degree of realism or fidelity of the resulting simulation of such a composite simulation system can only fully be known when all the composing simulation systems are known and their interactions (exogenous/endogenous variables coupling). Other simulation system inputs and output exist but these relate to operational and functional aspects of supporting systems not effecting the real-world representation of the contained simulation model. *Figure 3-3* at the next page gives the most general representation of a simulation system interface.

In practice, depending on the purpose of the simulation system, not all types of interfaces have to be available to the user. Three possible classes of simulation systems can be defined: *structural autonomous simulation systems, partial structural non-autonomous simulation systems* and *non-autonomous simulation systems*. *A structural autonomous simulation system* is a system that provides no interface capabilities to the user for the simulation model configuration nor for independent exogenous simulation model input. Simulation systems of this kind only provide one fixed form of simulation for a single purpose. These types of simulations are often found in entertainment business simulations like theme parks providing simulator-based rides. The second class of simulation systems is the *partial structural non-autonomous simulation systems*, which allows the user a certain degree of freedom to configure the simulation model to achieve an autonomous simulation system. Examples of this kind include training and test & evaluation simulation systems that train or evaluate human and system performance for a certain set of configurable conditions. The last class is the *structural non-autonomous simulation system* in which the user has the freedom to interconnect and configure several simulation systems from a library of reusable and interoperable simulation system elements to form a larger autonomous *federation* of simulation systems. Examples of such simulation systems involve various kinds of RD&E component based and distributed simulation approaches which offer the user great flexibility to compose a variety of simulation systems to address a range of similar related problem [11].

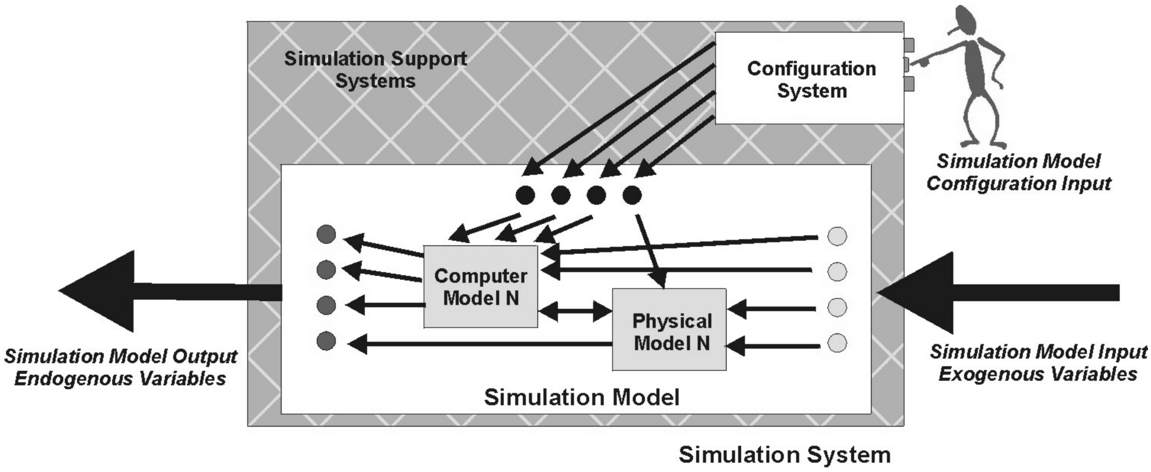


Figure 3-3 Simulation System and Model Relationships

Considering fidelity aspects of simulation systems with regard to its underlying simulation model structure is an essential and necessary element of fidelity assessment. However it is the simulation created by the simulation system that produces the results and answers needed to accomplish user objectives. Thus the realism of the simulation created by the simulation system is of interest to its user, not how it is achieved. This observation is the foundation for the subtitle of this dissertation and the reason for differentiating between model, simulation model and simulation fidelity as discussed later on.

In conclusion of this section three other frequently used simulation terms are discussed. These terms are *unitary simulation*, *distributed simulation* and *parallel simulation*. Similar to the classification of models in three dimensions as suggested by Harmon, these terms can be considered as a simulation dimension category. This category classifies the architectural concept underlying a simulation model. An *unitary simulation* refers to a simulation model architecture, which has the character of a single stand-alone operating unit. Usually, this implies that the computer model is implemented on a single-processor computer device. However, the computer model can also be implemented on a multi-processor computer device then referred as a *parallel simulation*. *Distributed simulation* refers to simulation model architectures that are composed around a set of independent geographically distributed computer and physical models that communicate or inter-operate with each other through a computer network.

3.3 Modeling and Simulation Enterprise

Nowadays modeling and simulations are used for many different purposes ranging from training, simulation based acquisition, test and evaluation to research, development and engineering. More and more simulations play an increasingly important and indispensable role in our society. Due to the technological advancements in computer science and simulation hardware these simulations become more complex and larger. Modeling and simulation is a maturing enterprise whose activities are centered on two closely interrelated engineering processes: development (section 3.3.1) and validation (section 3.3.2).

3.3.1 Model and Simulation Development Process

The goal of a model and simulation development process is to develop a simulation system capable of providing those simulation execution(s) that suite the user needs. To accomplish this, a set of generic development process stages are used, which can be identified in most commonly applied model and simulation development paradigms [18] [22] [81] [122]. Several ways exist to how these stages are traversed ranging from a single pass or waterfall model to more complex ones such as spiral development paradigm.

It is necessary to make a distinction here between simulation developer and user roles in simulation development, although in some cases these are the same persons. A developer develops a simulation system as a tool for simulations. It is the user that requests the tool development and then uses this tool to create the desired simulations.

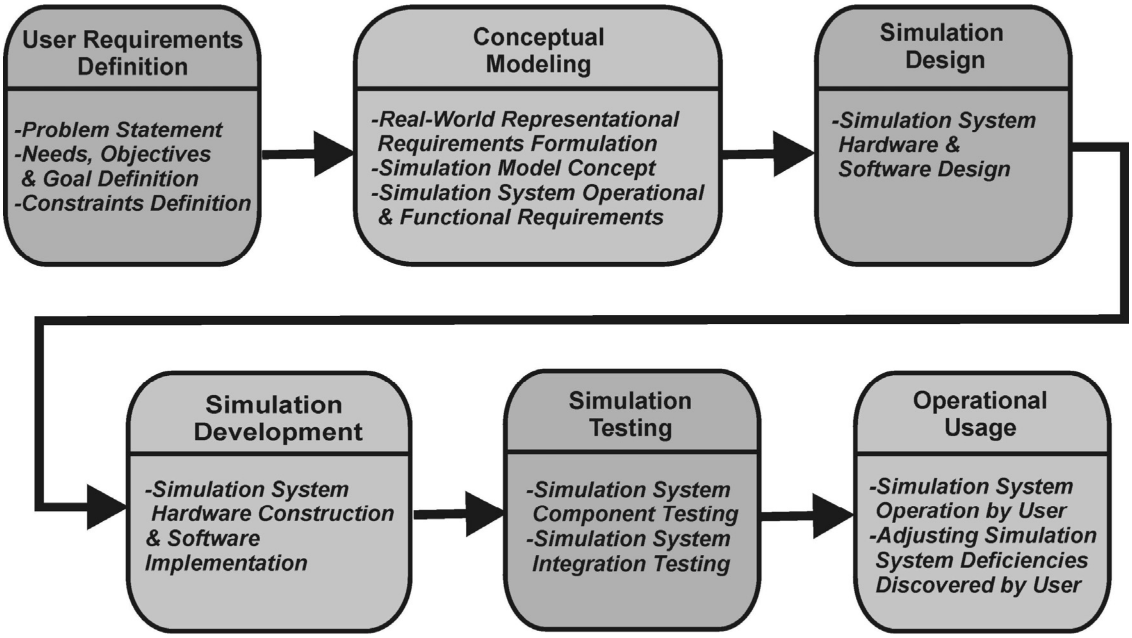


Figure 3-4 Generic Model & Simulation Development Process

User Requirements Definition

According to standard system and requirements engineering approaches, user requirements can be classified in functional and non-functional requirements [9] [43] [74] [82] [149]. For simulation system development functional user requirements development comprises the specification of the desired functions the system must offer to fit the user needs. In other words it specifies what the user expects from the simulation system. Such a specification must explicitly address the exact needs of the user for this simulation system and a clear problem statement to be addressed with the simulation(s) provided by the simulation system. For instance in case of a RD&E simulation system it could describe the objectives of the research and the goals of the simulation experimentation plan. Furthermore, it should also specify the environment in which the simulation system will be used. Often simulation systems are part of a larger environment in which they have to fulfill a specific role. For example in pilot training systems in which a full flight-simulator is a tools for training pilots in handling emergency situations that cannot be trained in real aircraft. Non-functional requirements describe the external constraints the simulation system and its development must meet.

The non-functional requirements limit the solution space in which the functional user requirements of the simulation system have to be attained. Since the major user function of a simulation system is to provide simulation of some part of reality this also yields that the degree of realism or fidelity of this represented reality is bounded by these non-functional requirements [121]. In practice this could mean that certain functional requirements could conflict with the non-functional requirements. Non-functional requirements include constraints like reuse of certain existing hard and software, usage of a certain standard, budget, time-schedule, personnel available and other resource constraints.

Conceptual Model Development

Like fidelity the term conceptual model is a confusing term with many connotations and interpretations. Research efforts show that there is no consensus of what is commonly understood to be a conceptual model [76]. However, conceptual modeling is an essential step in simulation system development not found in most other system development processes. Unlike other systems a simulation system is a representation of another system: the simuland (Section 3.2). This requires that for simulation systems not only a design has to be made for the simulation system itself but also a 'design' of how the simuland will be reproduced within this simulation system. Therefore a *conceptual model* is interpreted here as the formal specification of that part of reality to be simulated to meet the user needs. The conceptual model is a symbolic model or blue-print of how the simuland will be realized in the actual simulation model. In practice conceptual modeling bridges the gap between user requirements and simulation system requirements that guide the simulation system design. As discussed in an earlier publication user requirements and conceptual modeling stages are intrinsically coupled processes that derive from each other [121]. The result of conceptual modeling stage is twofold. First it specifies what part of reality should be represented and its quality to meet the user simulation purposes i.e. fidelity requirements (Section 8.2). Furthermore, this should also include the degree of freedom offered to the user to configure certain parts of this represented reality. Secondly a conceptual model specifies the simulation model requirements or concept by describing how the real-world requirements will be satisfied by the simulation model, which includes: assignment of what parts of the real-world will be represented by computer or physical models and their relationships, composition, assumptions, physical relationships and formulas, algorithms, data, time management approaches etc.

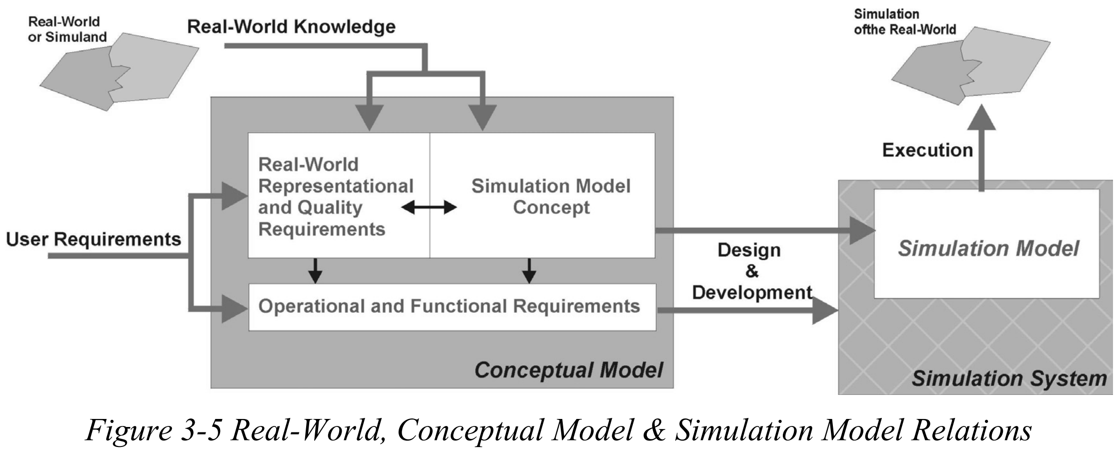


Figure 3-5 Real-World, Conceptual Model & Simulation Model Relations

Some operational/functional simulation system requirements do depend on the fidelity requirements for the real-world to be represented. For instance, if a suitable real-world representation requires the reproduction of motion que's is experienced in a vehicle a motion system is usually selected. Inclusion of a motion system as part of the simulation model implies new operational requirements such as starting and stopping the hydraulic pump driving the motion base or a neat procedure to return the motion system to a save condition in case of an emergency. Hence these requirements are also considered by many to be a result of conceptual modeling [76]. However, they are not treated here as a part of the conceptual model but as a necessary side product from both conceptual modeling activities and user requirements.

Simulation Design

During simulation design the simulation model requirements together with the operational/functional requirements and constraints are translated in to a detailed design for the complete simulation system. The simulation model requirements for computer models are translated in a hard and software design using common computer and software engineering methodologies such as Yourdon, OMT and UML [92] [114]. From the requirements for physical models the required hardware and interfaces to the computer models are designed using various mechanical, electronic and computer engineering methodologies. For instance the structural design of a motion platform, its actuators and control logic. Similarly the operational/functional requirements for the simulation support systems are designed. In all these designs the constraints have to be taken into account and when necessary compromises have to be made in the design. This means that the real-world representation as specified by this design might differ from the conceptual model and therefore possibly not capable of meeting the user requirements. If this is the case feedback to the user who has specified the requirements is needed.

Simulation Development

Simulation development yields the actual physical realization of the simulation system and its contained simulation models as prescribed by the simulation design. The required hardware is build and software design is implemented in a programming language and compiled. Or when components are readily available off the shelf through reuse of existing components or from suppliers, these are selected and obtained. Finally, all components are integrated into a complete simulation system. Unforeseen practical constraints and problems could emerge during development that causes the actual implementation to differ from its design. As a result the achieved real-world representation of the developed simulation system might differ from the conceptual model and therefore possibly not capable of meeting the user requirements.

Simulation Testing

The previous development step is error prone. Typographic errors made during the translation of the software design into the actual code are examples of such development errors. Therefore, the realized simulation system must be carefully tested for errors and problems not yet identified or not properly addressed in the simulation development step to ensure that it correctly reflects its intended design. Any problems and errors have to be corrected here whenever possible. Again such problems or errors may cause the achieved real-world representation to deviate from it requirements but also effect the other required operational and functional capabilities. Testing usually is performed in two stages: component and integration testing. Component testing comprises checking all parts of the simulation separately for isolated development errors while integration test checks for errors of the simulation system as an integrated whole. To enable proper and effective testing, simulation system in test plans have to be made early on in the simulation development process in conjunction with the requirements definition activities.

Operational Usage

The last step in the simulation development or life-cycle is the actual operation of the simulation system by its users. Operational usage may result in that the user discovers that the simulation system doesn't work or doesn't provide the simulation(s) required. One cause of this problem could be that the user requirements and its derived simulation system requirements were not properly elicited and formulated i.e. the simulation system addresses the wrong problem or purpose. A second cause is wrong usage of the provided capabilities of the simulation system by its user. This could readily happen when the users is offered a lot of freedom in creating different real-world configurations and carelessly starts experimenting with compositions of simulation systems and simulation model configuration settings for initially unforeseen purposes. Even though such real-world configurations might be practically realizable, they could reside outside its intended and tested operational ranges resulting in simulations not behaving as might be expected. Basically what happens in such cases is that the user starts to develop a new simulation system for a different purpose through reusing readily available simulation system building blocks. Users should therefore properly be instructed in the use and limitations of the simulation system capabilities and prevented from 'miss-using' the offered freedom.

3.3.2 Validation, Verification and Accreditation Process

The verification, validation and accreditation process (VV&A) is closely related to the simulation development process and is considered by many to be an essential process in assessing the credibility and validity of the simulation. Unfortunately, again many variations and interpretations of these terms exist. The goal of the VV&A process can be formulated as to assure the development of correct and valid simulations and to provide its users with sufficient information to determine if the simulation fits their intended application purpose. VV&A consists of three interrelated activities to accomplish this goal. *Verification* is defined as the process of determining that a model or simulation implementation accurately represents the developer's conceptual descriptions and specifications [47]. Verification thus establishes whether the simulation system has been build properly. *Validation* is defined as the process of determining the degree to which a simulation is an accurate representation of the real world from the perspective of intended uses of the simulation [47]. In other words validation addresses the issue of whether the right simulation is build. Finally *accreditation* is the official certification that the model or simulation and its associated data is acceptable for use for a specific purpose [22]. This accreditation decision not only depends on the results of verification and validation but also on the quality of both processes and the people that executed these processes in order to provide enough evidence for simulation credibility i.e. can the results be trusted. The importance of VV&A should therefore not be underestimated. Most simulation developers and users agree on this, but fail to see that VV&A is a very complex process, which needs to be planned and executed carefully during the whole simulation development process before credibility of the simulation can be established. Usually, simulation VV&A is executed as a discrete and ad hoc comparison of several simulation results with some reference after the simulation has been built. Hence, it is not hard to understand that the concept of fidelity is an essential and fundamental aspect of VV&A.

A generic description of the VV&A processes is described in the Defense Modeling and Simulation Office (DMSO) VV&A recommended practice guide [22]. That process consists of 6 major steps. Without going into too much detail these six VV&A process steps can be mapped to the model and simulation development process discussed in the precious section (3.3). First the following two terms are introduced: *VV&A agent* and *subject matter expert*. A *VV&A agent* is a person involved and responsible for the proper execution of the whole VV&A process. A is *subject matter expert* (SME) is a person recognized as an authority in a specific area (domain expertise, simulation technology expertise, etc.) and has some appointed expert opinion role in either the model and simulation development or the VV&A process.

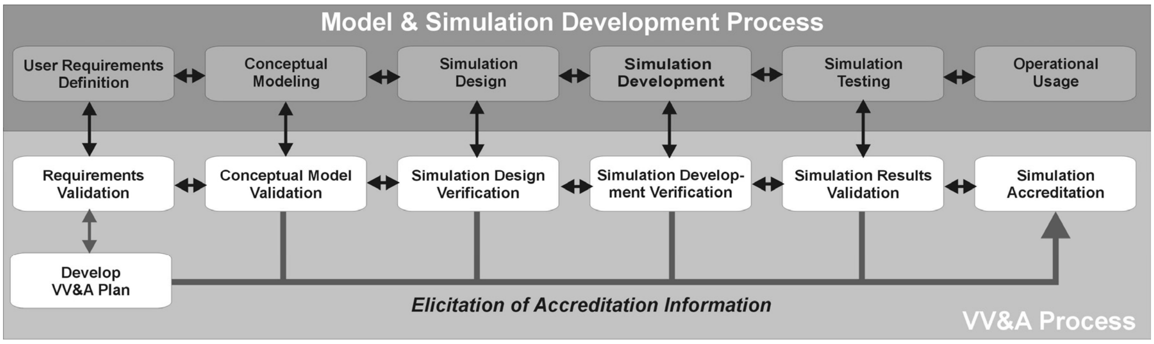


Figure 3-6 Verification, Validation & Accreditation Process Overview

The steps of the VV&A process as sketched in *Figure 3-6* are discussed in more detail in the next six paragraphs

Requirements Validation

The activities are twofold in this first step of the VV&A process. First a VV&A plan is developed that outlines the overall-strategy of how the process will be executed in terms of objectives, tasks, products, evaluation and acceptability assessment criteria, schedule, and designation of VV&A agents etc. Secondly, the user requirements are validated here by analyzing and assuring that the requirements are correct, consistent, clear and complete for the intended simulation system usage and problem domain.

Conceptual Model Validation

Conceptual model validation is performed to help demonstrate the simulation correctness and to enhance simulation credibility. It is assured that the developed real-world representational and the quality requirements are correct and fit the intended simulation system usage as prescribed by the user requirements. Next, the simulation model concept is evaluated to assess its overall real-world representational capabilities i.e. completeness, quality, limitations and assumptions. With this information it is then possible to determine whether the simulation model concept capabilities can meet the required real-world representational and quality requirements. Stated differently: it is assessed whether the simulation model as envisioned is likely to provide results realistic enough for the intended use.

Simulation Design Verification

Design verification is performed to ensure that the design accurately reflects the intent of the conceptual model and that no capabilities are omitted and altered or new capabilities are added affecting the realism of the simulation. Design verification includes also verification of network requirements, physical connections, and delineation of platforms against the developer's specifications.

Simulation Development Verification

In this stage VV&A performs tests on the developed simulation system components to ensure that the implementation of the simulation system accurately reflects the intended design and that no errors or unwanted changes are made.

Simulation Results Validation

In this step the simulation system is executed and its simulation results are analyzed. During the preparation for this execution the VV&A team makes sure that the simulation system is configured correctly and that the operators are properly instructed in the use of the simulation system. An acceptability assessment is conducted to determine whether the simulation system as implemented meets the real-world representational and quality requirements as stated in the conceptual model, and identifies any shortfalls and what their impact may be on the simulation results. If the simulation results do meet these validation assessment criteria they are labeled as valid.

Simulation Accreditation Assessment

The last activity is an accreditation assessment to establish that the simulation system indeed provides valid simulations, which do meet the user requirements with sufficient and acceptable credibility. This activity already starts in the first stage of the simulation development and VV&A process by collecting all necessary appropriate information as documented in the VV&A plan. After that the accreditation report and recommendations are prepared for the user. Finally, it is the user who makes the decision whether to accept the simulation system or not, based on the simulation system accreditation information and the risks associated with using the resulting simulation for the intended application purpose.

3.4 Abstract System Descriptions for Modeling and Simulation

To be able to develop a useful simulation fidelity framework, its development must be rooted in a general modeling and simulation theory (See 2.5.6). The foundation for such a modeling and simulation theory is provided by a hierarchy of system knowledge specifications [155] [156]. These specifications are based on principle systems engineering concepts or systems theory as will be outlined in section 3.4.1. Abstract system theory approach is utilized for a formal representation of the various types of system knowledge (Sections 3.4.3, 3.4.4, 3.4.5, 3.4.6). The concepts and formalisms developed here provide the necessary language for well-structured and consistent specification of system knowledge for the simuland, simulation models and simulation systems used in the modeling and simulation enterprise (section 3.3). All with the objective to provide the fundamental conceptual and mathematical formalisms for the formal definition of fidelity, its associated concepts and measurement methods discussed in the subsequent chapters.

3.4.1 Hierarchical Object Oriented System Specification Approach

The reality surrounding us can be thought of as being composed of tangible objects of various kinds. Objects are defined as thee human intuitive perception and abstraction of a separate unit in reality. They provide an encapsulating relationship that ensures a strong internal cohesion, and a weaker dependency or interaction with its environment. Objects in this regard refer to the application general object-orientation approach to the real-world domain in order to structure our understanding and knowledge of reality and its phenomenon, which is also known as object oriented modeling. This should not be confused with object oriented programming and associated languages for software development. Although both are based on the same general object-orientation concept they are not similar. Obviously, object-oriented models can be translated into a simulation computer model using object-oriented languages but this is not mandatory. A system is defined as an object or an organized group of objects forming a unified whole, which provides a specific set of functions. Interactions enable systems to influence (i.e. system output) and be influenced (i.e. system input) by other systems in its surroundings and therefore contributes to the evolution of its environment. In this environment a system fulfils a certain role and responsibility.

System theory approaches define system knowledge specification as a description of how a system behaves and of the mechanisms that make the system behave the way it does [99] [155]. This implies that system specifications comprise two major elements:

- *Behavioral description (dynamic element):* specifies the observable manifestation of how the system's characteristic features changes over-time.
- *Structural description (static element):* specifies the inner structural constitution and working of a system, and dependencies on the environment.

An important aspect of a system not explicitly addressed in classical systems approaches is the concept of identity. Identity characterizes the systems own existence, and enables to discriminate between systems in an unambiguous manner independent of its behavior or structural representation.

The principle of the structural decomposition of a system in a set of smaller more manageable parts (sub systems) or vice versa composition, is a natural way to deal with complex systems. This concept is known as the composite-component paradigm and is also one of the cognitive regularities of the human mind to organize or structure knowledge. Decomposition and composition principles make it possible to hierarchically structure and specify real-world system knowledge at different abstraction or aggregation levels in a uniform and intuitive fashion. In other words the real-world is considered here to be a system hierarchically composed of set of interacting (sub) systems (*Figure 3-7*). It can be proven that systems theory is closed under composition [156]. This means that systems composed of other systems can be expressed in the original systems theory terms and always exhibits well-defined structure and behavior.

The structural and behavioral system characteristics are perceived directly from those system properties that are externally observable or by an approximating set of other observable properties that are monitored and recorded by an agent over a period of time (i.e. time trajectories or histories). System properties are divided in four categories: input, output, state and parametric variables (*Figure 3-7*). The set of system input and output properties and recorded values specify how the system evolves over-time (system dynamics) from a black-box perspective or system input/output behavior, and how the system is influenced and how the system influences other systems in its environment (interaction). The interaction of systems is perceived through the external observable properties of input and output variables. System input thus defines how a system can externally be controlled or what information enters through the system boundary from other systems in its environment. System output variables specify what system information or products leave the system through its boundary to effect systems in its environment. State variables specify the internal status and conditions of a system over-time. Parametric variables specify those internal system properties that determine the system characteristics. Parameters are independent from the system state variables and mostly they are perceived to be constant within the observation timeframe. Together with the system state variables parameters determine the internal transition mechanism of how a system generates its outputs from its inputs. In practice state and parametric variables are either known by direct observation of the system internals when possible (white-box observation) or these variables are traced back from the system input/output variables (black-box observation). The latter case is usually caused by technical and resource constraints placed upon the direct observation of the systems internals and one has to fall back on techniques like state observers and system identification [80] [99].

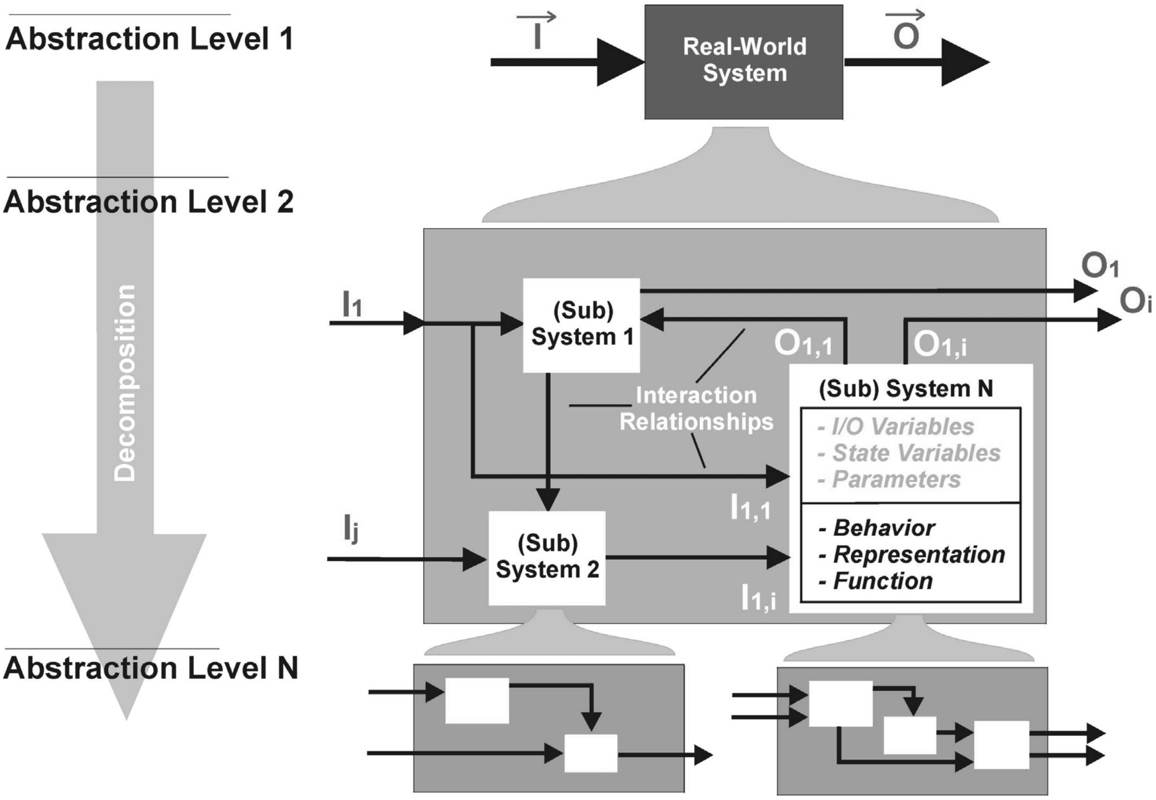


Figure 3-7 Hierarchical Object Oriented System Specification

Functional modeling approaches consider systems as an artifact with certain goals and functions [94]. These approaches provide another important cognitive regularity of the human mind to organize knowledge called means-end. The means-end concept specifies the way systems or components interact with each other to achieve useful overall behavior. Means-end starts with specifying what goals a system fulfils or needs to fulfill in its environment from a purpose perspective. Secondly it addresses what behavioral capabilities the system must poses to accomplish these goals. Finally, like in the classical system theory, it addresses how these behavioral capabilities are realized in terms of system internal structure (components and interactions etc.) and workings. Here goal is defined as the outcome to which certain activities of a system are directed. The concept of functional capability yields what role a system fulfils in the environment in terms of achieving of one or more goals. It should be noted that goals and functions are not inherent properties of systems, which means that they are characteristic properties ascribed to systems when considered in a certain context. Functional capabilities relate to system behavior as follows, a function is a subset of system behavior useful or of interest to its environment in a given context (*Figure 3-7*). System behavior specification is based on a pure context free observation of a system. Functional capabilities are thus a qualitative way of a set of system behaviors and explicitly specifying the roles of each system within a certain application context. Another characteristic of system functions is that they can be decomposed in a subset of smaller interacting functions realizing the overall parent system function. At the lowest level of decomposition these functions directly relate to a set of behavioral properties and internal workings of a system (I/O variables, state variables, parameters etc.). From a modeling and simulation application view, models and simulation systems are always developed and used with an application purpose in mind. Thus any system specification utilized in model and simulation development is an artifact to which identifiable goals and functional capabilities can be assigned. More on this important notion can be found in the subsequent sub-sections.

System knowledge can be specified at different levels (*Figure 3-7*) ranging from a high level description specifying only input and output to a low-level detailed description of the system [99] [156]. However, all these levels can be classified in two primary systems engineering specification levels: external and internal system specification. These system specification concepts are further discussed in the next sections.

3.4.2 Rationale for Modeling and Simulation Formalisms

So far an informal approach has been used for the definitions, concepts and processes of the modeling and simulation enterprise. Since modeling and simulation is still an art form it has produced many fuzzy and imprecise practices without sound mathematical foundations. With the increasing usage and complexity of simulations across different application domains the lack of more mathematical formalism hinders the rigorous treatment of simulation systems and the analysis of their outcomes. It especially complicates the understanding, analysis, management and reuse of the large amounts of simulation system information to be handled in today's simulation development and validation processes.

For the concept of fidelity as an extrinsic part of the modeling and simulation enterprise this is not much different (Section 2.5). To help demystify and develop a more precise definition for fidelity as well as for it's related concepts mathematical formalisms for modeling and simulation are indispensable. Such formalisms facilitate the long overdue more objective and structured approaches to fidelity characterization, and their quantitative specification where possible. Finally, a more mathematical rigorous approach supports the development of automated tools that can assist in the fidelity assessment and other simulation development and validation tasks.

The mathematical formalisms discussed in the next sections of this chapter build and expand upon results from other research efforts into modeling and simulation formalisms [8] [47] [155] [156]. Except for standard mathematical notations from discrete mathematics, set theory and system theory, these formalisms do not contain any unique simulation application or problem domain specific elements. The reader is referred to standard textbooks on discrete mathematics, set theory and mathematical system theory for detailed information on the used mathematical notations and formulations in here [78] [80] [99].

3.4.3 Behavior Specification Formalisms: Time base, Trajectory & Segment

Despite the fact that the concept of time is difficult to understand, its effects are known and it is used beneficially in many ways. In real-life, time is considered to be a fundamental mechanism underlying our perception and description of how the real-world evolves, i.e. capturing changes over a specific period of duration and an ordering mechanism for occurrences of changes. Mathematical system theory provides a common pragmatic approach to deal with the concept of time in modeling and simulation by defining what is called a time-base [99] [156]. In here time is assumed to be a variable orthogonal to all any other variables. Independence of time from spatial variables is a valid assumption in all cases except those needing the application of Einstein's theory of relativity. Such applications are beyond the scope of this work. The concept of time-base and its derived elements provide the terminology to discuss and analyze effects of time on simulation fidelity in subsequent chapters. A time base is defined as:

$$time = \langle T, < \rangle \tag{3.1}$$

where T is a set and < is the ordering mechanism operating on the elements of T. The ordering mechanism is transitive, irreflexive, anti-symmetric and *linear*. Linear ordering means that for every pair (t,t') either t < t' , t = t' or t > t'. Otherwise the ordering is designated as *partial*. Time base T could either be real numbers $\Re$, integers $\Im$, and all sets $c * \Im, c \in \Re$ a constant, isomorphic to $\Im$. Continuous time is represented by the time base $T_{\Re}$ and discrete time is represented by isomorphic time bases $T_{\Im}$. If the *present* time is represented by t then *past*, *future* and a closed *time interval* $[t_1, t_2]$ are defined respectively by:

$$T_{t>} = \{\tau | \tau \in T, \tau < t\}$$

$$T_{<t} = \{\tau | \tau \in T, t < \tau\} \tag{3.2}$$

$$T_{[t_1,t_2]} = \{\tau | \tau \in T, t_1 \le \tau \le t_2\}$$

where t_1 and t_2 are respectively called initial time and final time. Every time base has a minimum element t_0 such that $t_0 \le t$ for all $t \in T$ and a maximum element t_n such that $t \le t_n$ for all $t \in T$. Both t_0 and t_n may be respectively $-\infty$ and ∞ indicating there are no lower or upper bounds defined for the time base. This is called an *infinite time base*.

Given a time-base T it is possible to describe real-world behavior, i.e. variable changes over time, using a time function called *trajectory* or *signal*, which can represent an input, output or state variable set A of a real-world system as follows:

$$f : T \rightarrow A \tag{3.3}$$

In 3.3 the value of f at time t_i is given by $a_i = f(t_i)$ for all $t_i \in T$ and $a_i \in A$. Since T is an ordered set the set A resulting from this time mapping f will also be an ordered set. A time function f restricted to a time interval $[t_1, t_2]$ is called a *segment* and is defined as:

$$\omega :< t_1, t_2 > \rightarrow A \text{ or } \omega_{<t_1,t_2>} \tag{3.4}$$

The segment $\omega_{<t_1,t_2>}$ describes the motion through set A that begins at t_1 and ends at t_2, and for every $t \in [t_1, t_2]$, $\omega(t)$ describes what the value of the trajectory or signal is at time t. The *length* of a segment is defined as the length of time between the begin and end time of the segment: $l(\omega) = t_2 - t_1$. The *domain* of a segment is defined as the closed interval between the segment begin and end time: $dom(\omega) = [t_1, t_2]$. A pair of segments $\omega_{<t_1,t_2>}$ and $\omega_{<t_3,t_4>}$ are said to be *contiguous* if their domains are *contiguous*, i.e. $t_2 = t_3$. For contiguous segments a *concatenation* operation '$\bullet$' is defined:

$$\omega_1 \bullet \omega_2 :< t_1, t_4 > \rightarrow A \tag{3.5}$$

with

$$\begin{aligned} \omega_1 \bullet \omega_2(t) &= \omega_1(t) \text{ for } t \in < t_1, t_2 > \\ \omega_1 \bullet \omega_2(t) &= \omega_2(t) \text{ for } t \in < t_3, t_4 > \end{aligned} \tag{3.6}$$

A set Ω of segments over A and T is called closed under concatenation if for each contiguous pair $\omega_1, \omega_2 \in \Omega$ also $\omega_1 \bullet \omega_2 \in \Omega$ [156]. Set Ω is called closed under left segmentation if every left segment defined over A and the open interval $[t_1, t_2) \in T$ is also part of the set Ω.

Many segments can be defined by combining a type of time-base with a vector space. Three often recurring segments are: *continuous segments*, *event segments* and *sequences*. A continuous segment has signal or trajectory values that moves continuously through a n-dimensional vector space $\Re^n$ with $n \in \mathrm{N}$ over a continuous time base $T_{\Re}$ or:

$$\omega :< t_1, t_2 > \rightarrow \Re^n \text{ with } t \in [t_1, t_2] \tag{3.7}$$

A *piecewise continuous* segment is a segment continuous at all times t except at a finite number of points in the interval $[t_1, t_2]$. An event segment represents a continuous time base (T) ordered series of events:

$$\omega :< t_1, t_2 > \rightarrow A \cup \{\varnothing\} \tag{3.8}$$

In *Expression 3.8* $\varnothing$ denotes the *nonevent* which is not an element of *A*. Then ω is an event segment if there exist a finite set of *n* time points $t_i \in [t_1, t_2]$ such that $\omega(t_i) = a_i \in A$ for $i = 1,...,n$, and $\omega(t) = \varnothing$ for all other $t \in [t_1, t_2]$. Segments that are defined over a discrete time base $T_{\mathbb{Z}}$ are referred as a *sequence*.

3.4.4 External System Knowledge Specification Formalism

The external system knowledge specification is the most basic system abstraction and considers the system as a black box. In other words it doesn't specify the system's internal working and structure. An external system specification is defined in terms of its observation time base, inputs, outputs, input-output relations and functional capabilities. The input of a system is characterized by a set of *n* independent input variables *I*: $I = \{u_1, u_{2,} ...., u_n\}$ and their associated range sets of possible values $\{U_1, U_2, ..., U_n\}$. The set of all possible assignments to the input variables is represented by their cross product:

$$U_1 \times U_2 \times .... \times U_n = \{(v_1,..., v_n) | v_1 \in U_1,..., v_n \in U_n\} \quad (3.9)$$

where v_j represents a possible value of the input variable u_j. Then the system input is formally defined by the next multi-variable set *U*:

$$U = (I, U_1 \times U_2 \times ... \times U_n) \quad (3.10)$$

There exist two standard multi-variable set operations to retrieve the variable set and for each variable its belonging range set: *variables* and $range_{ij}$ [156]. Thus the system input is *variables(U)* = *I* and the range of the input u_j is denoted as $range_{u_j}(U) = U_j$. For instance, for an aircraft the input *U* can be defined as follows *variables(U)* = *I* = {*controlCollumPosition, flapLeverSetting, throttleLeverSetting*} then the range for the input variable u_2 = *(flapLeverSetting)* can be for instance $\mathrm{range}_{u_2}(\mathrm{U})$ = U_2= {0^0, 5^0, 15^0, 20^0, 30^0}.

Likewise, the output can be defined by the following multi-variable set *Y*:

$$Y = (O, Y_1 \times Y_2 \times ... \times Y_m) \quad (3.11)$$

In here output *variables(Y)* $= O = \{y_1, y_2, ..., y_m\}$ and the range of the output y_m is denoted as $range_{y_j}(Y) = Y_j$. For instance the aircraft output y_2 = *flapPosition* and with its associated range defined as the closed interval over $\Re$ $range_{y_2}(Y) = Y_2 = [0^0, 30^0]$.

Together with a time-base *T*, the input and output form a set $S_{IOF} = (T, U, Y)$, which is referred in literature as the I/O observation Frame (IOF) of system *S* [156].

Using the knowledge of S_{IOF} it is possible to start external system behavior observations or experiments by applying an input segment $\omega \in (T, U)$ and registering the corresponding output segment $\rho \in (T, Y)$. An input segment and the corresponding output segment together forms an ordered pair (ω,ρ) and is called an *I/O pair* or *external behavior instance* of a system. All registered I/O pairs united form a set B_{ext} called the *external behavior sample* of a system. The set of all registered input segments is referred to as Ω and is called the system *external input sample*. The related set of all registered outputs, the system *external output* sample, is denoted as Γ. The set B_{ext} thus represents the following binary relation between Ω and Γ:

$$B_{ext} \subseteq \Omega \times \Gamma \tag{3.12}$$

with $\Omega \subseteq (T,U)$, $\Gamma \subseteq (T,Y)$ and $(\omega,\rho) \in B_{ext} \Rightarrow dom(\omega) = dom(\rho)$. This relation from Ω to Γ doesn't necessary imply that their exists a unique inverse relation B_{ext}^{-1} from Γ to Ω. In other words the same input segment applied more than once may result in multiple different output segments. Obviously, this effect is caused by the difference in the system's internal initial conditions (time-dependencies, state variables and parameters) and the possible stochastic nature of the system internal workings. Useful external system behaviors are described in terms of the set F_{cap} specifying all $k \in N$ functional capabilities that a system exhibits:

$$F_{cap} = \left\{ f_{cap_1}, \ldots, f_{cap_k} \right\} \tag{3.13}$$

$f_{cap_i} \in F_{cap}$ is the functional capability description (see Section 3.4.1):

$$f_{cap_i} = \left(G_i, \overline{U}_i, \overline{Y}_i \right) \tag{3.14}$$

where the goal description G_i describes what goal a functional capability is to accomplish or what intended system purpose it serves. $\overline{U}_i \subseteq U$ and $\overline{Y}_i \subseteq Y$ specify those system input and output variables that characterize or are involved in a system functional capability. As an example a secondary radar system has a functional capability of air surveillance within a civil airspace system. The goal description $G_{surveillance}$ is defined as the identification and detection of 3D aircraft position in a designated part of the airspace. The radar system output set $\overline{Y}_{surveillance}$ involved in this interaction with an aircraft consists of an interrogation signal send to the aircraft and the response signal generated by the onboard transponder serves as the necessary input set $\overline{U}_{surveillance}$ in order to fulfill the radar air surveillance functional capability goal $G_{surveillance}$. Utilizing the previous definitions an external system knowledge specification can now be formally defined by the following structure:

$$S_{ext} = \left(T, U, Y, B_{ext}, F_{cap} \right) \tag{3.15}$$

3.4.5 Internal System Knowledge Specification Formalism

The internal system knowledge specification compared to the external system knowledge specification takes the approach of considering the system as 'glass-box'. Internal system knowledge specification takes the information specified by expression (*3.15*) and adds to it the system internal working and structure knowledge. As discussed in section 3.4.1 the internal system structure and condition is characterized in terms of system state and parameter variables.

The system state is defined by the multi-variable set *Q*:

$$Q = (X, Q_1 \times Q_2 \times ... \times Q_n) \tag{3.16}$$

Here *variables(Q)* = *X* are the state variables and the range of the state variable $q_j \in X$ is denoted as $range_{q_j}(Q) = Q_j$. *P* is the set of system parameters as defined by the expression:

$$P = (Z, P_1 \times P_2 \times ... \times P_m) \tag{3.17}$$

Here *variables(P)* = *Z* are the parameters and the range of the parameters $p_j \in Z$ is denoted as $range_{p_j}(P) = P_j$. The system internal working is described in two parts. Firstly by means of a state transition or evolution function, which describes how the internal state evolves over time and secondly by what is generally known as the output function or observation function [99] [156]. The state transition function Δ maps the current state at time t_1, parameter setting and input to another state at time t_2 as follows:

$$\Delta : T^2{}_+ \times Q \times P \times U \rightarrow Q \tag{3.18}$$

Where $T^2{}_+ = \{(t_1, t_2) \in T^2 | t_2 \geq t_1\}$. Furthermore, Δ satisfies the constraints that each input segment $\omega \in (T, U)$ is closed under concatenation and left segmentation and fulfils the semi-group property for each pair input segments $\omega_{\langle t_1, t_2 \rangle}$, $\omega'_{\langle t_2, t_{t_3} \rangle}$: $\Delta(t_1, t_{3,} q_{t_1}, p, \omega \bullet \omega') = \Delta(t_2, t_3, \Delta(t_1, t_2, q_{t_1}, p, \omega), p, \omega')$. This state transition function can be a function of any kind and is thus not constraint to linear or deterministic internal system behavior. The system output function Λ maps each current state, parameter and input variables to a set of values for the system output variables:

$$\Lambda : T \times P \times Q \times U \rightarrow Y \tag{3.19}$$

Given a state $q_{t_1} \in Q$ at $t{=}t_1$, $p \in P$ and an input segment $\omega :< t_1, t_2 > \rightarrow U$ and using mappings *3.18* and *3.19* than the state at time t_2 is given by $\Delta(t_1, t_2, q_{t_1}, p, \omega)$ and the final output is given by $\Lambda(\Delta(t_1, t_2, q_{t_1}, p, \omega), \omega(t_2), p, t_2)$. Combining all previous information the formal internal system specification yields:

$$S_{in} = (T, U, Y, F_{cap}, P, Q, \Delta, \Lambda) \tag{3.20}$$

It should be noted that Δ and Λ are abstract ways to specify the system internal working in a structured manner. The actual styles or language that is used in practical model and simulation development to express Δ and Λ depends on many factors and often multiple languages are needed for a sufficient specification of real-world systems. Languages can range from textual, graphical to mathematical-physical descriptions. Even an implemented set of computer algorithms to represent a real-world system within a simulation model is an example of such language. Factors that determine which specific language is needed include the nature of the system, the type of knowledge available, particular application and problem domain needs or standards, and the stage of model and simulation development (section 3.3 and 3.3.2) in which such specifications are used. Next chapters will address these issues and how they relate to simulation fidelity assessment. Note that system input, output, state and parameter are widely interpreted concepts that specify all possible system behavioral and representational information, in addition to the more conventional (pure mathematical or numerical representations) notion. For instance, the output of a traffic light could have color as a variable with a range of {red, orange, green}.

Internal system specifications are often called structural specifications since they specify the internal system structure and working to generate behavior in terms of an *internal* and *external behavior instance*. Here *internal behavior instance* refers to a registration of a state trajectory $q \in (T,Q)$. Both the internal and external system behavior instances can be reconstructed from (*3.20*) by taking an initial state $q_{t_1} \in Q$ at $t=t_1$, a parameter setting $p \in P$ and applying input segment $\omega :< t_1, t_2 > \rightarrow U$ as follows:

$$
\begin{aligned}
&q :< t_1, t_2 > \rightarrow Q \text{ with } q(t) = \Delta(t_1, t, q_{t_1}, p, \omega(t)) \; \forall \mathrm{t} \in < t_1, t_2 > \\
&\rho :< t_1, t_2 > \rightarrow Y \text{ with } \rho(t) = \Lambda(q(t), p, \omega(t), t) \; \forall \mathrm{t} \in < t_1, t_2 >
\end{aligned}
\qquad (3.21)
$$

For each allowable quadruple $(t_1, q_{t_t}, p, \omega)$ it is possible to construct or register a set of belonging *internal* (B_{in}) and *external* (B_{ext}) *behavior samples* indexed by the system initial time, state and parameter setting. The internal behavior sample is defined by:

$$B_{in} = \left\{ (\omega, q)_{t_1, q_1, p} \middle| t_1 \in T \wedge \omega \in (T,U) \wedge q \in (T,Q) \wedge p \in P \wedge q_1 \in Q \right\} \qquad (3.22)$$

with the constraint that $dom(\omega) = dom(q)$. Similarly the definition of external behavior given can now be refined as follows:

$$B_{ext} = \left\{ (\omega, \rho)_{t_1, q_1, p} \middle| t_1 \in T \wedge \omega \in (T,U) \wedge \rho \in (T,Y) \wedge p \in P \wedge q_1 \in Q \right\} \qquad (3.23)$$

with the constraint that $dom(\omega) = dom(\rho)$. Obviously, expression (*3.25*) can only be fully realized when in the actual system observation the internal state and parameters can be accessed or reconstructed through the system I/O variables (observability and identifyability criteria can be found in [80] [99]). Otherwise, the B_{ext} is given by the binary relation (*3.12*) with no or partial internal system knowledge. Being able to reconstruct a full initial state and unique parameter vector is necessary but not sufficient condition to obtain unique external behavior instances. The other remaining requirement to be able to do this is having a system internal working that is fully deterministic i.e. mappings Δ and Λ have to be deterministic. To distinguish between deterministic and stochastic mappings the following indices can be added to the internal system specification: Δ_{type} and Λ_{type} with $type \in \{Det, Sth\}$.

A special class of often-encountered deterministic systems is an invariant system. Such systems exhibit internal system workings that do not vary over time i.e. mappings Δ and Λ are not explicit functions of time. As a result an input segment $\omega' \in (T,U)$ when fed to a time invariant system at $t=t_1+\tau$ yields the same output segment as the original segment ω at $t=t_1$ $\rho \in (T,Y)$ but shifted a distance τ on the time base T under the same initial state and parameter settings. Moreover such systems can be formally defined as the internal system specification $S_{in} = (T,U,Y,F_{cap},P,Q,\Delta_{Det},\Lambda_{Det})$ with Δ and Λ fulfilling the constraints:

$$\begin{aligned} &\Delta_{Det}(t_1+\tau, t_2+\tau, q_1, p, \omega(t+\tau)) = \Delta_{Det}(t_1, t_2, q_1, p, \omega(t)) \\ &\Lambda_{Det}(q_1, p, \omega(t+\tau), t+\tau) = \Lambda_{Det}(q_1, p, \omega(t), t) \end{aligned} \tag{3.24}$$

where $q_1 \in Q$ is the initial state at $t=t_1$, $p \in P$, and an arbitrary input segment $\omega :< t_1, t_2 > \to U$.

3.4.6 Composite-Component System Knowledge Specification Formalism

As discussed in section 3.4.1 to properly deal with complex systems and their associated behavior, systems are usually addressed as being composed of a set of hierarchically ordered and interacting set of smaller subsystems. *Figure 3-7* gives the graphical representation for such system specification approach but it is possible to extend the formal system specification developed in the previous two sections to such systems as well. Such a system specification is in literature referred as coupled or composite-component system specification [156]. In this type of specification the system is composed of a series of components that are systems by themselves, which can be described by either an external (*3.15*) and/or an internal (*3.20*) system specification. The behavior of each component may be influenced by a set of other components or the external input of its parent system. Each component may influence the behavior of a set of other components or the external output of its parent system. Therefore, a composite-component system specification (S_{comp}) adds an extra level of knowledge to how a system is internally structured and how the overall or interactive system behavior is generated. This means that S_{comp} is a specialized form of an internal system specification S_{in} and is defined as follows:

$$\begin{aligned} S_{comp} = \Big(&T, U_c, Y_c, D, \{S_d \,|\, d \in D\}, \{I_d \,|\, d \in D \cup \{S_{comp}\}\}, \\ &\{E_d \,|\, d \in D \cup \{S_{comp}\}\}, F_{cap_c} \Big) \end{aligned} \tag{3.25}$$

Here, U_c and Y_c are the composite or parent system input and output respectively (see also *Figure 3-8*). The set D contains all component references that together form S_{comp}. S_d is the system specifications of a component d, which could take the form of either one of the following system specifications: internal (*3.20*), external (*3.15*) or in case the component is further decomposed it is a composite-component system specification (*3.25*) as well. The set E_d refers the to components that are affected by the behavior of a component d and could also include component d itself in case of a feedback loop or

S_{comp} in case the component output directly contributes to the external output Y_c. E_d is thus formally defined as:

$$E_d \subseteq D \cup \{S_{comp}\} \tag{3.26}$$

In expression (*3.25*) I_d represents those components influencing the component d by means of a set of component output to input mappings. These mappings specify how the input of a component is derived from other component's output and the external input U_c of S_{comp}. I_d is formally defined as follows:

$$I_d : \underset{j \in E_d}{\times} YU_j \rightarrow UY_d \text{ for } \forall d \in D \cup \{S_{comp}\} \tag{3.27}$$

and if

$$\begin{aligned}
& j = S_{comp} \rightarrow YU_j = U_c \\
& j \neq S_{comp} \rightarrow YU_j = Y_j \\
& j = S_{comp} \rightarrow UY_d : variables(UY_d) = variables(Y_c) \wedge range(UY_d) \subseteq range(Y_c) \\
& j \neq S_{comp} \rightarrow UY_d : variables(UY_d) = variables(U_j) \wedge range(UY_d) \subseteq range(U_j)
\end{aligned}$$

It should be noted that in expression (*3.27*) the external system output Y_c of S_{comp} is also included and represented by a similar mapping, which allows for specification of external output variables that uniquely characterize certain aggregated properties of S_{comp}. In other words these are properties that cannot be accommodated by a property of a single component of S_{comp}. For example, if S_{comp} represents an aircraft powerplant composed of two turbojet engines then the overall powerplant properties of total fuel flow are the summation of both engines separate fuel flows. To illustrate this coupling mechanism of a system S_{comp} as exemplified by expressions (*3.26*) and (*3.27*) the next figure is given

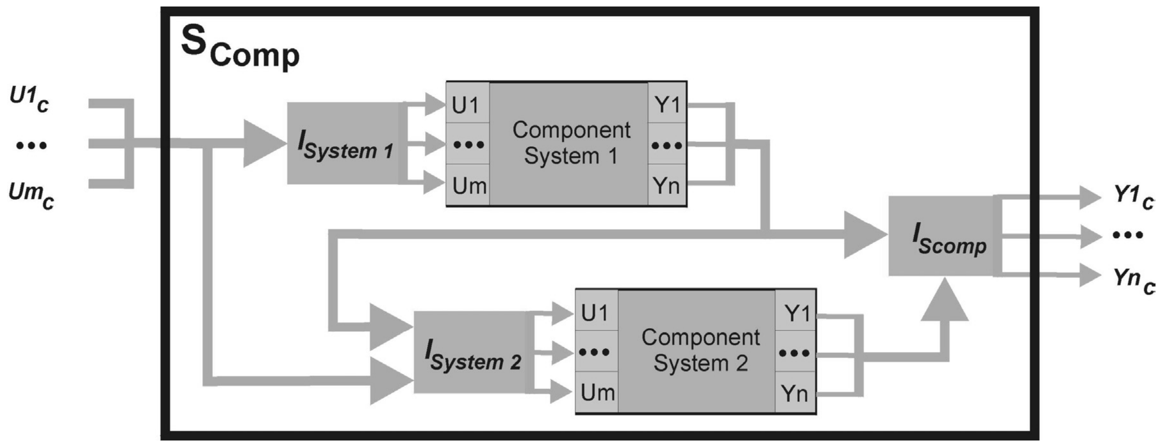


Figure 3-8 Composite Component System Specification

The last element in expression (*3.25*) F_{cap_c} specifies the means-end hierarchy of S_{comp} as being defined by the set of overall functional capability of S_{comp}:

$$F_{cap_c} = \left\{ \tilde{f}_{cap\,1}, \ldots, \tilde{f}_{cap\,m} \right\} \tag{3.28}$$

where $\tilde{f}_{cap_i}$ is:

$$\tilde{f}_{cap_i} = \left(G_{c_i}, \overline{U}_{c_i}, \overline{Y}_{c_i}, \overline{F}_{cap_i} \right) \tag{3.29}$$

Like in *Expression 3.13* $G_{c_i}, \overline{U}_{c_i}, \overline{Y}_{c_i}$ specify respectively the goal description and the system input and outputs for this overall functional capability $\tilde{f}_{cap_i}$. The set of functional capabilities of the components of S_{comp} that contribute to realization of the overall functional capability $\tilde{f}_{cap_i}$ is specified as:

$$\overline{F}_{cap_i} \subseteq \bigcup_{d \in D} F_{cap_d} \tag{3.30}$$

Here F_{cap_d} is the functional capability set of the component *d* (*3.14*). In case component d is a composite-component system specification itself $F_{cap_d} = \overline{F}_{cap_d}$, which gives access to deeper underlying subsystem functional capabilities that contribute to a higher level functional system capability. To illustrate this the functional capability example of a secondary surveillance radar (Section 3.4.4) is elaborated upon here. Suppose that the radar is decomposed in two subsystems a radar antenna mounted on a rotational electric engine. Than the radar functional capability $\tilde{f}_{cap_{radar}}$ becomes for instance:

{ G_{radar} = {provide air-surveillance in a circular area around the radar position},
$\overline{U}_{radar}$ = {detection range},
$\overline{Y}_{radar}$ = {detected aircraft plot},
$\overline{F}_{cap_{radar}} = \left\{ F_{cap_{antenna}}, F_{cap_{engine}} \right\}$ }.

The functional capability of the antenna is defined by its goal description $G_{antenna}$ formulated as the detection of 3D aircraft position within the antenna pattern area. Output and input for $F_{cap_{antenna}}$ are respectively the interrogation signal send to the aircraft and the response signal send by the onboard transponder. The goal description G_{engine} of the rotational engine functional capability is changing the orientation of the antenna. Its associate input is for instance the electrical power that drives the electric engine and the output is the rotation angle of the electric engine. Both mentioned subsystem functional capabilities are required to properly accomplish the air surveillance radar's goal G_{radar}.

Expression 3.25 provides the most general specification for composite-component systems. However, in modeling and simulation often, like for instance in the object-oriented modeling tool Dymola [24], composite-component systems are described by means of direct coupling of certain component output ports to other component's input ports. It can be proven that such a composite-component system specification is a specialized form of *Expression 3.25* [156].

3.5 Summary

This chapter has outlined a general applicable modeling and simulation framework for discussing, considering, specifying, using and developing models and simulations. It introduced and explained the necessary modeling and simulation terminology in order to facilitate the understanding and transmission of the fundamental fidelity related-terms and concepts developed in Chapter 4. An outline of the modeling and simulation enterprise has been discussed. This discussion focused on the two main engineering processes of the enterprise: the model and simulation development process and the verification, validation and accreditation process. Both these processes provide the contextual background for identifying the place and role of the concept of fidelity (Chapter 4), its assessment and application (Chapter 8) within the simulation system life cycle. Furthermore, both engineering processes provide the hooks for the development and insertion of a fidelity management process model within this life cycle (Chapter 9). The last sections of this chapter developed and discussed some principle object-oriented system specification formalisms for modeling and simulation purposes. These formalisms are the mathematical foundations for the development of the fidelity referent knowledge specification formalisms and associated fidelity measurement methods in respectively Chapters 5 and 7.

4 Unified Fidelity Framework: Fundamentals

4.1 Introduction

As shown in Chapter 0 there exist many connotations of what is considered to be the fidelity of a simulation. These differences primarily originate from the even more different application perspectives and contexts in which these fidelity approaches have been developed and tailored to suite certain specific needs (Section 2.5.6). Therefore, these fidelity approaches are usually unrelated to each other, not easily understood by other simulation developers and are not generally applicable to different application and problem domains. The objective of this chapter is the development of the basic definitions and elements for a unified fidelity framework, which provides a general fidelity theory and practice for simulation development. This objective is achieved by synthesizing existing fidelity knowledge into a consistent and formal fidelity assessment approach and addressing their overall deficiencies. The next Chapters will elaborate on this unified fidelity framework in more detail. In order to facilitate its proper understanding and utilization within simulation development, the fidelity theory and practice builds upon the general modeling and simulation contextual framework outlined in Chapter 3.

This chapter is organized in three sections and starts in Section 4.2 with the discussion of the strongest formulation for fidelity and the essential elements underlying a sensible fidelity theory. In Section 4.3 this theory is extended with the fundamental concepts for pragmatic simulation fidelity characterization and measurement. The fidelity principles and concepts presented in these first two sections are formalized by mathematical specifications developed in Section 4.4. These specifications provide the formal definition of simulation fidelity.

4.2 Simulation Fidelity Theory: Origin, Essence and Aims

Like in any science, the development of a theory for simulation fidelity must be based on a set of solid definitions, facts, key principles and concepts, which are considered within a single context and form an integrated whole. In this section these fundamental elements underpinning a theory for simulation fidelity are developed. From a discussion of the fidelity issue origins the strongest possible definition for fidelity is postulated (Section 4.2.1). Next the reality of fidelity measurement is assessed in order to determine its essence and inherent limitations, and to demystify the unrealistic and false expectations so many have attributed to the concept of fidelity (Sections 4.2.2 and 4.2.3). This puts fidelity back into a more realistic and pragmatic perspective that enables the creation of a well-defined fidelity theory, which can serve as a useful tool for improving simulation quality and the related development and validation processes (Section 4.2.4).

4.2.1 Esoteric Fidelity: The Most Conceptually Right Definition of Fidelity

In essence simulating is nothing more than deliberately counterfeiting reality to serve certain user objectives. That means simulation is always a limited approximation of reality. Therefore, all simulation users have the same recurring question regarding its validity: 'Is the reproduction of reality by this particular simulation good enough for our purpose' (Section 2.5). Similarly, simulation developers have to deal with the same problem of how to develop a simulation that reproduces reality well enough that the user accepts it as a valid placeholder of reality. From both front-end perspectives three key issues can be derived that everybody involved in the simulation enterprise, in one way or another, is faced with:

1. Establishing the needed degree of correspondence between the desired simulation and reality to be suitable for a specific purpose.
2. Measuring the degree of correspondence between the resulting simulation and reality.
3. Determining whether the degree of correspondence between the resulting simulation and reality meets the needed degree of correspondence.

Analysis of existing modeling and simulation literature reveals that it is always one of these three issues or combinations thereof which are associated with or referred to by the term *fidelity* (see Chapter 0). The common denominator in these three issues is the ability to make a comparison between reality and simulated reality and to specify the degree of correspondence between both. It is this specification that is often called the degree of realism of a simulation and which is looked for by many in the simulation community. Ideally, this is indeed the most conceptual right formulation for the term *fidelity*. Based on this premise fidelity is best defined as (see *Figure 4-1*):

'The inverse difference between reality and simulated reality'

A compact formulation that also meets the requirements posed in Section 2.5 of being an application context and fitness-for-purpose free formulation, and the normal apprehension that high fidelity implies a small difference.

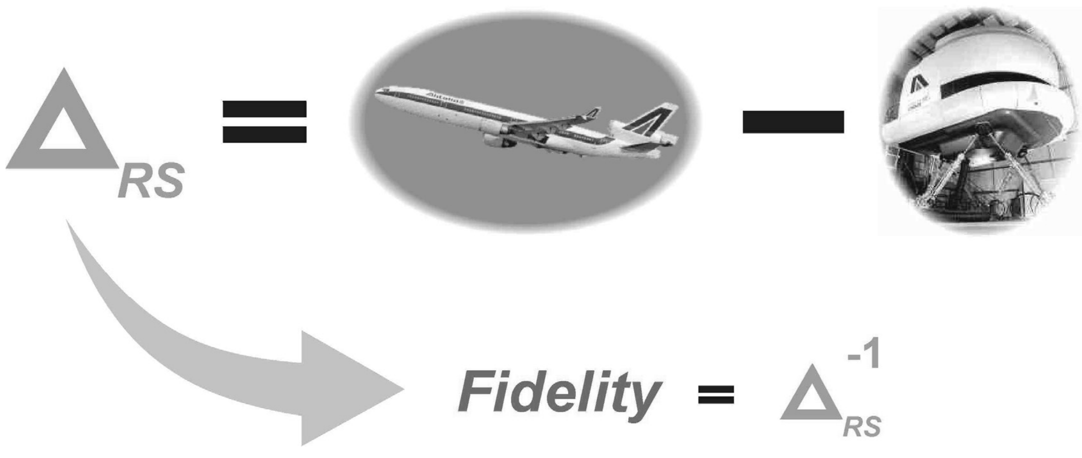


Figure 4-1 Strongest fidelity definition visualized for a MD11 flight simulator

Compared to the many existing definitions for fidelity (see Sections 2.2, 2.3, 2.4), the most beneficial property of this fidelity formulation is that it explicitly shows the essential principle underlying its measurement: the comparative analysis between reality and simulated reality. This definition is the strongest possible formulation for fidelity.

For reasons discussed in the next subsections (4.2.2 and 4.2.3) a practical implementation of such fidelity formulation is never attainable. Unawareness or failure to accept this has lead to many false beliefs and expectations that are attributed to the concept of fidelity and its associated practices, resulting in either skeptics or those who strive with dedication for unrealistic fidelity goals. As a result the fidelity concept is often put in perspectives that have hampered the development of practical fidelity assessment approaches that pertain some real usage in actual simulation development. Such perspectives are mentioned here as esoteric. Therefore, this strongest formulation possible for fidelity is referred here after as *esoteric fidelity*.

4.2.2 Basis of Fidelity Measurement: The Experience and Specification of Reality

The basis for modeling and simulation and its associated fidelity practices originates in the way the human beings perceive (see, feel, measure, etc.) the surrounding world, how they interpret and process these observations and subsequently how they abstract and hypothesize (universal laws of nature, etc.) in order to study, understand, specify, manipulate and profit from this very complex world. Reality in this regard is nothing more than the knowledge source for collecting information on what the surrounding world looks like and how it functions as it does. Even though it is conceptually true that we are all experiencing the same reality the perception, interpretation and specification of this reality varies from person to person. Ask two persons to describe aircraft behavior and why it behaves that way and both will give a different specification. This is known as the human world view [115]. These variations are caused by the differences in cultural, educational and cognitive background and more importantly by the context of interest in reality of the observer. An observer's awareness and appreciation of objects, processes or situations in his environment as mediated through his sensory organs is a highly intuitive and iterative process, and is guided by the objective the observer is after. The interpretation and specification of such observations always reflect the individual thought process and the applied rational principles as shaped by his background knowledge. Therefore there is always a natural indirection and subjectivity in specifying reality. Real-world knowledge elicitation and specification is thus an capricious process and doesn't have to result in single unique knowledge specification of the real-world or simuland. This also holds true for the knowledge perceived, interpreted and specified from the real-world as reenacted by the simulation system. Remember that the simulation system representation of the simuland as experienced by the simulation developer and user is also part of material reality.

These notions have several important consequences for the practical measurement of simulation fidelity. First of all, since it isn't possible to directly and objectively define reality in pure sense, the essence of practical fidelity measurement thus involves a comparative analysis of knowledge specifications developed for the simuland experienced in both reality and the simulation system. This immediately proves that *esoteric fidelity* is something that can never be measured in real-life practice.

From the esoteric perspective the measured fidelity of simulation should always be the same, even when used in a different context and evaluated by different people. In theory this is true simply because neither the simulation nor reality changes. However, the essence of practical fidelity measurement involves knowledge specifications for reality instead of reality itself. For reasons discussed in the beginning of this section the exact content of such knowledge specifications is not necessarily unique. This may result in inconsistent fidelity ratings for the same simulation, since the measurements are based on different knowledge specifications for reality, will differ as well. The only way to overcome this problem is to develop an universal and authoritative knowledge specification for reality by observing and experimenting with the simuland in all possible contexts and taking into account all possible users backgrounds during its formal specification. Obviously, this is never attainable in real-life.

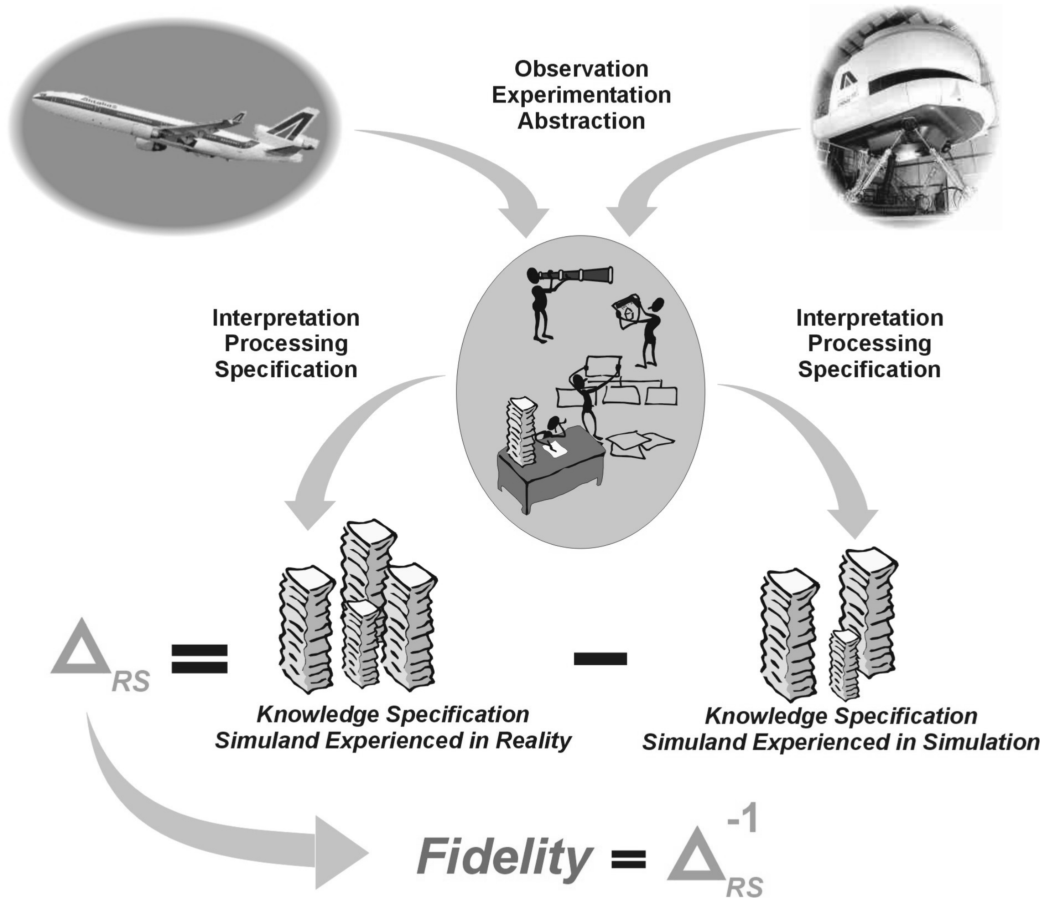


Figure 4-2 Essence of practical fidelity visualized for a MD11 flight simulator

Any attempt to develop an ultimate and universal reality description capable of serving the whole simulation community as part of a fidelity theory development will fail in practice, since their will always be someone or group for which it will not suite their context or needs [126]. What is needed is a separation between what real-world knowledge should be included in a real-world specification and the fidelity theory itself. Instead a fidelity theory must provide the means to collect, structure, formally specify and evaluate the suitability of this real-world information as part of the whole simulation fidelity assessment. Even though the problem/application domain-specific real-world knowledge to be used varies, such a separation will make a fidelity theory and associated concepts universally applicable across different domains. More importantly, this approach will explicitly express the limiting effects of reality experience and specification on the consistency, interoperability and (re)use of practical fidelity specifications.

4.2.3 Inherent Limitation of Fidelity Measurement: Error and Uncertainty of Real-World Data

Fidelity is considered and anxiously searched for by many as the exact measure of realism. From the esoteric perspective, fidelity is indeed per definition the exact measure of realism since this involves the direct comparison between reality and simulated reality. However, as argued in the previous section fidelity measurement in real-life simulation practice is based on a comparison of knowledge specifications developed from reality (*Figure 4-2*). Therefore searching for the exact degree of realism is a goal, which can never be accomplished, because it is practically impossible to know everything about reality due to our limitations in observing and measuring reality, interpreting the obtained real-world knowledge and explaining how reality works based on this information [124]. Similarly, Harmon formulates it as: *"complete consistency with all reality and other true information means that the only one who can test the truthfulness of any single piece of information would need ready access to all reality and true knowledge"* [54]. Since the unavailability of this in real simulation practice, it is only possible to verify the truthfulness of real-world information within the limits of the knowledge of that reality and truth. Due to this there will always be an error and uncertainty in our perception and formal specification of real-world knowledge. *Uncertainty* in this regard is characterized as incomplete information and the lack of information, which limits the exact correctness with which any kind of real-world knowledge can be known and is thus a cause of potential deficiencies [95] [108] [141]. In practice these errors and uncertainties originate from both the observation process (measurement precision and conditions, system availability, data completeness and adequacy etc.) itself and the process of developing knowledge specifications (interpretations, assumptions, decisions etc.) from these observations.
It should be mentioned here that this kind of uncertainty must not be confused with system *variability*. Variability is defined as the inherent variation associated with the system under consideration [96]. Examples of variability include stochastic processes such atmospheric conditions or system behavior due to manufacturing variations, which are usually modeled by means of probability distributions. To discriminate between these two types of uncertainty Oberkampf calls the first kind *epistemic uncertainty* and the second *aleatory uncertainty* [97] [98]. Obviously, epistemic uncertainty is the hardest one to deal with during fidelity assessment in terms of qualifying or quantifying its magnitude when possible. In this thesis the term uncertainty is equivalent to *epistemic uncertainty* unless stated otherwise.

The comparison between actual gathered data from material reality and the simulated counter-part is often advocated as the 'only' way to measure fidelity. Usually such proponents express this idea in relationship to research simulations of imaginary reality and those situations of material reality for which no empirical data is readily available or rare. These are situations that imply considerable large uncertainties and thus possibly large deviations from reality. Conceptually comparisons based upon actual gathered data might be the best method for fidelity measurement, certainly from the esoteric fidelity perspective. It is beyond dispute that fidelity measurement of simulation systems for which a rich set of data is available from material reality is more likely to provide more accurate measurements with lower uncertainty than for those with less or no actual empirical real-world data. However, one must realize that the measured empirical data from both material reality itself and the data elicited from the simulation are also in the same fundamental way subjected to uncertainties and errors as discussed above. For instance, empirical real-world and simulation data pairs are just samples of reality, which by themselves already create error and variability. Even when statistical methods, such as proposed in [2] [22] [66] [129] [142], can be used to compare these samples they still provide a probabilistic answer and not an exact answer. Furthermore, such comparisons only demonstrate how well the simulation is capable of replicating this known empirical data set and doesn't provide any guarantees on simulation correctness with respect to material reality. Especially for those cases when inferences must be made about a system replication of reality outside (interpolation) or beyond (extrapolation) the known data set grid and range. This notion raises another issue. In order to specify the exact degree of realism it is necessary to collect empirical data for every conceivable aspect, condition and scenario of material reality, which will result in an infinite real-world and simulation knowledge specification and their associated comparative tests to be performed. Obviously, this is something that is impossible in practical simulation development.

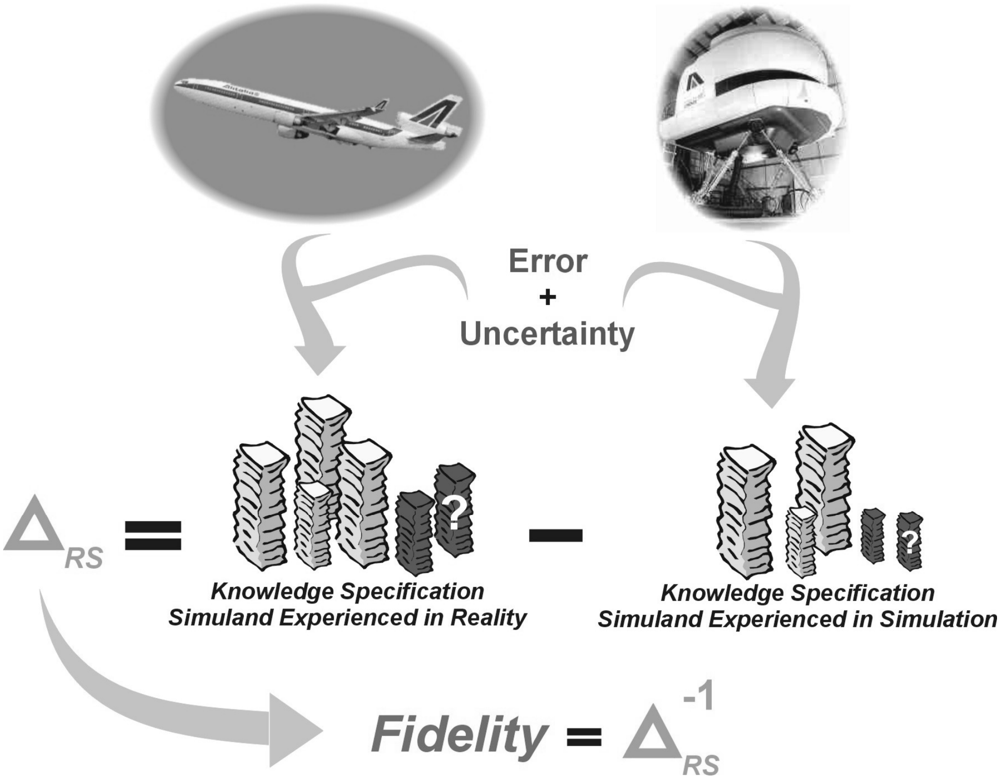


Figure 4-3 Limitations of practical fidelity measurement visualized for a flight simulator

It must thus be concluded that no form of fidelity measurement can ever conclusively establish the exact degree of realism of a simulation (*Figure 4-3*). None of this, however, should lead to the misconception that fidelity measurement of a simulation is useless or a waste of time. On the contrary, having some sort of a quantification or qualification of uncertainty in your available capabilities of measuring and estimating the simulation realism is a most important and powerful fidelity metric in judging the validity of a model or simulation. Knowledge of this uncertainty implies increased confidence in the validity of a simulation. As an example knowing that you are 90% or 40% sure that the error of a simulation model parameter has a certain value or is within a certain range, is equally important as knowing the error level itself. Remember that the purpose of validation is to establish the confidence that the simulation is reflecting reality, material or imaginary, to a certain extend, which makes it suitable for the desired user purpose i.e. having enough credibility. Here fidelity theory is the tool to specify the correspondence between reality and the simulation, including the expression of the correctness and uncertainties with respect to this specification. Therefore, even when it is not possible to know or quantify all aspects of reality with a hundred percent certainty, it is far better to specify the degree of realism as good as possible for any situation in a formal structured manner than specifying nothing for this reason.

The essence is to accept and express the uncertainty in any attempt to assess simulation fidelity in practice. By doing so, one becomes aware of its implications and then it's possible to transform the fidelity theory into a pragmatic tool that helps in assessing the level of confidence in the correctness of the simulation outcomes with respect to reality. This objective can only be achieved by means of a finite suite of tests comparing the simulation against the best quality knowledge specification available for both the simuland and its simulated representation.

4.2.4 Fidelity and the Modeling & Simulation Enterprise: Realistic Expectations

Modeling and simulations is used for many different purposes ranging from training, simulation based acquisition, test and evaluation to research, development and engineering. Fidelity is often seen by many as the panacea for all kinds of problems in the development and validation process of such simulations (see Section 3.3). Which usually implies that fidelity measurement by itself is often expected to immediately provide the answers to what level of fidelity is required for a certain purpose, how to realize this and afterwards automatically decides whether the developed simulation is indeed valid for this purpose. As already discussed in Section 2.5 many fidelity measurement approaches developed in the past therefore have been linked to specific contextual elements, which try to provide a non-reusable cure-all solution for a single problem or application at hand. Given the strongest formulation for fidelity (Section 4.2.1) and the fact that every simulation application has its own unique context and objectives (Section 4.2.2), it is posed here that no formal fidelity theory by itself can ever provide the answers to these above mentioned issues. It is simply not possible for a fidelity theory to anticipate to this infinite set of possibilities. Furthermore, measured fidelity is per definition an absolute measure for simulation realism and not some sort of a measure relative to the simulation purpose (Sections 2.5.4 and 4.2.1).

Neither, will a fidelity theory be able to provide new knowledge about reality not already encapsulated in the simulation model as often is expected from those who use simulations to numerically explore and validate new scientific system theories (generalizations and laws of nature for phenomenon etc.) in the form of a mathematical model. Hence material reality is the only reference source for inductive development and validation of a new theory with or without the usage of simulations. Even though simulation and theory validation have striking similarities, validation of a new theory or mathematical model to describe and understand phenomenon of material reality is thus not equal to simulation validation. System identification is an example of developing such new theories from empirical data sets, a set of candidate mathematical models and utilizing simulation based assessment to determine the best model in the set based on a chosen criterion of fit [80].

4.2.5 Selecting Hermeneutics as the Basis for Fidelity Theory and Practice

In the previous paragraphs it is argued that it is not possible to measure or specify the level of simulation fidelity to a full-extent and in some exact manner and with absolute certainty. Nor does it provide the so desired solution for scientific deductive reasoning issues. Failure to see this or not accepting it, will result in the continuation of the endless search to fidelity as being a mythical holy grail without practical usage. Therefore the hermeneutical perspective is adopted as the basis for fidelity theory and practice developed in this thesis. Hermeneutics is a concept originating from the philosophy of science, which can be related to simulation validation [37] [66] [68]. From this perspective simulation validation is seen as a court system in which the prosecutor has to legally and convincingly prove that the defendant is guilty of the committed crime. The crime in this context is the statement whether the level of simulation fidelity is good enough for the simulation purpose expressed in terms of fidelity requirements (Chapter 8). Convincingly in this regard yields developing and specifying the best credible fidelity evidence possible to prove with an acceptable level of certainty i.e. beyond a reasonable doubt that the level of simulation fidelity is good enough (Chapters 5 and 7). Finally, legally in a validation context yields following a well-defined systematic process with a set of traceable, repeatable coherent rules and methods to obtain this fidelity evidence (Chapters 8 and 9). Such obtained fidelity evidence will also assist in tracing the source of simulation fidelity problems and when possible in developing pragmatic solutions for it during simulation development.

A fidelity theory and practice should thus provide the tools to properly assist the simulation developer and VV&A agent in addressing these issues. Based on how simulation fidelity can be formally characterized, measured and specified, it will also be possible to better specify fidelity requirements. From that specification various application and problem domains fidelity requirement assessment strategies and standards can be empirically developed. Since fidelity theory and practice is a tool it can never on its own decide on or chose fitness for purpose criteria. However the inhere proposed approach to fidelity theory and practice will help to turn the concept of fidelity into a beneficial and general applicable tool for providing the fidelity evidence necessary to make well-considered design and validity judgments and decisions.

4.3 Simulation Fidelity Theory: Fundamental Concepts

The previous section introduced the strongest formulation for fidelity, the motivation and the essential elements underlying a pragmatic and sensible theory for fidelity. In this section these elements are translated into a set of fundamental concepts, which together provide the definition and foundation of a pragmatic simulation fidelity theory. Section 4.3.1 presents, in an informal manner, the cornerstone of this pragmatic fidelity theory and measurement, the authoritative real-world knowledge standard paradigm or fidelity referent. This paradigm provides the bases for a more pragmatic definition of the term fidelity along with the definition of a set of principle fidelity theorems (Section 4.3.2). Finally, the basic fidelity characterization concepts are presented that serve as the bases for the development of fidelity qualification and quantification methods (Section 4.3.3). Later on in this thesis these methods and metrics will be discussed more rigorously.

4.3.1 Fidelity Referent: The Real-World Knowledge Standard Paradigm

In Section 4.2 it was shown that pragmatic measurement of simulation fidelity depends upon the knowledge of reality, since reality can only serve as a knowledge source. Therefore, the availability of a formal specification of real world knowledge for that part of reality the simulation tries to reenact, is the most essential element of fidelity theory and practice. Without such a real-world knowledge description there is no *reference* for specifying the required fidelity or measuring the achieved level of simulation fidelity and to perform other fidelity assessment activities throughout the simulation system development process.

As defined in the previous chapter, system knowledge specification is a description of how a system behaves and of the mechanisms that make the system behave the way it does. This involves the system *behavioral description* that specifies the observable manifestation of a system over-time, and the system s*tructural description* that specifies the inner structure and working of a system. This collection of observed real-world knowledge is used as reference for fidelity measurement and is also a form of system knowledge specification. Since it is impossible to fully specify reality, such specifications will always contain assumptions and approximations (Section 4.2.2 and 4.2.3). It is postulated here that any attempt of observing, explaining and documenting how reality works, either material or imaginary, is thus a form of an abstraction process. Therefore, reality specifications can be considered as an approximated abstraction of reality or *model* in a general sense and is here referred to as the *fidelity referent*. Unlike a simulation model however a fidelity referent is not a generative model, which means it is not *directly* focused on or not *necessary* capable of generating or animating the behavioral data and representational structural properties of its contents.

The fidelity referent paradigm formalizes the natural level of indirection of fidelity measurement i.e. in determining fidelity one never actually measures against reality itself but against an approximated interpretation of reality (see *Figure 4-2*). This is achieved by linking the effects of errors and uncertainties in reality perception to the fidelity referent. In this way it explicitly separates the real-world correspondence in 'exact' sense into a correspondence between reality and the referent and correspondence between the referent and a model or simulation. Fidelity measurement now solely specifies the correspondence between this referent knowledge and the knowledge obtained from model or simulation. The actual correctness of the measured fidelity level in 'exact' sense is determined by the quality of the real-world knowledge contained within the fidelity referent and the model and simulation knowledge specification. Since the limitations of available real-world knowledge are tight to the referent its contents thus comprises the best real-world knowledge available i.e evidence to compare the simuland with (*Figure 4-4*). The availability of such real-world knowledge is also limited by other simulation development constraints placed on the fidelity referent development process such as regulatory, time, money, security constraints etc. Based on these premises a *fidelity referent* is now defined as:

> *'A codified, structured, and formal specification of real-world knowledge about what is commonly perceived, understood and accepted by a defined group of people to be the truth or reality, capable of serving as the comparative standard for reality correspondence assessment and associated activities of model or simulation development'*

This definition for a fidelity referent also formalizes the notion that it is practically impossible to create a single universal authoritative reality description that is profound enough to serve the whole simulation community (Section 4.2.3). In practice each application or problem domain, even each specific application or organization, may have its own authoritative fidelity referent, which is an accredited knowledge specification of that part of the real-world they are interested in and forms the comparative standard for their model and simulation types. Despite this it still will provide an acceptable and productive platform for simulation fidelity exploration as long as all real-world knowledge contained in the fidelity referent is complete, consistent and unambiguously specified in a hierarchical and traceable manner. In order to be a productive platform for simulation fidelity measurement such real-world knowledge specifications have to be compatible with how fidelity is characterized and quantified (Section 4.3.3). What is needed are formal and structured languages or documentation templates to specify and communicate the real-world knowledge relative to ourselves, the organization, the application domain, etc. In Chapter 5 a possible instance of such a generic documentation template and formal specification language for a fidelity referent is developed. This proposed fidelity referent specification format is based on the systems engineering approach and contextual framework for fidelity as discussed in the previous chapter.

The exact content of a fidelity referent is not part of a fidelity theory. However, from a high-level view it is possible to identify in general what kind of real-world knowledge classes and sources can be used to populate a fidelity referent structure in order to serve as a comparative standard (*Figure 4-4*). Material reality surrounding us comprises existing phenomena and systems with which it is possible to experiment and to observe. Information and data that result from experiencing, observing and experimenting with material reality form the first of three real-world classes that can be imparted in a fidelity referent. This class of knowledge is here referred to as *empirical simuland knowledge*. Empirical is considered here as any data resulting from observation and experience of material reality. Experimental data is a major subset of this empirical data and results from consciously designed experiments and observations ('in vitro' or 'in vivo'). That means data, which is systematically collected from well-defined test regions in material reality under a set of predefined well-known and well-controlled conditions. However this doesn't necessarily imply that every condition is known, observable or controllable in the test region for which the data is valid (see Section 4.2.3). The other remaining empirical knowledge part results from 'ad-hoc' or arbitrary observations and experience of material reality where no conditions were consciously controlled and other important conditions might not or not-fully known or even not recognized at all. Obviously, this kind of empirical data is more likely to contain significant uncertainties than the first kind. Empirical simuland knowledge specification formats and types can range from tables or graphs with measured time-histories to written eyewitness reports or findings of subject matter expert (SME).

In pure scientific research the empirical knowledge is used to develop appropriate system theories to describe material reality in order to explain and understand why and how material reality works that way and to predict not yet seen phenomenon or system behavior. Such theories are developed through a rational process of inductive reasoning and logical analysis (hypothesis), which is guided by the researcher's abstractions and assumptions about the fundamental nature of reality [38] [68] [141]. These theories lead to what is defined here as *theoretical simuland knowledge* and comprises two classes of real-world knowledge that can be imparted in a fidelity referent (*Figure 4-4*). First, is the system theory itself which includes generalizations, laws of nature, mathematical formulations and models, handbook knowledge and methods, but also SME knowledge and intuition. Secondly, the theory can be used to experiment with to create artificial or theoretical data for phenomena and system behavior by means of analytical and numerical calculations (i.e. simulation). This theoretical data is used to explore, refine or develop new system theories for material reality through a deductive reasoning and logical analysis process. Remember that such refined or new theory can only be proven against empirical data that is outside the control of its developer. Thus one should not equate theory validation to simulation validation. More importantly in all other cases and simulation applications (training, engineering, simulation based acquisition, etc.) this theoretical data can serve as legitimate fidelity referent data and opens the gateway to fidelity assessment of simulation involving imaginary reality elements (*Figure 4-4*).

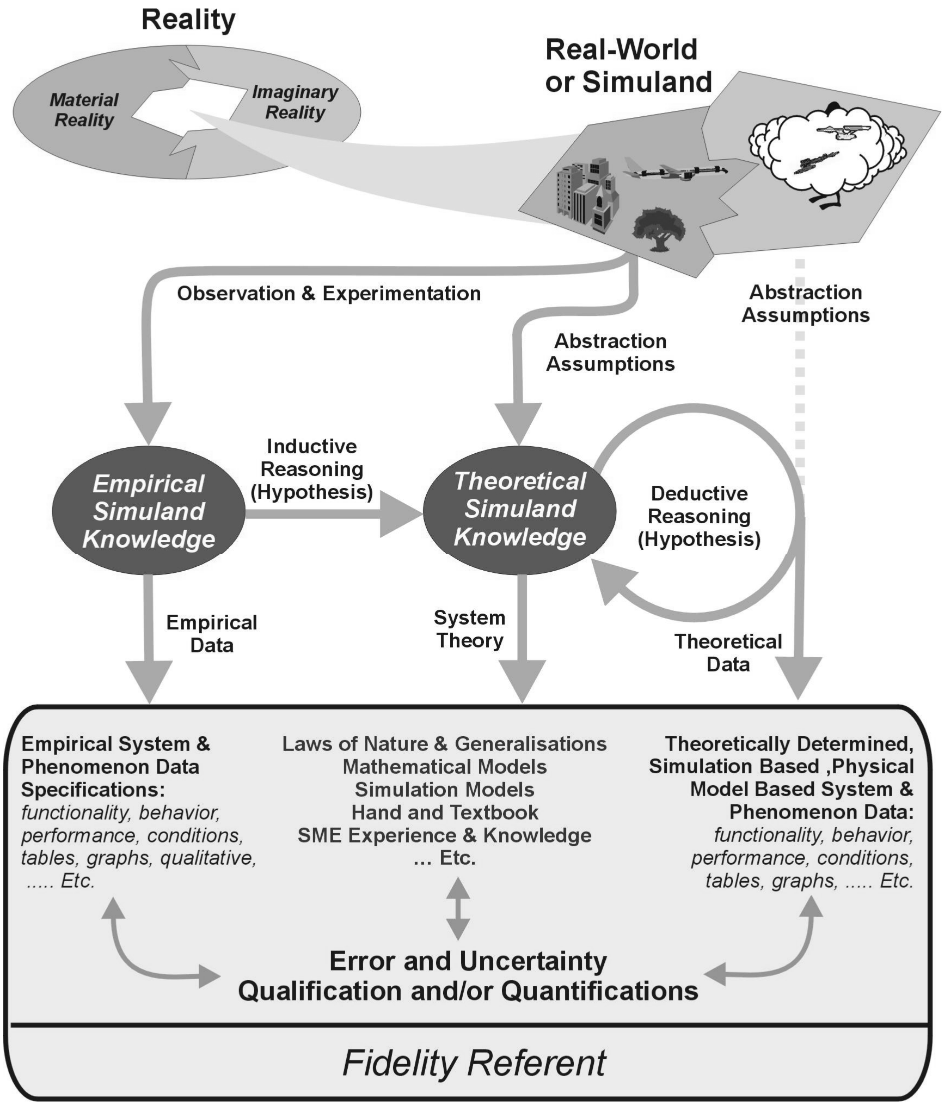


Figure 4-4 Fidelity referent and the real-world relationships

Imaginary reality mostly refers to by man-made systems that are in the human-mind or just on paper of developers and that will possibly become material reality somewhere in the future. In order to be physically realizable in material reality these systems have to comply with the common accepted generalizations, laws of nature, and any other accepted true knowledge about material reality. If not, such systems will be never be physical realizable in which case fidelity assessment is nonsense. Such simulation applications are usually found in the entertainment industry. This means that it is possible to infer or artificially construct (general theory based, mathematical model, physical scale model tests, other similar systems and their simulations, manufactures specifications) comparative reference knowledge as the best placeholder for non-existent system data in order to assess the fidelity of a simulation. This type of fidelity referent data might have larger errors and uncertainties compared to empirical simuland data. The major difference here is that it is determined and accepted to be the best available knowledge i.e. evidence for the simuland at the given moment and conditions until new and better knowledge for the simuland becomes available or proves otherwise. In order to make this data suitability judgment and for other simulation validation purposes a fidelity referent should impart handles, qualitative and/or quantitative, to assess the correctness and uncertainties of all real-world knowledge contained in its structure (*Figure 4-4*).

Fidelity referent development could either start from scratch, by adapting existing or by composing it from parts of other referents when no suitable referent for the current application is already available or dictated by a certain domain. This means that an already developed fidelity referent itself might serve as a source of existing real-world knowledge that can be reused for other simulation development and fidelity assessment activities. Since such referent knowledge is domain dependent, it might not directly suite a different domain or context. This requires that fidelity referent knowledge needs to be traceable to its original used knowledge sources to enable proper reuse.

The real-world knowledge as specified by the fidelity referent is something that evolves when more data or insights are gained about how reality works. These changes may result from different causes and stages in a fidelity referent's life-cycle. Obviously, more knowledge comes available when imaginary reality aspects become existent in material reality. For instance in development of new aircraft, simulations are already used for making preliminary design decisions. In that stage not much knowledge is available except for handbook knowledge and data from similar but already existing aircraft. When wind tunnel tests, structural model test or aircraft flight tests are performed more and more experimental data comes available that can be used to populate and improve the fidelity referent content. Such experimental information is mandatory for fidelity assessment of pilot training simulators to be developed for that aircraft [117]. Other causes include simply discovering already available knowledge sources and knowledge that comes into range due to changes in certain constraints such as improved experimental facilities, additional funding, security clearance removal etc. Especially, for accredited fidelity referents that are commonly (re)used within a certain organization, application or problem domain, this implies that the referent content needs to be reconsidered and revised from time to time. Therefore fidelity referent development must be well managed and maintained (see Chapter 9).

Fidelity theory can never solve the limitations of real-world knowledge error and uncertainties (Section 4.2.3). However, by associating these errors and uncertainties to the fidelity referent, qualitatively or when possible quantitatively estimates, it provides a pragmatic and formal approach to properly deal with these limitations in simulation development and validation. First of all it does more explicitly state that the exactness of the measured degree of simulation fidelity with respect to reality (i.e. approximation of esoteric fidelity) heavily depends on what real-world knowledge is used and the quality of that knowledge rather than on the assessment methods employed and their produced values. Stronger formulated one can say that the 'exact' level of measured fidelity is as good as the quality of fidelity referent. Secondly, the fidelity referent approach makes it possible to specify, analyze and compare the fidelity level of any model or simulation in a consistent and uniform manner, regardless whether it represents material reality (existing systems) or imaginary reality (non-existing systems) or combinations thereof. Obviously, for data-rich environments and under the proper conditions the error and uncertainties of the real-world knowledge contained within the fidelity referent might be smaller than for data-poor environments. In this way fidelity can be used as a beneficial tool for describing and comparing model and simulation capabilities of any kind.

The level of indirection introduced by using a fidelity referent during simulation development also enhances the whole fidelity assessment process by means of separation of concerns. Completeness, error and uncertainty issues of the referent knowledge are most important in judging the credibility of the simulation fidelity measurements or estimates based on this referent within the context of the simulation purpose. For other fidelity assessment activities it is usually sufficient to know that there is such proper knowledge available or at-least the best knowledge available. Since it is impossible to know everything about reality (see Section 4.2), this is an activity of inherent subjective and intuitive nature seeking for adequate data with acceptable error and uncertainties that suite the simulation purpose as good as possible. Therefore, instead of reducing the error and uncertainty of real-world knowledge at any cost, validation must establish accredited confidence levels for this knowledge, which relate to the importance and risks involved with the application and usage of the simulation outcomes. These notions should underlie all validity judgments and are made more transparent by the fidelity referent approach [124] [126]. Chapter 8 discusses how this can be achieved in terms of *fidelity requirements* and *fidelity-based validation*. Finally a carefully defined fidelity referent structure provides the opportunity to develop automated tools in order to cost-effectively deal with the cognitive complexity of reality and fidelity assessment in the simulation enterprise [120] [122] [126]. Examples of such tools include multi-criteria analysis tools, expert systems, fuzzy logic, neural networks, formal language tools or any other soft computing techniques to support efficient fidelity measurement [46] [125] [141].

4.3.2 Practical Fidelity: The Pragmatic Definition and Theorems for Fidelity

Based on the premise that practical fidelity measurement is performed using a fidelity referent as developed in the previous paragraph it is now possible to provide a more pragmatic definition for fidelity:

'The formal specification of the inverse difference between the referent and the knowledge specification of the simulated real-world'

A compact and unambiguous formulation for the term fidelity that meets the requirements posed in Section 2.5 of being an application context and fitness-for-purpose free formulation, and the normal apprehension that the higher the fidelity the smaller the difference and thus the better the simulation resembles the real world. This definition is a weaker formulation for the term fidelity than the *esoteric fidelity* definition developed in Section 4.2.1. Since this definition takes the essence and limitations of practical fidelity measurement (Sections 4.2.2 and 4.2.3) into account it is a fidelity formulation for which a practical and viable implementation is attainable in the modeling and simulation enterprise. Therefore this definition for fidelity is called *pragmatic fidelity*. The relationship between both fidelity formulations is visualized in *Figure 4-5*.

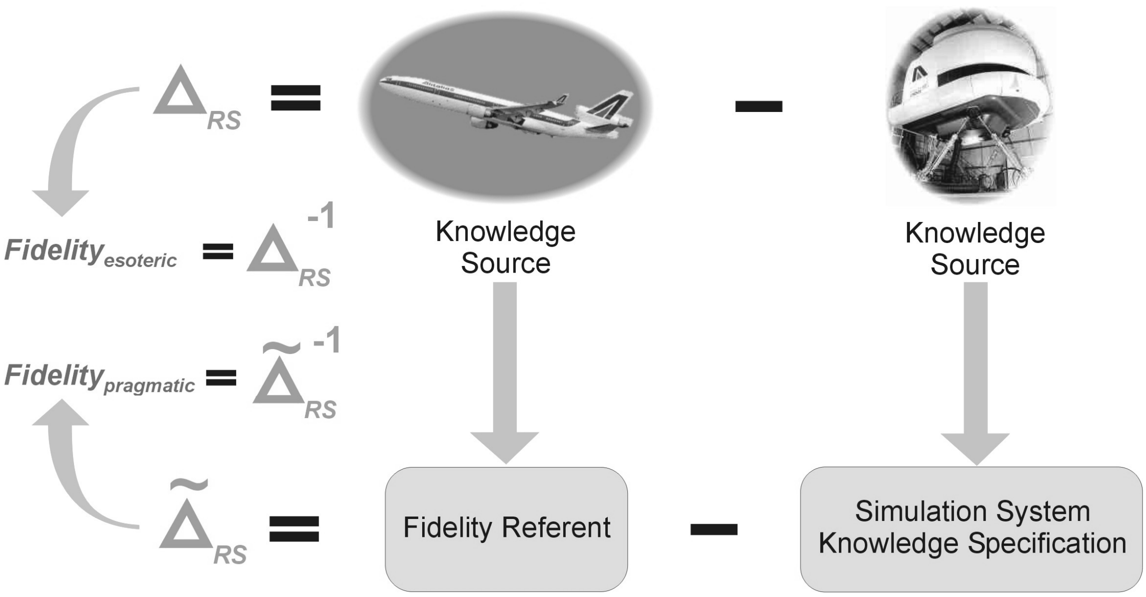


Figure 4-5 Esoteric versus pragmatic fidelity definition

Utilizing the pragmatic definition for fidelity and *Figure 4-5*, *fidelity measurement* can now be more explicitly defined as:

> '*The comparative analysis between the fidelity referent for the real-world and the simulated representation of this real-world in order to specify the inverse difference between both.*'

The pragmatic fidelity formulation is supported by twelve fidelity theorems, which further define the concept of simulation fidelity and its relationship to the modeling and simulation enterprise. These theorems together outline a framework of fundamental principles, propositions, and postulates for the development of pragmatic simulation fidelity assessment methodologies and practices.

Theorem 1

The strongest conceptual most correct formulation of simulation fidelity, 'esoteric fidelity' can never fully be articulated in practice.

The prime reason for this is the inherent limitations in every experience, specification and practical measurement of reality. Secondly, it cannot be attained due to practical constraints placed upon every simulation development and validation process that limit thoroughness and extent with which simulation fidelity can be assessed. For instance, time constraints might impose that only those real-world aspects can be evaluated which are of most critical for the application at hand. Therefore in practice fidelity is qualified and quantified with respect to a *fidelity referent* instead of reality itself.

Theorem 2
Fidelity qualification and quantification of any form always has an inherent level of uncertainty, which limits the exactness with which the level of fidelity can be specified.
The uncertainty in fidelity specification is largely introduced by the limitations in the available knowledge of the real-world. Since the simulation system and its produced replication of the real-world are part of material reality, knowledge specifications developed from this source is also subjected to the same limitations and therefore introduces uncertainties as well. However, the difference is that simulation system representations of the simuland are usually better controllable and observable, and their internal constitution and working is better known than the real-world. In practice the extent and exactness of the measured fidelity thus depends on the availability, adequacy, correctness and uncertainty of the knowledge used in the measurement. Therefore the extent and exactness increases with increasing knowledge about the real-world and its replication by the simulation system, and the quality and appropriateness of that knowledge. Furthermore, remember that the specification process itself is also error prone and therefore the specification process itself can introduce uncertainties. Such uncertainties will decrease with the thoroughness of the process with which the fidelity is specified. Therefore it is important to prevent such extraneous errors and uncertainties as good and as much as possible in the fidelity assessment process.

Theorem 3
Fidelity is an intrinsic or absolute property of any model or simulation characterizing its degree of realism.
This means that fidelity is solely a specification for how much the modeled or simulated real-world differs from the actual real-world. The rationale for this is the fact that the fidelity of a model or simulation always remains the same when (re)used as-is in different application contexts, simply because the simulation doesn't change and therefore its representation of reality doesn't change. Obviously this is per-definition true for esoteric fidelity, but also from a pragmatic fidelity perspective even though the fidelity measured and specified in practice, may vary due to different world views and available referent data.

Theorem 4
Fidelity of a model or simulation is qualified and quantified by an enumeration of various multidimensional and multifaceted metrics for all model or simulation aspects that characterize its degree of realism with respect to the real-world.
Fidelity in this regard must be seen as an umbrella concept that combines all conceivable descriptors, metrics and methods necessary to properly specify the degree of realism of a certain model or simulation. The great complexity of both reality and simulation systems require a far larger information content than can be offered by a singular overall fidelity quantification or qualification, in order to make fidelity assessment a useful element within the modeling and simulation enterprise.

This can be illustrated by means of simplified example in *Figure 4-6*: here it is assumed that capabilities of three simulations (*A*, *B* and *C*) of the same aircraft can be visualized by means of a spider chart. The axes in each chart represent for each simulation the simulated aspects of the real aircraft such as for instance velocity or acceleration. A point on each of the axes in *Figure 4-6* quantifies the difference between the real aircraft aspect and the simulated aspect utilizing a certain metric. For example, in this case it could be the maximum norm of the measured difference in the real and simulated aircraft speed. This means the closer the point is to the origin the closer the simulated aspect corresponds to the real aircraft value.

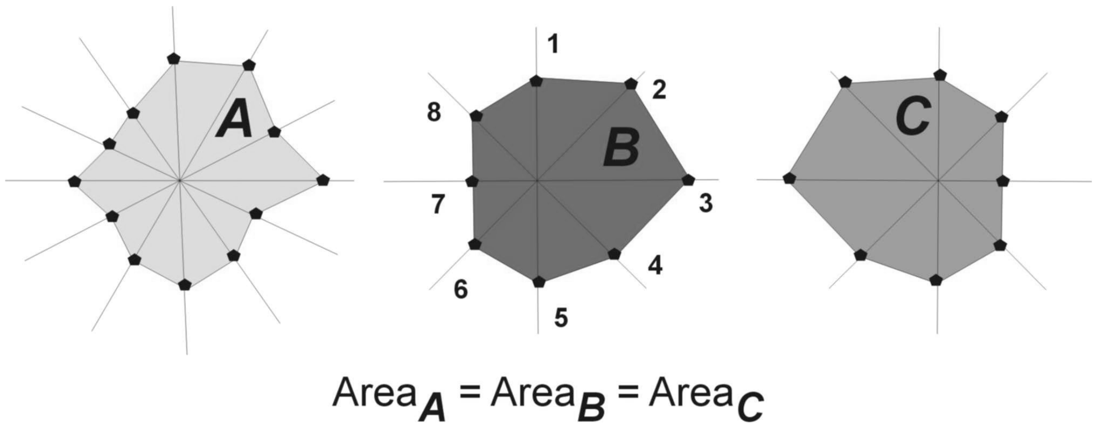


Figure 4-6 Multiple versus Singular Value Fidelity Specification

A fidelity specification in terms of a singular overall metric for these three aircraft simulations could be the inverse of the surface encapsulated by the connective lines of each point in the above spider chart i.e. $1/Surface_{SimA}$ etc. Since all three surfaces are equal in area this suggests that the fidelity of each simulation would be identical. Evidently this not true. Compared to simulation *B* and *C*, simulation *A* reenacts more aspects of reality. Furthermore, although simulations *B* and *C* may reenact similar aspects of reality the quantified differences differ significantly. A multidimensional and multifaceted approach to fidelity measurement for this example could yield, but is not limited to, the enumeration of the reenacted aspects of the real aircraft in combination with the set of belonging measured maximum norm values. This type of fidelity characterization would immediately reveal these differences in realism. Therefore, adoption of a singular overall metric approach for fidelity has no meaningful utility in the modeling and simulation enterprise since it obscures essential characteristics, which easily leads to false conclusions or decisions. Due to this a composite singular metric is also often harder to be interpreted. An umbrella approach to fidelity characterization does provide the information content capabilities necessary to properly support the decisions to be made throughout a simulation development and validation process.

Theorem 5

Fidelity assessment has an inherent subjective and qualitative element.

Fidelity specification in full objective and quantitative terms is almost impossible in practice, since the real-world knowledge acquisition and specification always involves an abstraction process. Every abstraction process inherently contains a subjective and qualitative element. Furthermore, if quantitative referent data is unavailable or the uncertainty of the referent data is not explicitly quantifiable, one can only rely on qualitative evaluations and subject matter experts (SME) opinions as the best evidence available. Individuals in general have different subjective impressions of the same objective fact or event and may decide to initiate different actions. This subjective element of fidelity assessment is specifically eminent in the experience, cognition and interaction with simulated reality by human subjects taking part in the simulation exercise and in reality itself but also in the area of human behavior representations. Qualitative and subjective fidelity elements are thus inherently parts of characterizing simulation fidelity. For more technical utility the usage of objective and quantitative should be favored over subjective and qualitative elements where possible. When subjective and qualitative specifications have to be used they should be used with caution and in a structured rigorous manner to reduce bias due to self-interest, misinterpretation, level of expertise etc.

Theorem 6
Fidelity quantification and qualification doesn't equate to suitability or validity of a simulation.
This theorem is the consequence of theorem 3, which states that fidelity is an absolute measure of realism. Simulation system suitability and validity are all *relative* judgments with respect to the users objectives and application purpose requirements. *Validity* means that it credibly has been demonstrated that the specified available level of simulation fidelity is good enough for the specific application purpose of the simulation system. This judgment not only takes the errors and uncertainties in the specified level of fidelity into account but also the thoroughness of the validation process itself and risks involved in using the simulation systems outcomes for its intended purpose. The simulation system *suitability* is determined by both the validity plus any other non real-world aspects such as: development time, money, resource, simulation system size, simulation execution management functionality and those of other sub-system (i.e. training systems, experimental control stations etc.). Therefore fidelity specification and metrics must not be mixed with user objective, problem and application dependent factors. Instead such *relative* judgments that have to be made throughout the simulation life cycle should be based on a separate set of evaluation metrics and methods utilizing or incorporating these absolute fidelity specifications. To support and to differentiate between these various usages of fidelity specifications in the simulation life cycle, fidelity adjectives such as *required*, *available, differential* and *achievable fidelity* could be used.

Theorem 7
Model fidelity and simulation fidelity do not equate.
As defined in Chapter 3 on the definition and relationships of *model* and *simulation*, simulation is the execution of a simulation model within the simulation system, which results in the replication or animation of the simuland over time. It is this simuland replication produced by the simulation system execution when compared to the simuland itself that specifies the fidelity. Even though the conceptual and simulation model are important and necessary vehicles to create this simulation (*Figure 3-3 and Figure 3-5*), it is this resulting simuland replication that is used in training, engineering design or any other application purpose. Therefore, it is the fidelity of the simuland replication resulting from simulation that should be known to be able to assess the validity of the simulation results. This is here designated as *simulation fidelity*. The simuland described by a model comes alive when implemented and executed in a simulation system. In this regard a model is considered to be a static representation of the simuland. Comparison of a model with the simuland is necessarily a static analysis. The result of this comparative analysis is designated as *model fidelity* and specifies the simulation system's structural capabilities of replicating the simuland behavior. In case of descriptive and computer

models, model fidelity focuses on the mathematical abstractions, assumption and algorithm errors of the simuland representation. In case of physical models, model fidelity accounts for the differences in functionality, construction, geometry etc. Besides the model fidelity, the actual replication of the simuland by a simulation system execution is also determined by scenario/configuration settings made prior to the simulation execution and other (un)foreseen realism disturbing factors (environment noise, heat, etc.) outside the developers or users control that occur during simulation execution. Therefore simulation fidelity specification comprises the assessment of all these above mentioned elements and adds to it the comparative analysis of the resulting simulated simuland representation and behavior. Assessment of model fidelity in here will provide additional evidence for inferring about the level of representation and behavior realism beyond the available reference knowledge.

Theorem 8

Proper fidelity measurement requires that the knowledge specification of the simuland as replicated by the simulation system is defined in terms similar as the fidelity referent.
Rationale for this theorem is that fidelity in practice is the difference between the fidelity referent and the knowledge specification of the simuland as replicated by the simulation system. Obviously, a sensible and practicable comparative analysis requires both knowledge specifications to be a consistent and comparable pair. The consequences for the fidelity referent and simulated simuland knowledge specification are twofold. First, both knowledge specifications have to be considered and developed from the same contextual and experimental frame (i.e. conditions under which knowledge is elicited). Secondly, the knowledge should be specified in the same format (i.e. table vs. table, graph vs. graph etc.). In case when the knowledge specification or parts of it do not fulfill these rules there is a significant change that one makes a less useful or completely useless comparison. This judgment depends on the degree of misalignment between both knowledge specifications, because error and uncertainty of the comparison result grows with an increasing misalignment. Since the fidelity referent is the best-available knowledge for the simuland it inherently defines both the boundaries on what can be compared i.e. test region and the yardsticks for making these comparisons. Therefore, the fidelity referent is the best driver to elicit, structure and specify knowledge of the simuland replication within the simulation system in order to obtain consistent and comparable knowledge pairs.

Theorem 9

Comparison of the fidelity levels of models or simulations for the same real-world counterpart is most useful when the fidelity specification of each model or simulation is based on the same set of fidelity metrics and measured against the same fidelity referent.
This theorem is the logical result of the fact that in real-life there is no single unique fidelity referent for describing the real-world. Since the fidelity referent structure and its contents determines how and what aspects of fidelity can be measured (*theorem 8*), also the fidelity metrics that can be used for this measurement vary with the fidelity referent. Therefore, directly comparing fidelity specifications based on dissimilar fidelity referents is not straightforward and its outcomes are hard to interpreted. In practice this would require the construction of some sort of a equivalence mapping between both specifications. However, constructing such mapping can be a complex and error prone task. This is a task, which can only go from a more detailed fidelity specification to a less detailed more abstract or qualitative specification. Comparison of fidelity levels of

existing simulations is therefore best performed against a single fidelity referent using the same fidelity metrics, which is a consistent and correlated composition of the different specifications and metrics.

Theorem 10

The shorthand qualitative label 'simulation X in its entirety has a higher, similar or lower fidelity than simulation Y' can only be assigned when simulation X for all used metrics to characterize their level of fidelity with respect to the same fidelity referent shows a closer, equal or lesser resemblance of this referent than simulation Y.

In all other cases when comparing simulations, one can only speak in the terms of higher or lower fidelity in direct relationship to a single or a certain subset of the complete set of aspects and metrics that characterize the fidelity of each simulation. This is the immediate result of the combination of *theorem 4* and *theorem 9*, which state that fidelity is specified by an enumeration of metrics and fidelity of simulations is best compared when all are measured against the same referent using the same metrics. To illustrate this theorem, consider the following simplified example where it is assumed that the fidelity for two pairs of simulations ((*A*,*B*) & (*C*,*D*)) for two types of aircraft can be specified by means of a spider chart (*Figure 4-7*). Each axes in these spider charts represent the simulated aspects of the real aircraft and a point on each axes quantifies the difference between the real aircraft aspect and the simulated aspect utilizing a certain metric that in this case returns an absolute value for this measured difference. This means the closer the point is to the origin the closer the simulated aspect corresponds to the real aircraft value.

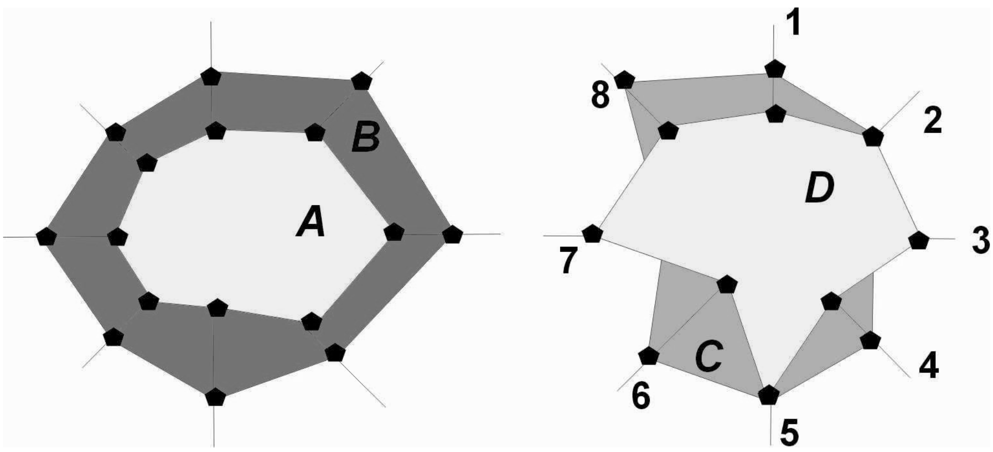


Figure 4-7 Fidelity Comparison of Different Simulations

As can be seen in the left spider chart simulation *A* compared to simulation *B* represents equal aspects of the same real aircraft. Simulation *A* has for every aspect a smaller measured difference with respect to all real aircraft values specified in the fidelity referent than Simulation *B*. In this case it thus is legitimate to say that simulation *A* has higher fidelity than simulation *B*. Even though the fidelity of simulations *C* and *D* is dissimilar it cannot be said that the fidelity of simulation *D* in its entirety is higher than the other. In this case one can only say that simulation *D* compared to simulation *C* provides higher fidelity for the represented aspects *3* and 7 of the same real aircraft, and lower fidelity for aspects *1, 4, 6* and *8*. The technical utility of the concept of fidelity in simulation development and validation is however not affected by this observation. The opposite is true since it is indicated which parts of the simulation representation of the real-world have better or worse fidelity compared to the other simulation. This provides tremendous technical utility for development trade-off and validation decisions when selections have to be made between various models and simulations that have to fulfill the same given user requirements and objectives.

Theorem 11
Simulation fidelity assessment is not an ad-hoc analysis that is only performed after the simulation system has been developed and executed.
Instead it is a continuous process whose activities should be carefully managed throughout the whole simulation system life cycle. The rationale for this is that fidelity is one of the major aspects and cost drivers of any simulation development. The usual ad-hoc approach has several disadvantages that diminish the technical utility of fidelity in simulation development. First, if the level of fidelity of the simulation system does not meet the simulation objectives, it would require more costly redesigns and modifications than when such fidelity problems would have been detected earlier in the development process. Secondly, such approaches implicitly support the common accepted idea the more fidelity the better, which most often will result in unnecessary expensive simulation systems (Section 2.5). Third and most important, such ad-hoc approaches are less likely to provide sufficient and convincing evidence to demonstrate simulation credibility with enough certainty, necessary to make reliable decisions based on the simulation results, and do not support simulation system reuse and interoperability. Therefore, it is essential that fidelity assessment is conducted throughout the whole simulation life cycle. This process of monitoring and controlling the specification of all fidelity characterizations and qualifications or quantifications, its usages and of transforming fidelity characteristics from one stage to the next in the simulation development and VV&A process is called *Fidelity Management*. The thoroughness of such simulation fidelity assessment increases with proper planning, the time and resources spent on fidelity management in this process. The required thoroughness of simulation fidelity assessment depends on the simulation application importance and risks involved with using the simulation execution results by its users.

Theorem 12
In the assessment of simulation system fidelity there is a difference between metrics that truly qualify or quantify the level of simulation fidelity against the referent and those of the implementation and solution space, which do not directly relate to the fidelity referent.
Within simulation systems there are many sub-systems, both hardware and software, whose characteristics do contribute to or influence the replication of the simuland. However, these sub-systems characteristics are not part of or do not directly relate to the simuland itself. As stated in *theorem 7* the simuland replication produced by the simulation system execution in comparison to the fidelity referent specifies the fidelity of a simulation system. Therefore, metrics qualifying or quantifying such non-simuland related sub-system characteristics are per definition not considered to specify or measure the simulation fidelity. Instead they form a separate set of implementation and solution space oriented metrics and methods within the fidelity management process to help characterize or specify the effects of simulation sub-systems on the eventual simulation fidelity as will be available during the execution. This set of metrics is labeled *fidelity performance metrics* and will provide technical utility in determining possible sources

for fidelity disturbances and predicting the effects of development trade-off decisions, such as simulation system component reuse, on the simulation fidelity. As a simplified example consider the *Figure 4-8* at the next page, which presents an aircraft simulation system with a simulation model composed of the two sub-systems *Computer Model* and *Motion System*. Based on an implementation of the aircraft dynamics the computer model calculates the acceleration as would be experienced in the real aircraft flight deck. The difference between this computed and the real value found in the fidelity referent ($\Delta_{RS_1}{}^{-1}$) is a specification of the computer model fidelity. Computer hardware performance characteristics such as CPU speed determine the maximum real-time update rate and thus the minimal integration time step with which the aircraft dynamic model state can be updated. Therefore, computer hardware indirectly affects the error magnitude of the computed flight deck acceleration with respect to the fidelity referent, but the computer system CPU speed variable itself is not part of the fidelity referent.

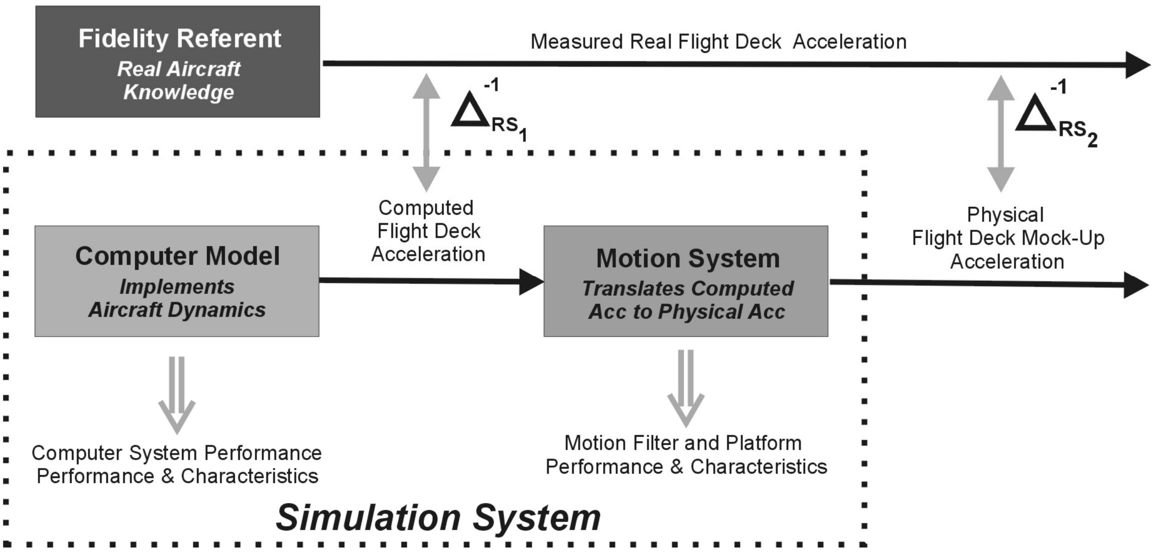


Figure 4-8 Fidelity Metrics versus Sub-Systems Performance and Characteristics

Likewise, the motion system performance characteristics such as acceleration limits, may add additional deviations to the computed flight deck acceleration. Since the motion system is not part of the real aircraft to be simulated, its performance characteristics have no counterpart in the fidelity referent. The resulting physically flight deck acceleration from the motion system does however directly relate to the actual measured flight-deck acceleration as described in the fidelity referent. Therefore, the measured difference $\Delta_{RS_2}{}^{-1}$ (*Figure 4-8*) is one example of a true fidelity quantification for the mimicked physical flight deck acceleration as experienced by the pilot during training and not the motion system performance characteristics themselves.

4.3.3 Fidelity Characterization Concepts: Basis for Fidelity Analysis, Qualification and Quantification

The major underlying principle of the unified fidelity framework is the hierarchical object-oriented system specification approach as presented in Section 3.4. This powerful approach to deal with complex systems can be applied to simuland or simulation system alike. It enables the characterization of simulation fidelity at different levels of abstraction or (de)composition in an uniform and coherent fashion [122] [125]. In the unified framework, fidelity is characterized by eight descriptive concepts: *detail*, *resolution*, *accuracy*, *interaction*, *temporality*, *causality, precision* and *sensitivity* (*Figure 4-9*). These descriptive concepts are the fulfillment of the umbrella approach to fidelity as expressed by the fourth fidelity theorem (Section 4.3.2). Together these concepts outline the basis for the development of fidelity analysis, qualification and quantification methods and metrics that can be used in the simulation system life-cycle.

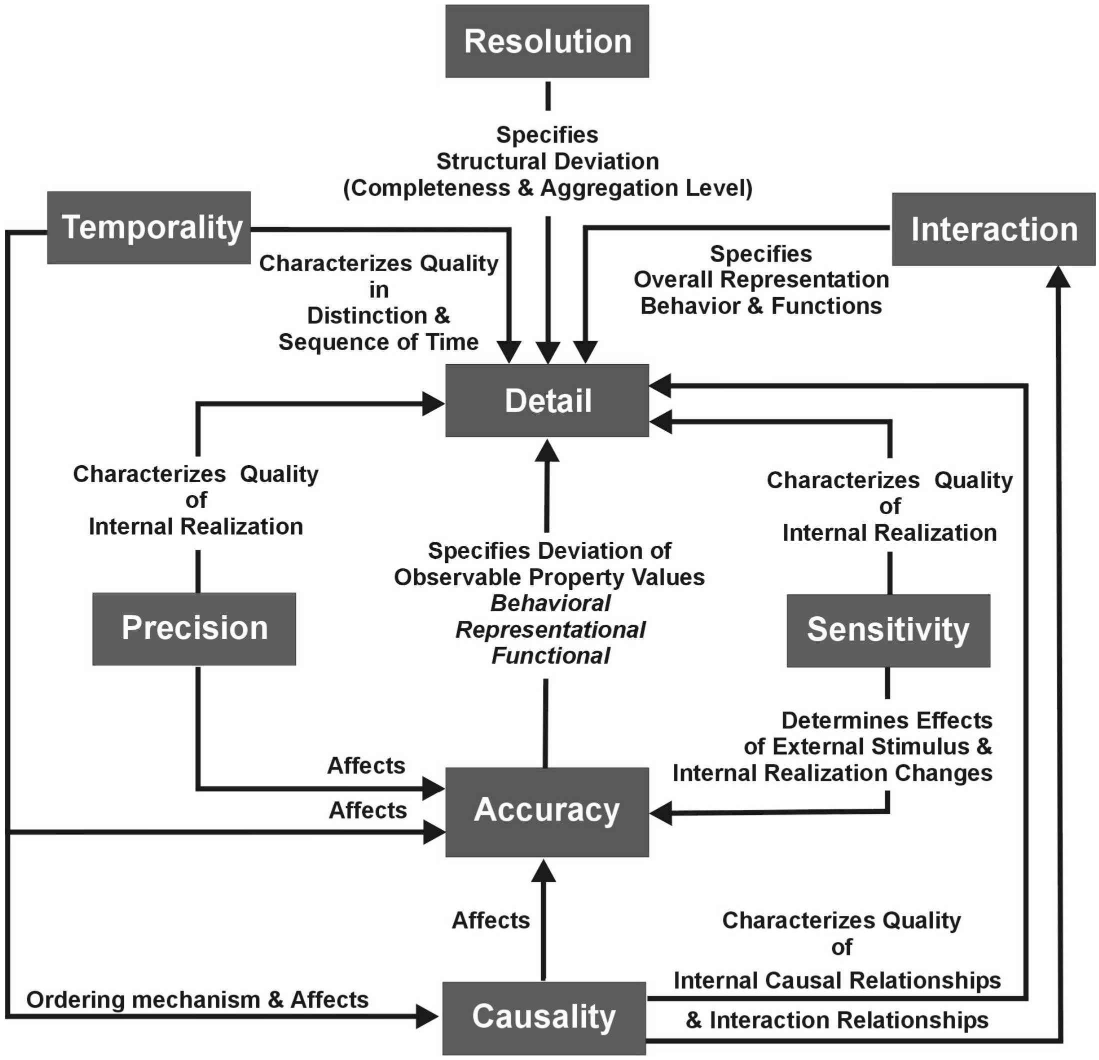


Figure 4-9 Fidelity Characterization Concepts Scheme

Detail: Common Structural Element Denominator

Detail is considered as the major building block for abstracting and specifying aspects of the real world. It resembles a designated part (entity, system, components, object, process, coupling relationships, etc.) of reality with certain identifiable representational and behavioral characteristics and functions. These characteristics and functions are directly or indirectly perceived through a set of observable properties (parameters, state, input and output variables, etc.). Detail is thus a common denominator for all these structural elements of system knowledge specification outlined in Section 3.4, which together describe the inner structural constitution, working and dependencies of a system.

Resolution: The Level of Detail

The level of detail used in a simulation or model with respect to the real-world is what is called here resolution. *Resolution* thus describes what identifiable structural aspects and elements of the real-world are represented by a model or simulation and what aspects have been left out. In short it specifies the structural deviation in terms of model or simulation completeness and abstraction level. Resolution is the roughest form of characterizing the difference between reality and the simulated or modeled representation of this reality.

Accuracy: The Level of Representational, Functional and Behavioral Error

The difference between how the magnitude of the observable characteristics of each simulated or model detail changes over time with respect to its reflecting part in the real-world is referred as detail error. *Accuracy* specifies how close the observable manifestation of each represented system's characteristic representational, functional and behavioral features within a model or simulation resemble those seen in the real-world. Therefore, accuracy is equal to the inverse of detail error, which is in accordance with the common apprehension that high accuracy implies small error. Unlike resolution, the concept of accuracy focuses more on the dynamic elements of the system knowledge specification outlined in Section 3.4. Accuracy is thus a finer form of characterizing the difference between reality and its simulated counterpart.

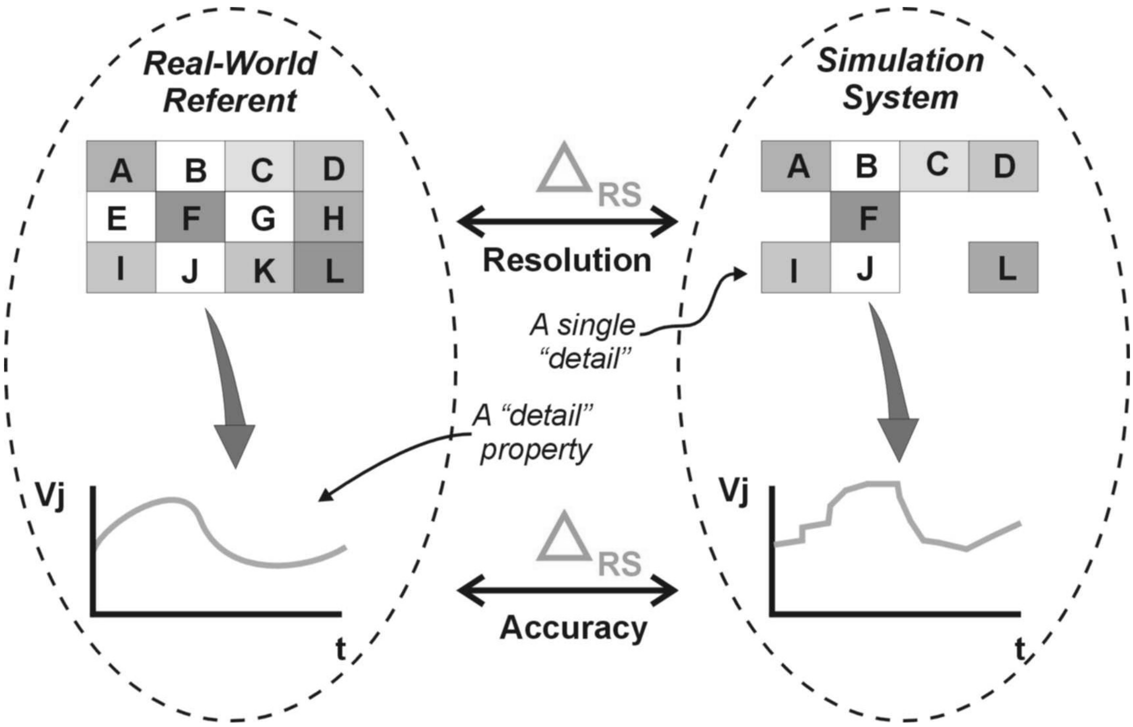


Figure 4-10 Detail, Resolution and Accuracy Concepts Illustration

Interaction: The Overall System Representation, Behavior and Functions

As discussed in Section 3.4 a system can be considered as being composed of a smaller set of coupled subsystems that influence each other over time. The way real-world objects, entities, components, systems, models and simulation systems affect or influence each other is defined here as *interaction*. This interaction describes how lower level (sub)systems collaborate with each other to achieve the emergent representation, behavior and functions of the total system (aggregate). Within the real-world domain, interactions are perceived as the physical phenomenon that form the coupling between the real-world systems. Interaction as seen in the model and simulation world comes in two forms: *technical* and *substantive*. Interaction in technical form refers to the model and simulation components ability of exchanging data with each other via the simulation system infrastructure (memory, network and other I/O devices). Technical interaction physically couples the inputs and outputs of the simulated (sub)systems, which are locally represented within each simulation system component. Proper technical interaction is a necessary but not sufficient condition to assure that each component will interact in a logical meaningful manner and realistic manner. This form of interaction is called substantive. Interaction is a generalized form of the terms technical and substantive interoperability that are used within the distributed simulation community to signify the interaction between networked simulation systems [14].

The interactions between simulation system and model components when integrated, do affect the overall simulation system level of fidelity in terms of overall accuracy and resolution. It is a well-known fact within the simulation community that when all simulation subsystems and model components work correctly and have a good level of fidelity in isolation with respect to the simulation objectives, there is no guarantee that this level is retained when integrated. When the technical integration isn't done properly and/or the level of fidelity of each subsystem or model elements are not fully compatible and consistent, the total simulation system fidelity will deteriorate. Therefore, the emergent fidelity must be evaluated at the total simulation system level, which is characterized by the interactions between its composing model and simulation system elements in conjunction with their local resolution and accuracy levels.

Temporality: The Level of Correspondence to the Notion of Time

Natural sciences use the assumption of the existence of some time-base to describe system dynamics and natural phenomena (Section 3.4). Scientific problems carry in general an inherent time-base, the *physical time* of nature. Here physical time of nature is assumed to be an unstoppable linear progression, the global flow that moves every aspect of the real-world in real time. Time itself is not an a priori existence, which measures changes. It is a clock's regular change relative to the observer that specifies time (relativity theory). The consequence of this is that physical time can be defined globally and locally using different time-base representations.

The modeling and simulation community has taken time beyond this sense of global progression and locality. In modeling and simulation time becomes an additional knowledge source, providing information that is meticulously recorded so that moments in physical time can be recreated at will. Simulation is by definition a physical dynamic process, which tries to represent the real-world state and characteristics changes over a single physical time-base. The hierarchical organization and interaction of model components and simulation subsystems imply complex relationships between the global time-base of the whole simulation system and possible local time-bases of its constituent parts. Therefore, time has a multi-dimensional character in simulation systems. A characteristic that requires careful management to ensure that all local time-bases representations are logically correlated, synchronized and projected onto the global time-base of the simulation systems representation for the same physical time-base. Such time management issues are most apparent and significant in parallel and distributed simulation systems. For example, in a distributed simulation there may be simulations that run at different real-time update rates that have to be synchronized to a global wall-clock time and take into account the effects of communication latencies on the time-frame validity of certain data.

The correspondence of the various representations of time within a model to the simulation system of the physical time affects the representational, behavioral and functional accuracy. Furthermore, it also affect and are an ordering mechanism for occurrences of system changes i.e. causality. Characterizations of multiple representations of the physical time and time management within a model or simulation system, and their effect on the eventual level of simulation fidelity are captured by the concept of temporality. *Temporality* is defined here as the model or simulation system quality of corresponding to the sequence of time. All together the concept of temporality is thus an essential aspect of simulation fidelity characterization, which allows to distinguish and to analyze the afore-mentioned effects of the multi-dimensional nature of time within a model or simulation systems.

Causality: The Level of Correspondence to Cause and Effect Relationships

In both the real-world and the simulation system there exist many observable relationships between a cause and its effects such as between regularly correlated events or phenomena of (sub)systems. These relationships are referred to as causal relationships, and provide explanations and reasoning mechanisms for system behavior and function. Causal relationships are classified as either incremental or continuous [40]. Incremental indicates a series of discrete set of causes and effects i.e. discrete system changes. Continuous causal relationships are relationships in which a cause results in continuous system changes called a coextensive effect. Each causal relationship type can apply to both the internal system behavioral and functional features as well as to external in terms of the interactions between systems. Having an internal causal relationship means that a specific cause at the system input results in a state change and in effect in the observed system output. In an internal causal relationship, a causal relationship yields that a local system state change leads to the local state change in the other systems over some time interval. Chains of these interactions create the ordered sequences of events and phenomena that are experienced in the real-world as being the overall system behavior and functions.

How well the simulation systems mimics the causal relationships of the real-world is important, since they affect the simulation outcome accuracy. Therefore the degree of this causal relationship preservation is thus an essential aspect of simulation fidelity characterization and specification, which is referred here by the concept of *causality*. Due to its close relationship to the independent time variable, causality is significantly influenced by the effects of temporal anomalies (time-based ordering and intervals). Causality in a simulation is not only affected by the anomalies in the actual implementation of the real-world causal relationships, they can also be significantly disturbed by the actual simulation system implementation in terms of its hardware and software characteristics. Such effects are most significant in parallel and distributed simulation systems where time management is a complex issue: communication latencies between the simulation system parts can create deviation of the real-world causality Furthermore, various local time-base representations and different update rates can produce similar problems.

Precision and Sensitivity: The Level of Internal Realization Quality

What aspects of reality (resolution) and with what level of accuracy they are represented by a simulation system depends on how these aspects are internally realized in terms of used models (mathematical and physical representations, algorithms and logic), associated parametric data-sets and their quality, software and hardware components (network capabilities, visual system characteristics, etc). The concepts of precision and sensitivity are fidelity descriptors for characterizing and specifying the quality of these internal realizations. In real-life system behavior and function, in terms of state, input and output variable values, have known (in)finite distances with which each of these values can be discriminated and a possible variance in the spread in the observation of these values. This also yields for parametric data-sets used in the simulation model. Limitations induced by a computational approximation and its implementation in the simulation system will also have their effect on how precise variables and parameters can be represented. Such limitations include round-off procedures, data interpolation and extrapolation intervals, integration step-sizes, and finite computer word lengths.

Precision characterizes how meticulously the simulation model internally realizes the replication of real-world system behavior and functions. This precision does affect the accuracy that can be achieved by a simulation system. For instance, given a true initial state value of *0.500* and two simulated representations (a float value and an integer value) of this state value. The accuracy defined as the inverse of the absolute difference between the true and simulated state value can never be higher than *2* for the integer value while for the float value representation it can become much larger.

Sensitivity describes the effects of imperfections and uncertainties of external stimulus (input variables) and internal simulation system parameters (data values) or structure on the accuracy of the simulated real-world system behavior and functions i.e. output. This approach captures the two common notions of the concept sensitivity as found in VV&A and fidelity literature, and which both have utility in simulation fidelity characterization and analysis.

The first notion considers sensitivity in terms of how sensitive the simulation output is to small changes in the system input variables, which links directly to the concept of interaction and accuracy [8] [42] [47]. Since any model or simulation system component is an approximation of the real-world its output will be inaccurate and when these components interact with each other via input output coupling this error will directly or indirectly propagate to all other components. Eventually these errors will affect the overall behavioral accuracy. Furthermore, this notion of sensitivity focuses on establishing the threshold for a change in value of a system input or combinations of system input must exceed before any observable changes in the system output can be noticed. Any differences of such thresholds compared to the real-world can cause anomalies in the overall behavioral accuracy.

The second notion of sensitivity focuses on the effects of model changes (parameter values and structure), extreme input values and combinations thereof on the simulation output [66] [67] [146]. Sensitivity from this perspective has utility in simulation fidelity characterization and analysis since it helps to determine the most important simulation model aspects. That means identifying those aspects that have the most significant effect on the measured simulation execution outcome accuracy. This information is most useful in determining where more effort should be placed on evaluating and when necessary reducing simulation errors and uncertainties to improve the simulation fidelity during the development and validation process. Therefore, any knowledge about these errors and uncertainties for these significant factors in the simulation model are essential drivers for the credibility that can be placed upon the simulation results and the associated risks involved in their usage.

4.4 Simulation Fidelity Theory: Formalisms

The fidelity definitions, concepts and characterizations discussed in this chapter have been treated in a non-formal manner. Here mathematical formalisms for both esoteric and pragmatic fidelity are developed, which provide a clear and pertinent definition for simulation fidelity. These formalisms together form the foundation for a more rigorous scientific approach to specify, quantify and utilize simulation fidelity in a coherent and uniform manner. Notations from set theory, discrete mathematics, statistics and probability are adopted to specify these formalisms, since this provides a generic, and simulation application and problem domain independent formulation that is commonly used in other formal approaches within modeling and simulation [8] [47] [62] [79] [137] [155] [156].

4.4.1 Esoteric Fidelity Formally Defined

Reality R is the knowledge source for specifying the structural and behavioral manifestation of any real-world system and is defined as:

$$R = R_{mat} \cup R_{im} \tag{4.1}$$

Here R_{mat} is material reality and R_{im} is imaginary reality (*Figure 3-1*). From the esoteric fidelity notion as discussed in Section 4.2.1, R_w is the real-world which is considered to be the fully known and error-free structured knowledge specification of that part of reality to be modeled or simulated. Therefore, the following relationship holds for R and R_w:

$$R_w \subseteq R \tag{4.2}$$

In this regard S represents the fully known and error-free structured knowledge specification of the simulation system. S is defined by the set:

$$S = \left(S_{exec}, \left\{ \left(S_{rwr}, S_{config} \right)_s \middle| \forall s \in S_{exec} \right\}, S_{model}, S_{support} \right) \tag{4.3}$$

Here S_{exec} is the set of identifiers for all possible simulation executions that can be performed with the simulation system. S_{rwr} is the external system knowledge specification of real-world reproduction as results from the simulation execution $s \in S_{exec}$ for the given simulation configuration input setting S_{config} (Section 3.2.3). S_{model} and $S_{support}$ are the internal system knowledge specifications for the simulation model and support systems respectively, which together form the simulation system's architecture as discussed in Section 3.2.3. Depending on their complexity both S_{model} and $S_{support}$ can be specified with a composite-component system knowledge formalism (Section 3.4.6). This means that the simulation model and support systems can be considered as a composition of interacting sub-models and systems. Summarized the complete structured knowledge specification for a simulation system is thus the knowledge about the simulation model and support systems itself complemented with knowledge of its reproduction of the real-world (S_{rwr}) during simulation execution. Mathematically, S_{rwr}, S_{config}, S_{model} and $S_{support}$ are related according the following non-invertible simulation execution mapping E_{exec}:

$$E_{exec} : S_{config_s} \times S_{model} \times S_{support} \rightarrow S_{rwr_s} \quad \text{for} \ \forall s \in S_{exec} \tag{4.4}$$

For structural autonomous simulation systems $S_{config} = \varnothing$ since the configuration is statically embedded in the simulation model. In case a simulation system has a deterministic and non-random representation of the real-word, E_{exec} will then be a *one-to-one* mapping. Having defined the fully known and error-free structured knowledge specification for the real-world and the simulation system, it is now possible to formally determine the difference between both specifications in order to specify *esoteric fidelity*. As stated by fidelity theorem 4, fidelity qualification and quantification is a multidimensional and multifaceted problem (Section 4.3.2). Therefore, it is assumed that there exists a finite set $C_{\Delta_{RS}}$ containing all N possible, so-called, *fidelity evaluator functions* operating on R_w and S, and is defined as follows:

$$C_{\Delta_{RS}} = \left\{ c_{\Delta_1}\left(R_w, S\right), c_{\Delta_2}\left(R_w, S\right), \ldots, c_{\Delta_N}\left(R_w, S\right) \right\} \tag{4.5}$$

In here c_{Δ_i} represents the i^{th} fidelity evaluator function. A *fidelity evaluator function* is a multi-variable function, which generates a structured set Δ_{RS_i} of m quantifications and/or qualifications that specify the difference between a certain subset of aspect(s) of the simulated real-world reproduced by simulation system S and the real-world R_w. These differences are measured in the simulation fidelity characterizations areas of resolution, accuracy, temporality, interaction and causality (See Section 4.3.3). A fidelity evaluator function c_{Δ_i} can formally defined by the following mapping:

$$\begin{aligned} & c_{\Delta_i} : R_w \times S \rightarrow \Delta_{RS_i} \\ & \text{with} \\ & \Delta_{RS_i} = \left\{ \left(d_1, \ldots, d_m\right) \middle| d_1 \in \Delta_{RS_{i,1}}, \ldots, d_m \in \Delta_{RS_{i,m}} \right\} \end{aligned} \tag{4.6}$$

Here d_j is the value of the j^{th} difference quantification or qualification that results from a fidelity evaluator function c_{Δ_i}. The *range* of d_j is $\Delta_{RS_{i,j}}$.

Fidelity is per definition inverse proportional to the specified difference between the simulated real-world reproduced by simulation system S and the real-world R_w (Section 4.2.2). This is expressed by the set Δ_{RS}^{-1} defined as:

$$\Delta_{RS}^{-1} = \bigcup_{i=1}^{N} c_{\Delta_i} \circ k_i^{-1} \tag{4.7}$$

In *Expression 4.7* Δ_{RS}^{-1} is the union of all inverse proportional scaled quantifications and/or qualifications that result from the *fidelity evaluator function set* $C_{\Delta_{RS}}$. Each element of Δ_{RS}^{-1} is derived through a function k_i^{-1} that maps each fidelity evaluator function output Δ_{RS_i} to its inverse proportional equivalent as follows:

$$k_i^{-1} : \Delta_{RS_i} \rightarrow \left\{ k_1 \cdot \left(d_{i,1}\right)^{-1}, ..., k_m \cdot \left(d_{i,m}\right)^{-1} \middle| d_{i,j} \in \Delta_{RS_i} \wedge k_j \in \Re_1^+ \right\} \qquad (4.8)$$

Esoteric fidelity is now formally defined utilizing equations 4.1 to 4.8:

$$F_{esoteric} = \left\langle R_w, S, C_{\Delta_{RS}}, \Delta_{RS}^{-1} \right\rangle \qquad (4.9)$$

Expression 4.9 states that the esoteric fidelity $F_{esoteric}$ of a simulation system is characterized by the fully known and error-free structured knowledge of the real-world R_w and the simulation system S plus the set of all possible fidelity evaluator functions, which formally quantify and/or qualify the inverse difference between R_w and S as specified by the resulting set Δ_{RS}^{-1}. This mathematical formulation for *esoteric* fidelity is fully consistent with both the fidelity theorems (Section 4.3.2) and the informal textual definition developed in Section 4.2.1.

4.4.2 Practical Fidelity Formally Defined

According to fidelity theorems 1 and 2 *esoteric* fidelity can never be attained in practice due to various kinds of limitations in observing and specifying both the real-world R_w and simulation system S (Section 4.3.2). In practice fidelity is thus not directly measured against R_w itself but against the so called the fidelity referent R_{ref} (Section 4.2). The real-world reference knowledge specification R_w^{ref} contained within the fidelity referent is related to the real-world through what is referred here as a *reality knowledge elicitation* function E_{rw}. A relationship that establishes a correspondence between a pair of system knowledge specifications whereby features of one system are preserved in the other are referred in literature as *preservation relations* or *system morphism* [156]. Since in practice R_w^{ref} is not a perfect and exact representation of R_w their relationship is considered to be an *approximated morphism*. These deviations and associated uncertainties are caused by a set of limitations L_R^{elicit} encountered or placed upon the elicitation of real-world knowledge E_{rw} during the simulation development and validation process. Formally R_w^{ref} is related to R_w through mapping E_{rw} as follows:

$$E_{rw} : R_w \times L_R^{elicit} \rightarrow R_w^{ref} \qquad (4.10)$$

When the limitation set $L_R^{elicit} = \{\varnothing\}$, E_{rw} becomes the identity mapping resulting in that $R_w = R_w^{ref}$. In all other cases the knowledge in R_w^{ref} and R_w are symmetrically different:

$$\delta R_w^{ref} = R_w \oplus R_w^{ref} = (R_w \setminus R_w^{ref}) \cup (R_w^{ref} \setminus R_w) \qquad (4.11)$$

In *Expression 4.11* δR_w^{ref} represents an n-dimensional set of errors made in the real-world reference knowledge specification process with respect to the real-world R_w. As can been seen from equation 4.11 and abstractly illustrated in the Venn diagram below (*Figure 4-11*) this fidelity referent knowledge error is composed of two parts. The first part $R_w \setminus R_w^{ref}$ represents the real-world knowledge not included in the fidelity referent. The second part $R_w \setminus R_w^{ref}$ represents the faulty real-world knowledge or misinformation imparted in the fidelity referent.

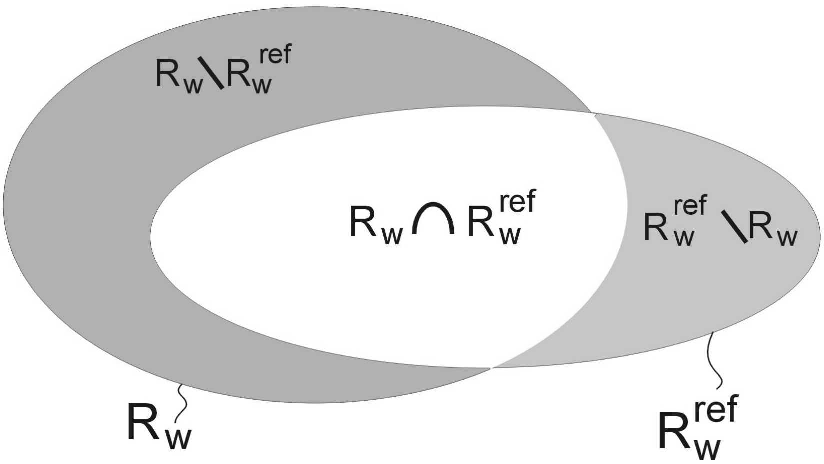


Figure 4-11 Venn Diagram Fidelity Referent Error Components

Although a portion of the errors in the set δR_w^{ref} is identifiable but due to limitations in L_R^{elicit} their magnitude is often unknown or at best roughly estimated. However, another portion of possible existing errors in δR_w^{ref} are usually unidentifiable due to elicitation limitations in L_R^{elicit}. These lacks of knowledge are a cause for uncertainties in the fidelity referent knowledge. Characterization of uncertainty in practice yields a mixed set of qualifications and where possible quantifications for the various uncertainty sources or components [31] [95]. The uncertainty associated with the fidelity referent is defined as:

$$U_\delta\left(R_w^{ref}\right) = \left\{u_\delta\left(R_w^{ref}\right)_1, \ldots., u_\delta\left(R_w^{ref}\right)_n\right\} \tag{4.12}$$

Here $u_\delta\left(R_w^{ref}\right)_i$ represents the i^{th} of n measures of uncertainty resulting from a single or a combined contribution of several uncertainty sources in real-world reference knowledge specification R_w^{ref}.

Using the previous results, the fidelity referent can now formally be defined by following high-level quintuple:

$$R_{ref} = \left\langle R_w^{ref}, \mathrm{E}_{rw}, L_R^{elicit}, \delta R_w^{ref}, U_\delta\left(R_w^{ref}\right)\right\rangle \tag{4.13}$$

Chapter 5 will elaborate more on the underlying concepts, the structured knowledge specification format details and formalisms for the components of R_{ref}.

The elicitation of simulation system knowledge from S is subjected to similar kinds of limitations (L_S^{elicit}) as placed upon in the aforementioned elicitation of real-world knowledge from R_w. Therefore, only an approximated simulation system knowledge specification S_{appx} can exist in practice. This S_{appx} is related to S through what is referred here as a *simulation system knowledge elicitation* function E_{sim} and is defined as follows:

$$\mathrm{E}_{sim} : S \times L_s^{elicit} \to S_{appx} \tag{4.14}$$

When the limitation set $L_S^{elicit} = \{\varnothing\}$, E_{sim} naturally becomes the following identity function $1_s : S \to S = S_{appx}$. Like for R_w^{ref} the knowledge contained in S_{appx} and S are symmetrically different:

$$\delta S_{appx} = S \oplus S_{appx} = (S \setminus S_{appx}) \cup (S_{appx} \setminus S) \tag{4.15}$$

In 4.15 δS_{appx} represents the set of errors made in the simulation system knowledge elicitation process. As for δR_w^{ref} the knowledge error is composed of two parts: $S \setminus S_{appx}$ and $S_{appx} \setminus S$. These represent the simulation system knowledge not included and the faulty simulation system knowledge or misinformation imparted in the S_{appx} respectively. Since the simulation developer is the creator of the simulation system, he or she has in normal circumstances good control over and knowledge about the simulation system. In other words the limitation set L_S^{elecit} will usually be much smaller than for the development of the fidelity referent. Therefore, compared to δR_w^{ref} it is most likely that δS_{appx} will have a significant smaller knowledge elicitation error and associated uncertainty or can even be reduced to almost zero. The uncertainty associated with this simulation system knowledge specification is defined as:

$$U_\delta\left(S_{appx}\right) = \left\{u_\delta\left(S_{appx}\right)_1, ..., u_\delta\left(S_{appx}\right)_m\right\} \tag{4.16}$$

Here represents $u_\delta\left(S_{appx}\right)_i$ the i^{th} of m measures of uncertainty resulting from a single or a combined contribution of several uncertainty sources in the simulation system knowledge specification. The total simulation system knowledge specification S_{spec} is now defined by the following quintuple:

$$S_{spec} = \left\langle S_{appx}, \mathrm{E}_{sim}, L_s^{elicit}, \delta S_{appx}, U_\delta\left(S_{appx}\right)\right\rangle \tag{4.17}$$

Knowing that the real-world R_w and simulation system S knowledge cannot be fully known and is thus subjected to errors and uncertainties in practice, it is only possible to formally determine the difference between both R_w and S based upon using their approximating equivalents R_w^{ref} and S_{appx} within the fidelity evaluator function set $C_{\Delta_{RS}}$ (Equation 4.5). Although $C_{\Delta_{RS}}$ is a finite set, its cardinality will usually be too large given the practical constraints to perform all fidelity evaluations. For instance, simulation development constraints such as time, costs or regulations may enforce that only a subset of all possible fidelity evaluations can be performed or are necessary (see Section 8.2 on fidelity requirements). Furthermore, the limited data availability for R_{ref} and S_{spec} may result in insufficient data, which is needed to properly perform certain fidelity evaluations. Therefore, in practice there exist a set $\tilde{C}_{\Delta_{RS}}$ of practically possible fidelity evaluator functions such that $\tilde{C}_{\Delta_{RS}} \subseteq C_{\Delta_{RS}}$:

$$\begin{aligned}&\tilde{C}_{\Delta_{RS}} = \left\{ c_{\Delta_1}\left(R_w^{ref}, S_{appx}\right), c_{\Delta_2}\left(R_w^{ref}, S_{appx}\right), \ldots, c_{\Delta_n}\left(R_w^{ref}, S_{appx}\right) \right\} \\ &\text{with} \\ &n \leq \left| C_{\Delta_{RS}} \right| \end{aligned} \tag{4.18}$$

In here c_{Δ_i} is defined by the following mapping:

$$\begin{aligned}&c_{\Delta_i} : R_w^{ref} \times S_{appx} \to \tilde{\Delta}_{RS_i} \\ &\text{with} \\ &\tilde{\Delta}_{RS_i} = \left\{ \left(\tilde{d}_1, \ldots, \tilde{d}_m\right) \middle| \tilde{d}_1 \in \Delta_{RS_{i,1}}, \ldots, \tilde{d}_m \in \Delta_{RS_{i,m}} \right\}\end{aligned} \tag{4.19}$$

Here $\tilde{d}_j$ is the value of the j^{th} difference quantification or qualification that results from a fidelity evaluator function c_{Δ_i} now operating on R_w^{ref} and S_{appx} instead. Chapter 7 will elaborate more on possible fidelity evaluator functions that can be part of set $\tilde{C}_{\Delta_{RS}}$ along with their underlying concepts, methods and formalisms.

Likewise, the result of executing these fidelity evaluator functions on R_w^{ref} and S_{appx} will also be a set $\tilde{\Delta}_{RS}^{-1}$ of fidelity quantifications and/or qualifications such that $\tilde{\Delta}_{RS}^{-1} \subseteq \Delta_{RS}^{-1}$. Thus $\tilde{\Delta}_{RS}^{-1}$ is defined as:

$$\begin{aligned}&\tilde{\Delta}_{RS}^{-1} = \bigcup_{i=1}^{n} c_{\Delta_i} \circ k_i^{-1} \\ &\text{with} \\ &k_i^{-1} : \tilde{\Delta}_{RS_i} \to \left\{ k_1 \cdot \left(\tilde{d}_{i,1}\right)^{-1}, \ldots., k_m \cdot \left(\tilde{d}_{i,m}\right)^{-1} \middle| \tilde{d}_{i,j} \in \Delta_{RS_i} \wedge k_j \in \Re_1^+ \right\}\end{aligned} \tag{4.20}$$

Each fidelity evaluator function can introduce an additional error δc_{Δ_i} above the errors in R_w^{ref} and S_{appx}, due to things such as calculation precision limits, mistakes and blunders in its execution. The total error $\delta\tilde{\Delta}_{RS_i}$ in the i^{th} difference specification $\tilde{\Delta}_{RS_i}$ is thus determined by the following error specification function:

$$\begin{aligned}&E_{\tilde{\Delta}_{RS}} : \delta c_{\Delta_i} \times \delta R_w^{ref} \times \delta S_{appx} \to \delta\tilde{\Delta}_{RS_i} \quad \text{with} \\ &\delta\tilde{\Delta}_{RS_i} = \left\{ \left(\delta d_1, \ldots, \delta d_m\right) \middle| \delta d_1 \in \Delta_{RS_{i,1}}, \ldots \delta d_m \in \Delta_{RS_{i,m}} \right\}\end{aligned} \tag{4.21}$$

In *Expression 4.21* δd_j represents the error in j^{th} difference qualification or quantification resulting from the belonging i^{th} fidelity evaluator function execution. This means that the error of $\tilde{\Delta}_{RS}^{-1}$ is the union of all inverse proportional scaled quantification and/or qualification errors $\delta\tilde{\Delta}_{RS_i}$, which result from the execution of the *fidelity evaluator function set* $\tilde{C}_{\Delta_{RS}}$ (Equation 4.18). Although the fidelity evaluator functions themselves do not exhibit or introduce any uncertainty when executed properly or negligibly small with respect to the uncertainties $U_\delta\left(S_{appx}\right)$ and $U_\delta\left(R_w^{ref}\right)$ in practice, they in theory contribute to the total uncertainty of each resulting difference quantification or qualification $\tilde{\Delta}_{RS_i}$. Therefore, the total uncertainty of $\tilde{\Delta}_{RS}^{-1}$ looks as follows:

$$U_\delta\left(\tilde{\Delta}_{RS}^{-1}\right) = U_\delta\left(R_{rw}^{ref}\right) \cup U_\delta\left(S_{appx}\right) \cup U_\delta\left(\tilde{C}_{\Delta_{RS}}\right) \tag{4.22}$$

Utilizing the previous results a weaker mathematical formulation of the fidelity concept can now be developed in the form of the following sextuple, which is the formal definition for *practical* fidelity (Section 4.3):

$$F_{practical} = \left\langle R_{ref}, S_{spec}, \tilde{C}_{\Delta_{RS}}, \tilde{\Delta}_{RS}^{-1}, \delta\tilde{\Delta}_{RS}, U_\delta\left(\tilde{\Delta}_{RS}^{-1}\right)\right\rangle \tag{4.23}$$

Expression 4.23 is fully consistent with both the fidelity theorems and the informal textual definition for *practical* fidelity developed in Section 4.3.2. This formal definition states that, in practice, simulation fidelity is characterized by the fidelity referent R_{ref} and the structured simulation knowledge specification S_{spec} plus a limited set of practically possible fidelity evaluator functions, which formally quantify and/or qualify the inverse difference between R_w^{ref} and S_{appx} as specified by the resulting set $\tilde{\Delta}_{RS}^{-1}$. Furthermore, it explicitly states that fidelity in practice can never be exactly known ($\delta\tilde{\Delta}_{RS_i}$) and contains a level of uncertainty $U_\delta\left(\tilde{\Delta}_{RS}^{-1}\right)$. Since expression 4.23 not only provides the resulting quantified and/or qualified fidelity ($\tilde{\Delta}_{RS}^{-1}$) but also all information i.e. evidence necessary to make this fidelity specification results fully traceable, reproducible and assessable for its credibility during simulation development and validation. This is also useful for simulation system reuse and fidelity comparisons. See also fidelity theorems 9 and 12 in Section 4.3.2.

4.5 Summary

This chapter developed the fundamental theory for a unified fidelity framework by synthesizing existing fidelity knowledge into a single consistent and formal approach for simulation fidelity. It has been shown that the most conceptually right definition for fidelity, esoteric fidelity, is something idealistic that can never be attained in real modeling and simulation practice. Based upon the arguments underlying this conclusion the fundamental concepts for a pragmatic simulation fidelity theory have been developed. The most fundamental concept herein is the real-world reference knowledge standard paradigm, called fidelity referent, for practical fidelity measurement. From this paradigm a pragmatic formulation for the concept of fidelity has been created. This formulation is supported by twelve presented fidelity theorems, which together outline the basic principles, propositions and postulates for a generic simulation fidelity assessment methodology and practice. Using the object-oriented approach to system knowledge specification presented in the previous chapter, eight descriptive concepts for simulation fidelity characterization have been developed: detail, resolution, accuracy, interaction, temporality, causality, precision and sensitivity. All these fundamental elements have been formalized into an application and problem domain independent mathematical formulation for practical fidelity:

$$F_{practical} = \left\langle R_{ref}, S_{spec}, \tilde{C}_{\Delta_{RS}}, \tilde{\Delta}_{RS}^{-1}, \delta\tilde{\Delta}_{RS}, U_{\delta}\left(\tilde{\Delta}_{RS}^{-1}\right)\right\rangle$$

This formulation is thus composed of six elements: the fidelity referent specification (R_{ref}), the simulation system knowledge specification (S_{spec}), the fidelity evaluator function set ($\tilde{C}_{\Delta_{RS}}$) and their outcomes ($\tilde{\Delta}_{RS}^{-1}$) plus associated errors ($\delta\tilde{\Delta}_{RS_i}$) and uncertainties ($U_{\delta}\left(\tilde{\Delta}_{RS}^{-1}\right)$). As long as each of these six elements comply with the simulation fidelity theory developed in this chapter, each element can be constructed and implemented as desired by a simulation system developer in order to suit the particular simulation problem at hand.

The remaining part of this thesis will focus on the detailed treatment of the structure and contents each of these elements of pragmatic fidelity as part of the unified fidelity framework, starting with the practical implementation of the fidelity referent paradigm in Chapter 5. Next Chapter 6 will detail on the practical simulation system knowledge specification, which together with the fidelity referent form the basis for qualifying and quantifying the level of simulation fidelity. Based on the eight descriptive fidelity concepts, the fidelity referent and simulation knowledge specification, Chapter 7 provides a presentation of the major metrics and methods to perform this simulation fidelity qualification and quantification. Chapter 8 shows several important fidelity application concepts and assessment techniques that all build upon these forgoing chapters. Finally, Chapter 9 will detail on a methodology to systematically integrate and apply all these unified fidelity framework building blocks within the model and simulation enterprise presented in Chapter 3. In here the unified fidelity framework is expanded with a fidelity management process model.

5 Unified Fidelity Framework: Real-World Reference Knowledge Specification

5.1 Introduction

The previous chapter developed a general applicable simulation fidelity definition by developing a set of basic definitions, characterization and formalisms for a unifying fidelity framework. In here fidelity is formally defined in terms of six constituent elements (*Expression 4.23*). One of these elements and key element in specifying the simulation system fidelity capabilities and the fidelity of the simulation execution is the fidelity referent paradigm (Section 4.3.1). A fidelity referent provides the comparative base of real-world knowledge for the pragmatic measurement of simulation fidelity. This chapter will focus on the practical realization of this fidelity referent paradigm within the unified fidelity framework in terms of the development of a fidelity referent knowledge-base architecture. Furthermore, the underlying and associated mathematical formulations of its constituent specification templates will be developed and presented simultaneously.

This chapter starts in Section 5.2 with a high-level overview and relationships of the major building blocks that together form the fidelity referent knowledge-base. Next each of these building blocks is developed and presented in detail in Sections 5.3 to 5.5. Excerpts drawn from two aerospace simulation fidelity case studies are used in this chapter to illustrate each part of the developed fidelity referent knowledge-base template (Appendix C). A fully detailed presentation of both case-study referent knowledge-bases, implemented as a workbook of linked spreadsheets, is beyond the scope of this dissertation. Therefore, the majority of content of these referents, although equally important in the complete fidelity characterization process, is deliberately summarized or omitted.

5.2 Fidelity Referent Knowledge-Base: Structural Overview

In chapter 4 a mathematical definition for the fidelity referent (R_{ref}) has been developed (*Expression 4.13*). The unified fidelity framework implementation of this R_{ref} constitutes a structured set of generic and linked specification templates, which together define a knowledge-base architecture. These templates are grouped in three major subsets:

- *Real World Reference Knowledge Specification* (Section 5.4), consisting of a structured template for properly specifying the actual real-world reference knowledge $R_w^{ref} \in R_{ref}$, which is used for the actual fidelity measurement of the simulation system.
- *Elicitation, Error and Uncertainty Knowledge Specification* (Section 5.5), consisting of a structured template for specifying the remaining elements of the

quintuple R_{ref}. These are the real-world reference knowledge error (δR_w^{ref}) and uncertainty ($u_\delta\left(R_w^{ref}\right)$), plus how this knowledge has been elicited (E_{rw} and L_R^{elicit}).

- *Additive & Management Knowledge Specification* (Section 5.3), containing all additional knowledge necessary to properly manage, apply, maintain and trace the referent knowledge-base contents (R_{ref}) throughout its life cycle.

Figure 5-1 gives a graphical representation of the fidelity referent knowledge-base architecture. The intersections between the subsets signify that for more practical utility some certain knowledge, discussed in the following sections, is shared or specified at multiple places. Arrows indicate the physical links to navigate between related knowledge-base sections.

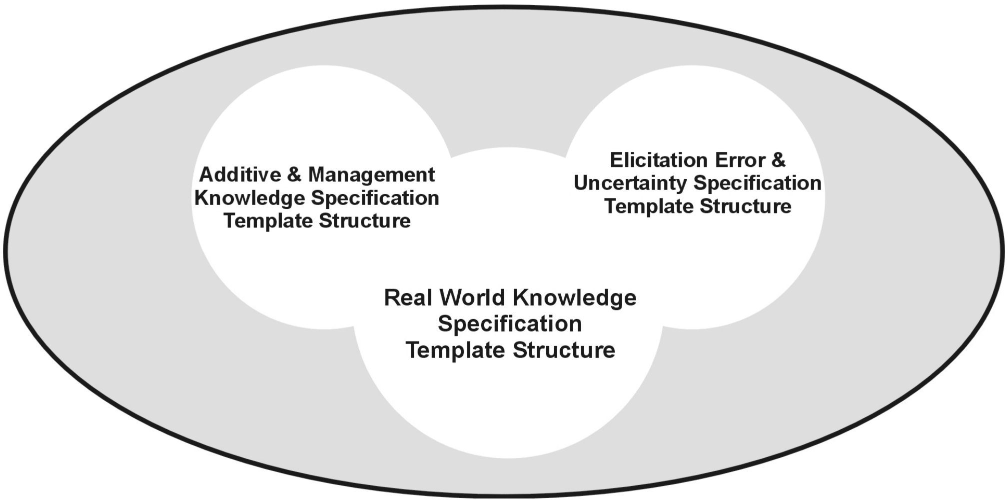


Figure 5-1 Fidelity Referent Knowledge Base Structural Overview

5.3 Fidelity Referent Knowledge-Base: Additive & Management Knowledge Specification Templates

Besides the actual real-world reference knowledge specification elements of R_{ref} (*Expression 4.13*) discussed in the next sections, the fidelity referent knowledge base structure should also provide a well-defined body of additive knowledge elements for managing and maintaining this reference knowledge in terms of quality level, applicability, suitability and traceability. These additive elements are essential for the proper application of a fidelity referent within the fidelity assessment activities throughout the whole simulation development and validation process. A fidelity referent knowledge-base structure must at least contain and address the following five additive specification elements which are discussed in next the subsequent sections:

- *Identification and Management Information Section*
- *Applicability and Status Information Section*
- *Developer and Validation Agent Information Section*
- *Used Knowledge and Data Sources Section*
- *Utility Knowledge and Data Section*

5.3.1 Identification and Management Information Section

This section provides the referent identification, contact and management information necessary for the life-cycle management of the referent. The section comprises four subsections:

- *Referent Identification*. This section assigns a textual name or an identification code, by which the referent can be identified, referred to and searched for. Furthermore, it specifies the version, creation and latest revision date of the referent. This provides additional identification to help differentiate between possible versions of the same referent that have been developed and used in the past or may still be in use.
- *Referent Managing Organization*. The name and general contact information of the company or organization that is responsible for managing the referent and maintaining its contents. In case some change occur
- *Referent Managing Director*. This section gives the name, position and contact information of the person currently in charge of the referent life-cycle. The managing director serves as the first point of contact to obtain additional information and resolving any problems with the referent.
- *Revision History*. This section provides a list of revision data sheets containing the revision dates, description and links to the real-world reference knowledge data that has been changed or added (Section 5.4.1), the rationale for the modification and reference to the participants in the *Developer and Validation Agent Information Section* (Section 5.3.3) that made and accredited the modification. This is necessary for traceability of any changes and possible problems that may occur due to this such as entering incorrect knowledge.

Table 5-1 below gives an excerpt from the Future Airspace Simulation Environment (FASE) referent Identification and Management Information section. For brevity only one of the two revision data sheets is given, which outlines only one example of the made modifications to the real-world reference knowledge. Underlining of text represents links to other parts of the referent structure where additional knowledge is provided.

Referent Identification	
Referent Name:	*Future Airspace Simulation Environment (FASE) Referent*
Creation Date:	*10-09-2000*
Referent Version:	*V0.3*
Revision Date:	*27-03-2002*
Referent Managing Organization	
Name:	*Delft University of Technology Aerospace Control & Simulation Division*
Address:	*Kluyverweg 1, 2629 HS Delft, The Netherlands*
Phone:	*(+ 31) 15-2782094*
Fax:	*(+ 31) 15-2786480*
Website:	*http://www.cs.lr.tudelft.nl/*
Referent Managing Director	
Name:	*Z.C. (Manfred) Roza M.Sc.*
Address:	*Kluyverweg 1, 2629 HS Delft, The Netherlands. Room 0.24*
Phone:	*(+ 31) 15-2785374*
E-mail:	*z.c.roza@lr.tudelft.nl*

Revision History	
…	
Revision Data Sheet 2	
Revision Number:	*2*
Revision Date:	*27-03-2002*
Modification Description	
General Remarks:	*The referent is extended with additional knowledge of the existing airspace navigation systems and aircraft avionics systems using this navigation aids. This extension was performed in the context the FASE infrastructure enhancement with a new CNS federate simulating an airspace navigation system environment.*
1 Element Changed or Added	
Name:	*Jeppensen navigation aids information charts for Europe*
Rationale:	*Added for assessing the completeness, accuracy and correlation of the simulated navigation aids in the FASE CNS federate and their representations in ATCO working position and avionics display systems.*
Comments:	*Jeppensen charts are governmental certified (FAA, JAA etc.) and the used charts are listed in the Utilized Knowledge and Data Sources Section.*
2 Element Changed or Added	
Name:	...
Rationale:	...
Comments:	...
People Involved	
ID Participant:	Participant_001
ID Participant:	Participant_006

Table 5-1 Referent Identification and Management Information Section Example

The contents of this referent section may change overtime due to the release of a new update version, or changes in management organization or director point of contact details. It is the referent managing organization responsibility to inform the current and/or potential referent users of any changes. Depending on the applicability range and authority level of the referent it could be necessary to report the changes to a broader community than those referent users registered in the applicability and status information section (Section 5.3.2).

5.3.2 Applicability and Status Information Section

This section specifies to which problem and application domain the referent is intended for. Furthermore, it specifies the authority status within these domains and its known applications. This section comprises the next five subsections:

- *Application Domain Specification*. This subsection documents the application domain(s) for which the referent is intended.
- *Problem Domain Specification*. Here the problem domain(s) for which the referent is intended are specified.
- *Real-World Coverage Description*. This subsection summarizes what part of reality is covered by the referent. Together with the previous two domain specifications it provides a first indicator for deciding whether the referent content is possibly suitable for the user's application.
- *Authority Status*. The authority status section specifies the degree and application range for which the referent content has been or is accredited. In here three standard types of qualifiers are used. *Normative Standard Level*: the referent is accredited for

specific applications within a small simulation business, such as among several divisions of a company. *Authoritative Standard Level*: the referent is a community wide accepted and accredited standard for a problem or application domain. *Non-standard Level*: the referent is accredited for a single specific application. Furthermore, information is provided on the organizations that are responsible for the accreditation, the record ID with this organization, and any comments made during the accreditations process itself. This altogether provides a high-level indication of the quality and credibility of the referent's content.

- *Usage History*. This section gives a list of all known simulation applications that have used this referent in their fidelity assessment. It provides a link to all applications that could serve as an example of how this referent should be (re)used. Furthermore, identifies possible similar and compatible simulations that might be used in the current application. For each listed application the point-of-contact (POC) information is specified along with a summary description of the simulation application. This POC information is also used by the referent management organization for notifying the referent users when a new update is available. For these two reasons referent users have to be asked to renew their POC information and referent applications with the referent management organization when necessary.

Table 5-2 at the next pages gives an excerpt from the FASE referent Applicability and Status Information section. As can be seen in this table the referent knowledge has been accredited as a normative standard by and for the Aerospace Control & Simulation division.

Referent Target Domains	
Application Domain(s):	*Research, Development & Engineering (RD&E)*
Problem Domain(s):	*Air-Traffic Control and Management of Airspace Systems*
Real World Coverage Description	
Describes the navigation, surveillance, communication systems, airborne entities, ground based controller facilities, airspace structure, procedures and organization of current and possible future airspace systems	
Referent Authority Status	
…	
1 Accreditation	
Accreditation Level:	*Normative*
Process Status:	*~~Not Yet Initiated, In Progress~~, Accomplished*
Process Comments:	*Accredited in May 2002*
Applicability Range:	*Aerospace Control & Simulation research projects: En-Route Airspace 2020 Research Simulations for in-house Research to Operational and Human-Factor Issues of the Free Flight Paradigm*
Accreditation Agency:	*Delft University of Technology Aerospace Control & Simulation Division*
Address:	*Kluyverweg 1, 2629 HS Delft, The Netherlands*
Phone:	*(+ 31) 15-2785374*
E-mail:	*info@cs.lr.tudelft.nl*
Website:	*http://www.cs.lr.tudelft.nl/*
Accreditation Record ID:	*CS-AC-RECORD-001*

…

Usage History	
...	
1 Application	
Organization:	*Delft University of Technology Aerospace Control & Simulation Division*
POC Name:	*Z.C. (Manfred) Roza M.Sc.*
Address:	*Kluyverweg 1, 2629 HS Delft, The Netherlands*
Phone:	*(+ 31) 15-2785374*
E-mail:	*z.c.roza@lr.tudelft.nl*
Website:	*http://www.cs.lr.tudelft.nl/*
Application Description:	*Free-Flight paradigm demonstration federation to illustrate and evaluate the usage of HLA distributed simulations for the research on future airspace architectures and procedures.*

Table 5-2 Referent Applicability and Status Information Section Example

5.3.3 Developer and Validation Agent Information Section

This section lists all individuals and subject matter experts (SME) whom were involved in the development and validation of the referent real-world knowledge contents specification. It describes and rates their contribution. For each person involved the next data is specified:

- *Participant ID and Personal Details*. This section gives the participant ID. The exact personal details (name, employing organization, and contact information) are stored in a separate list. The reasons for this are the following. First of all for privacy reasons of each participant. Secondly, hiding personal details might help prevent biased judgments about the participant's contributions by either validation or accreditation agents. Thirdly, the personal details are of importance during the referent development and upgrades. Therefore once the referent is accredited the personal details have no real significance for its users and can be decoupled for personal privacy reasons.
- *Qualifications and Expertise*. Specifies of each person involved his or her general knowledge, specialization, education and experience relevant to the referent application and problem domain. Helps rating the value (error, uncertainty) of his or her contribution as described in the next sub-section.
- *Previous SME Appointments and Interest*. Listing of previous participations as a SME in simulation development and validation process. This information is useful for quality of the real-world reference knowledge specification or review process. An experienced SME is more likely to know what is expected from him in terms of good judgment, perspective and process itself [109]. Any special interest or conflicting positions that might effect positively or negatively the person's objectivity should be specified.
- *Referent Contributions and Competence Ratings*. This section describes the roll of the participant. This could be either developer or validation agent. It also states all contributions of the participant to the referent. For each contribution the exact task description of this person in the development or validation is specified, which helps tracing the persons responsible for certain knowledge decisions or contributions in order to obtain additional information during referent development or upgrades. In this tasks description links can be added to the specific participant's contributions in the referent real-world knowledge specification part (Section 5.4). Furthermore, for each participant's contribution a competence rating is specified to qualify his or her

competence level in this specific referent knowledge area. In here the following nominal rating scale is used:

Named Score	Fuzzy Numerical Value
Very High	*[100, 86]*
High	*(86, 72]*
More than Average	*(72, 58]*
Average	*(58, 44]*
Less than Average	*(44, 30]*
Low	*(30, 16]*
Very Low	*(16, 0)*

Table 5-3 Referent Developer & Validation Agent Contribution Rating Scale

In *Table 5-3* the boundary value zero is not included, since this value indicates that the participant is totally incompetent and therefore his contribution to this referent must be omitted. The referent developer and domain SME competence ratings are confidence indicators in the development and validation process used as evidence to decide on the sufficient completeness, correctness and uncertainties of the real-world reference knowledge contributions in relationship to the target application and problem domain at hand. Furthermore, the competence ratings of the referent validation agents are assigned and used as evidence in the simulation system accreditation process to demonstrate quality and credibility of the simulation fidelity specifications to the sponsor. Incase several SME collectively contribute to the same part of the referent their individual competence ratings can be combined into an single reference section, for instance a weighted arithmetic average, to assist in the reference knowledge development and validation. More on this SME issue is presented in this and the next chapters.
As an example, consider an engineer with who may has high competence in structural engineering, contributes to an aerospace application referent as a domain knowledge SME. His contribution to the structural deformation behavior knowledge for an aircraft wing would be significant while his contribution to aerodynamic knowledge about the same aircraft wing would be very minimal. Obviously, if this engineer was the only one to create the aerodynamic referent knowledge for this aircraft simulation this competence rating could be an indicator for a sponsor not to accept the simulation system.

Table 5-4 presents an excerpt from the FASE referent developer and validation agent information section. This instance gives the details of one of the students who participated in the FASE project. Again underlining of text means a physical link to another part of the referent.

Referent Developer and Validation Agent Information	
	…
1 Participant Information	
ID Participant:	Participant_003
Expertise and Qualification	
General Knowledge:	*- General Aerospace Engineering Knowledge*
Specific Knowledge:	*-Conventional and Future Air-Traffic Control and Management*
Education:	*- B.Sc. in Aerospace Engineering from Delft University*
Experience:	*-Development of ATC/ATM simulation tools at Euro-Control Brussels* *-Development of Airspace Navigation and FMS Tool for Bae-Systems*

Appointments and Interests	
SME Appointments:	*None*
Interests & Conflicts:	*Contribution is part of M.Sc. thesis*
Referent Contributions	
…	
1 Contribution	
Participant Role(s):	*Domain Subject Matter Expert*
Task Description:	*Providing information on the aspects of aircraft operation procedures in en-route airspace structure plus associated air-traffic controller roles in such an airspace*
Task Competence:	*More than average (equivalent value: 70)*
Task Execution Period:	*January 2001 to July 2001*

Table 5-4 Referent Developer and Validation Agent Section Example

5.3.4 Utilized Knowledge and Data Sources Section

This section fully references and lists all knowledge and data sources that have been utilized in the referent development. For each source the next information is specified:

- *Source Type*. Specifies the type of the source. The template provides the option to select from a predefined source type set that can be expanded with other types when needed: *Fidelity Referent*, *Textbook*, *Journal/Conference Paper*, *Internal/External Report*, *Knowledge/Data-base*, *Internet Website*.
- *Full Reference*. Gives a traceable reference in order to retrieve the source. Could be a textbook reference, company details that owns a fidelity referent, etc. For references with a short life span such as Internet hyperlinks it is recommended to store the original source or copy in the archives of the referent managing organization.
- *Summary Origin and Contents*. This section briefly specifies the origin of the source and summaries what parts of the real-world are covered by the source. Helps tracing more details about certain parts of the specified real-world knowledge and selecting possible suitable knowledge sources when upgrading, reusing or developing new referent using current referent as a basis.
- *Quality Description and Rating*. This section describes and rates the quality level of the knowledge contained by the source that is used in the real-world knowledge specification part of the referent. The rating provides a handle for assessing the referent knowledge correctness and uncertainties in its development, validation and accreditation. Particularly, incase knowledge for the same part of reality from multiple sources are available and have to be combined into or selected as the real-world reference evidence. As stated by Oberkampf in relation to evidence theory, how to combine multiple sources, including SME real-world knowledge, is a not unique and an open issue with no single appropriate method for dealing with all possible application or problem domain specific knowledge [98]. More on this issue is presented in this and the next chapters. Like for the participant rating a nominal rating scale is used which looks as follows:

Named Score	Fuzzy Numerical Value
Excellent	*[100,80]*
High	*(80 60]*
Moderate	*(60, 40]*
Low	*(40, 20]*
Very Low	*(20, 0)*

Table 5-5 Referent Knowledge Source Quality Rating Scale

The next table gives two knowledge sources that are imparted in FASE referent and utilized in the development of the referent's real-world knowledge specification.

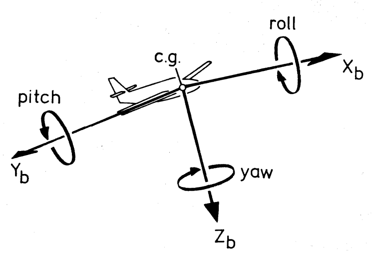
1 Knowledge or Data Source	
Type:	*Knowledge / Data-base*
Full Reference:	*Base of Aircraft Data (BADA) V3.1. Eurocontrol Experimental Centre. Bretigny-sur-Orge. France: 1998*
Quality Description:	*Standard Data-Base used by Eurocontrol and other well established and international ATC/ATM research institutes*
Quality Rating:	*High (equivalent value: 79)*
Summary Origin and Contents:	*Contains aircraft performance data and procedures for 151 aircraft types based upon actual pilot aircraft manuals*
2 Knowledge or Data Source	
Type:	*External/Internal Report*
Full Reference:	*An Object Oriented Analysis of Air-Traffic Control. Celesta Ball, Rebecca Kim. MITRE CAASD Corporation, McLean Virginia. August 1991*
Quality Description:	*Published Internal Report sponsored by FAA forms a base model for CAASD ATC/ATM Simulation Facilities*
Quality Rating:	*High (equivalent value: 70)*
Summary Origin and Contents:	*Provides a reference model for ATC experiment definition and simulation development for conventional and future airspace system research obtained by extensive domain analysis*
3 Knowledge or Data Source	
Type:	*Textbook*
Full Reference:	*Fundamentals of Air-Traffic control 3th Edtion. M. S. Nolan. Brooks/Cole Publishing Company 1999.*
Quality Description:	*Standard college level textbook used by various universities.*
Quality Rating:	*High (equivalent value: 60)*
Summary Origin and Contents:	*Provides a description of the history and current air-traffic control systems, procedures and regulations illustrated with real-life examples*

Table 5-6 Utilized Knowledge and Data Source Section Example

5.3.5 Utility Knowledge and Data Section

The last section of the referent additive knowledge specification part is the utility data section, which provides additional knowledge and data necessary to help understand the referent knowledge specification part. This section of the referent documents the definition and description for all axes systems referred in the next real-world knowledge specification section (Section 5.4). Understanding and transformations of axes systems are possible sources for additional errors in the real-world knowledge elicitation process and its usage during fidelity measurement. Therefore, they should be carefully be specified. A listing with used data conversion factors is also specified, since they can be a source for false gains in the accuracy of the specified real-world knowledge [112]. Furthermore, it also gives a notation and terminology lexicon. Any other utility knowledge and data can be added to this section when necessary. An excerpt from the FASE referent utility knowledge and data section is presented in *Table 5-7* below. In here underlining of text represents a link to another part of the referent.

Dimension Conversion Factor		
Dimension 1	>>- Conversion Factor ->>	Dimension 2
knots	0.514444444444	*m/s*
foot	0.3408	*m*
…	…	…

Axis System Specification		
Name	Description	Diagram
Aircraft Body Axes System	The origin of the system is at the aircraft center of gravity. The x-axis lies in the plane of symmetry of the aircraft and points out to the nose of the aircraft. The z-axis is perpendicular to the x-axis, lies in the plane of symmetry, and is directed downward for a normal flight attitude. The y-axis is directed out to the right wing of the aircraft. The body axes is fixed to the aircraft and oriented by reference to some geometrical datum (example flat-earth axis system). The rotational components about the x, y and z-axis are called roll, pitch and yaw, respectively. (Knowledge source: Elements of Airplane Performance)	
...	...	...

Notations and Terminology Lexicon	
Term or Symbol	Description
ADF	Automatic Direction Finder *(onboard aircraft avionics system for navigation)*
ADS-B	Automatic Dependant Surveillance – Broadcast *(onboard aircraft to aircraft and aircraft to ground data-link system for surveillance communication)*
HGL1	*Pressure Altitude Above the Ground*
...	...

Table 5-7 Utility Knowledge and Data Section Example

5.4 Fidelity Referent Knowledge-Base: Real-World Reference Knowledge Specification Templates

This second part of the unified fidelity framework referent knowledge-base template structure specifies the actual real-world reference knowledge (R_w^{ref}) to assess the fidelity of the simulation system. It is the here-specified real-world reference that is used by the fidelity evaluator function set $\tilde{C}_{\Delta_{RS}}$ to qualify or quantify the level of fidelity (Chapter 7).

The pillar of the real-world knowledge specification template is the treatment of the real-world as a system S^{rw} composed of a m-dimensional set of hierarchically related and interacting (sub)systems:

$$S^{rw} = \left\{ S_1^{rw}, S_2^{rw}, \ldots, S_m^{rw} \right\} \tag{5.1}$$

This means that any knowledge about this real-world system S^{rw} can be specified according the hierarchical object oriented system specification approach as discussed in Section 3.4. Furthermore, the associated composite-component system specification formalisms developed in that same section provide a well-structured and formal manner to specify this knowledge (Section 3.4.6). The advantage of such approach is that it facilitates a more objective and precise measurement of simulation fidelity. It also enables a better requirement definition and design of computer aided fidelity assessment tools. Therefore, the reference knowledge about S^{rw} is organized as a m-dimensional set of hierarchically ordered and interacting composite-component (sub)systems:

$$R_w^{ref} = \left\{ S_{ref_1}^{rw}, S_{ref_2}^{rw}, \ldots, S_{ref_m}^{rw} \right\} \tag{5.2}$$

In *Expression 5.2* $S^{rw}_{ref_i}$ is the referent system knowledge specification for the i^{th} real-world system $S^{rw}_i \in S^{rw}$. Following the system knowledge specification paradigms of Section 3.4, the knowledge specification for each $S^{rw}_{ref_i} \in R^{ref}_w$ can be broken down into two major interrelated elements:

- *Real-World System Structural Knowledge*
- *Real-World System Behavioral Knowledge*

Therefore the real-world reference knowledge specification part of referent template is structured according this subdivision. Both these elements will be discussed in detail in the subsequent sections. In these sections the hierarchical object oriented system specification approach will be further refined and expanded with additional specification elements and formalisms necessary for the referent template to properly facilitate simulation fidelity assessment throughout the simulation life cycle. In practice these two elements have to be implemented as a pair of interlinked knowledge-base sets.

5.4.1 Real-World Structural Composition and Relationships Section

This referent section focuses on the knowledge specification of the structural composition and relationships of the real-world system S^{rw}. Which comprises the specification of each (sub)system $S^{rw}_{ref_i}$ that populate R^{ref}_w in terms of their internal constitution, working, functional capabilities and, interaction and composite-component relationships with the other (sub)systems in R^{ref}_w. All real-world structural knowledge is organized in the unified fidelity referent template in three information areas: *Overall Structural Properties*, *Real-World System Topology Map* and *Real-World System Characteristics Description List*.

Overall Structural Properties

This template area gives an overall textual and qualitative description of the structural composition of R^{ref}_w in terms of its constituent (sub)system $S^{rw}_{ref_i}$, their interactions and properties. Furthermore, the knowledge sources used, general assumptions and simplifications made, and any other applicable boundary conditions placed upon its development can be specified here and/or otherwise links to must be included to the *elicitation process knowledge specification* (Section 5.5). Since R^{ref}_w is developed utilizing the composite-component system specification approach, at least the following types of cardinal numbers can be specified to quantitatively characterize the real-world structure set:

- $n\left(R^{ref}_w\right)_{system}$: The total number of (sub)systems $S^{rw}_{ref_i}$ that populate R^{ref}_w.
- $n\left(R^{ref}_w\right)_{leafs}$: The total number of *leafs* or sum of all (sub)systems $S^{rw}_{ref_i} \in R^{ref}_w$ for which yields that their component reference set $D = \varnothing$.
- $n\left(R^{ref}_w\right)_{forks}$: The total number of *forks* or sum of all (sub)systems $S^{rw}_{ref_i} \in R^{ref}_w$ for which yields that their component reference set $D \neq \varnothing$.

- $n\left(R_w^{ref}\right)_{interaction}$: The total number of interaction relationships between the sub-systems in R_w^{ref}, which is the sum of the interaction set I_d cardinality of each (sub)system $S_{ref_i}^{rw} \in R_w^{ref}$:

$$n\left(R_w^{ref}\right)_{interaction} = \sum n\left(I_d\right)_{system} \text{ for } \forall S_{ref_i}^{rw} \in R_w^{ref} \tag{5.3}$$

For every i^{th} leaf in the real-world composite-component system tree a so-called *branch length* (B_{length_i}) can be defined as the number of forks or composite parent systems in between the root and the leaf. The branch length can be used to assign the next two metrics to the real-world structural composition specification:

- $B_{length_{\max}}$: The maximum branch length, which is defined as follows:

$$B_{length_{\max}} = \max\left(B_{length_1}, \ldots, B_{length_m}\right) \text{ with } m = n\left(R_{ref}\right)_{leafs} \tag{5.4}$$

- $B_{length_{\max}}$: The average branch length, which is defined as follows:

$$B_{length_{ave}} = \left(\sum_{i=1}^{n\left(R_{ref}\right)_{leafs}} B_{length_i}\right) / n\left(\mathrm{R}_{ref}\right)_{leafs} \tag{5.5}$$

Together with the cardinal numbers these two metrics are high-level indicators for the scope, limitations and complexity of the specified real-world knowledge. As an example the overall structural properties of the *FASE referent* are given:

Overall Structural Properties	
Total NOF Subsystems	*4911*
Total NOF Leafs	*4726*
Total NOF Forks	*185*
Total NOF Interactions	*718386*
Maximum Branch Length	*5*
Average Branch Length	*3.1*

Table 5-8 Overall Structural Properties Specification Example

As can be seen from the table above the *FASE referent* is an extensive complex referent considering it large amounts of subsystems and their interactions. The knowledge of hundreds of navigation systems and 135 different aircraft types are the primary cause for this. The small average branch length and large number of leafs and indicate that the referent structure has a rather flat and wide structural hierarchy instead of a narrow and highly detailed sub-system specification. Also the maximum branch length is not significant larger than the average branch length. This indicates that there are no real-world area's that are of much more importance for the application domain of the referent than others. In this case the maximum branch length is caused by a more detailed sub-system specification of the B747-400 series aircraft necessary to achieve the application needs and its associated fidelity requirements for having a pilot-in-the loop large transport aircraft flight simulator available (Appendix C). For all other aircraft a far less detailed sub-system specification is sufficient. This because the performance characteristics of the aircraft as a whole predominate the actual airspace loads and flows. It are these loads and flows that are researched with this simulation system.

Real-World System Topology Map

The real-world system topology map is a graphical representation for the inner constitution or topology of the real-world structural hierarchy in terms of interaction and composite-component relationships between the (sub)systems $S_{ref_i}^{rw} \in R_w^{ref}$. Since the composite component system specification is based on the common known paradigm of object-orientation, the topology map presented here has many similarities with graphical object-oriented software engineering languages such as the Unified Modeling Language (UML) [92]. However, the topology map is tailored for the modeling and simulation application domain. The strength of the topology map is that it allows for quick visual inspection and interpretation, and navigation through the structural specification of the real-world (R_w^{ref}). Automated tool implementations could facilitate navigation by allowing the user to expand and hide systems decompositions, and thereby zooming in or out on specific parts of the structural hierarchy. Similarly, the detailed real-world knowledge about the system or interaction as discussed in the next paragraphs should be shown up when pointed to. To discuss and illustrate the symbolic elements of a real-world system topology map an excerpt of the *Airspace users* branch from the FASE referent topology map is presented (*Figure 5-2*). This branch specifies all civil aircraft and interactions within a civil airspace.

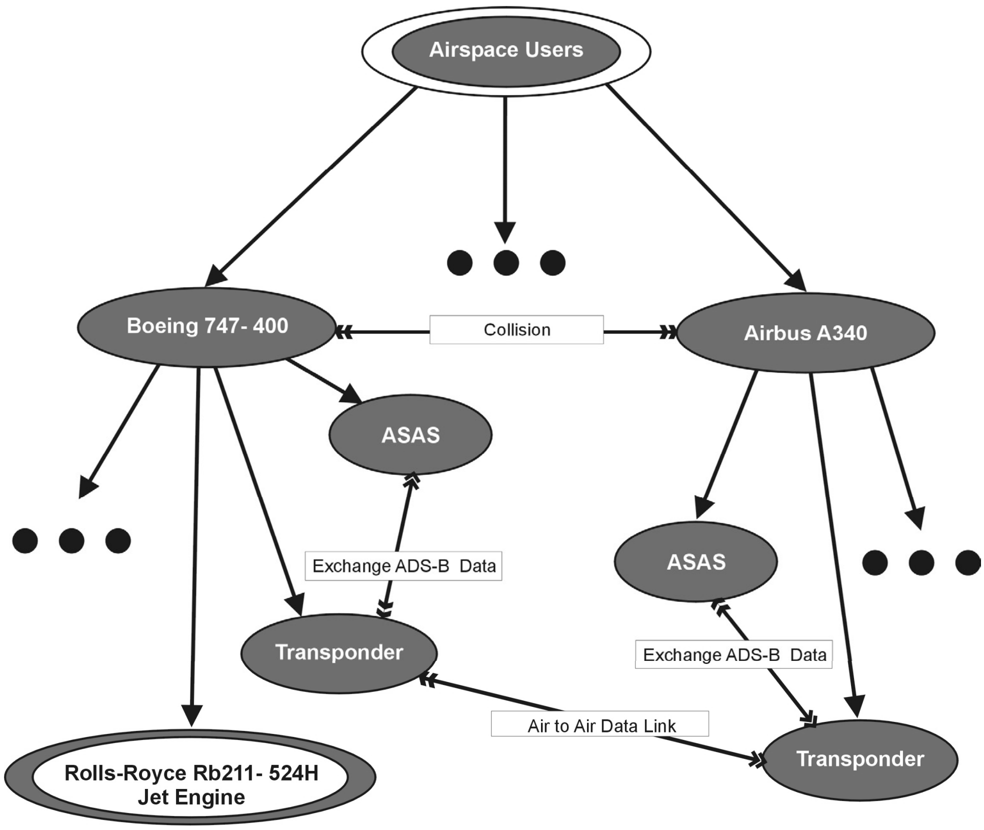


Figure 5-2 Excerpt FASE Referent: Real-World System Topology Map

In a real-world system topology map a single real-world system $S^{rw}_{ref_i}$ is represented by an ellipsoid. Each single system can be decomposed into a constituent set of interacting subsystems (Section 3.4). These subsystems can in turn be decomposed into a lower abstraction level of interacting systems as well. An arrow with a single arrowhead visualizes this composition relationship between two systems. For instance in *Figure 5-2* the *Boeing 747-400* is decomposed into an *ASAS*, *Transponder*, *Rolls-Royce jet-engine*, etc. The *Rolls-Royce jet-engine* system has also a lower-level topology map of its own outlining its subsystems like high-pressure turbine, combustion chamber etc. However, this underlying topology map is not shown in *Figure 5-2*. Furthermore, for brevity of this discussion the majority of all possible aircraft are omitted and replaced by three fat dots in this topology map example. A reference to a hidden lower-level topology map is indicated by a red ellipsoid with a blue border. The other way around a system that is part of a not shown system at a higher abstraction level, such as *Airspace Users*, is represented by a blue ellipsoid with a light gray border. Labeled arrows with double arrowheads are used to specify the interaction relationship and their direction between associated systems, such as the *Collision* interaction between the *B747-400* and *Airbus 340* aircraft. As can be seen in *Figure 5-2* interactions can be placed between any systems, in dependent of their position, in the topology map.

Real-World System Characteristics Description List

The real-world system characteristics description list is an array of detailed descriptions for the structural properties of each $S^{rw}_{ref_i} \in R^{ref}_{w}$. The internal system knowledge formalism given in Section 3.4.5 is used as a basis for this. For each system $S^{rw}_{ref_i}$ the following knowledge elements are specified:

- *Name*. A unique name to identify the specific system.
- *Reality Class*. Specifies whether the system belongs to *material* reality or *imaginary* reality. Which is a high-level indicator for the uncertainty level and correctness of system knowledge specification.
- *System Type*. Specifies to which type of systems this system belongs, giving an indication of the complexity nature of the system. For example, human system, human-made system or part of the atmosphere.
- *Behavioral Data List*. Provides references to all behavioral data available for this system in the referent behavior data specification section (Section 5.4.2).
- *Functional Capabilities List*. Description of all system's function capabilities in relationship to other systems in the topology map. These capabilities are specified according to the functional capability F_{cap} (*3.13*) and means-end hierarchy F_{cap_c} (*3.28*) specification formalisms as outlined in Section 3.4.
- *Representational and Behavioral Properties List*. This list describes all representational and behavioral properties of the system in terms of the internal system specification formalism parameter set P, state variable set Q and I/O variable sets U and Y. For each of these properties besides their name also a description of what the property stands for, mathematical symbol and where applicable the dimension is specified.
- *Topology Characteristics List*. Specifies the system's subsystem reference set D and also the reference to the system's parent system in the composite-component system hierarchy. In addition the set with the possible number of subsystem instances within

the real-world, called *Multiplicity*, are specified for all subsystems in *D* (*3.25*). The second part of this list is the detailed description of all the system's interactions with the other systems in the topology map. For each system interaction a textual description of the interaction mechanism and reference to the other system that is involved is given along with the observable properties (name, dimension, input or output identifier) associated with the interaction. These interaction descriptions are thus the practical implementation of the composite component system specification sets E_d and I_d (*3.26* and *3.27*).

Table 5-9 gives an excerpt of the real-world system characteristics description for the Boeing 747-400 as depicted in the FASE referent topology map example (*Figure 5-2*). Underlining of text again represents links to other referent sections.

Identification and General Properties	
Name:	*Boeing 747-400*
Reality Class:	*Material, ~~Imaginary~~*
System Type:	*Man-made system*
Behavioral Data List:	*Total Thrust Non-Causal Relationship, Phugoide Behavior Instance1, …*
Functional Capabilities	
…	
Functional Capability 1	
Goal:	*Safe and efficient air transportation of passenger and cargo between aerodromes*
Inputs:	*Number of Passengers, Cargo Payload, Wind Vector Field, …*
Outputs:	*Aircraft Position, Flight time, Total Fuel Consumption, …*
Sub Functional:	*F_{cap} Rolls-Royce RB211-524H Jet Engine, …*
Representational & Behavioral Properties	
…	
Property 1	
Name:	*Total Thrust Vector*
Description:	*The summated engine forces vector with respect to the aircraft body axes system*
Symbol:	T_{total}
Dimension:	*[N] (Newton Force)*
Type:	*Output, ~~Input~~, ~~Parameter~~, ~~State~~*
Topology Composite Component Characteristics	
Parent Name:	*Airspace User*
Sub System 1	
Name:	*Rolls-Royce RB211-524H Jet Engine*
Multiplicity:	*{4}*
…	
Topology Interactions Characteristics	
…	
Interaction 1	
Name:	*Atmospheric Wind Encounter*
Description:	*Wind causes a change in the aerodynamic forces and moments acting on the aircraft. This induces aircraft linear {xdot, ydot, zdot} and rotational {psidot, thetadot, phidot} accelerations*
Involved System:	*Windfield*
IO Variable 1:	*Wind vector magnitude {V_{wind}} (input, [knots])*
IO Variable 2:	*Position vector {x, y, z} (output, [m])*
…	…

Table 5-9 Excerpt Real-World System Characteristics Description for a B747-400

The identification, general and representational & behavioral properties descriptions of *Table 5-9* are self-explanatory. The *Rolls-Royce jet engine* is a subsystem of the *B747-400*, which has a multiplicity of 4 engine instances (*Figure 5-2*). The primary functional capability goal, like any other aircraft in this referent, is to transport passengers and cargo efficiently and safely between aerodromes. Both the associated input and output are the *B747-400* variables that characterize this function's realization. The *B747-400's Rolls-Royce jet engine* component functional capability of delivering most efficient engine thrust against specific fuel consumption contributes to the realization of this overall functional capability of the *B747-400*. Each interaction is referenced by its name in the related real-world system interaction chain description(s) as discussed in the next section on real-world behavioral data specification. In this interaction the wind field system's wind vector magnitude (input), which is encountered by the *B747-400* causes a change in the aircraft accelerations (state variables). In return this causes the airplane to move to another position (output) in the wind field resulting in another wind vector acting on the aircraft body.

5.4.2 Real-World Behavioral Data Section

This section of the referent specifies the various kinds of real-world behavioral knowledge associated with each of the real-world system $S_{ref_i}^{rw} \in R_w^{ref}$ and their interactions. Within this section the behavioral knowledge, where possible quantitatively or otherwise qualitatively, is specified for each system's observable property. In this specification four sets of behavioral knowledge areas are distinguished and listed: *real-world system interaction causality knowledge specification*, *non-causal behavior knowledge specification*, *qualitative behavior specification* and *behavior samples specification*. This classification is based on the work of Birta and Ozmizrak [8]. The behavioral knowledge of these four referent subsections are used to analyze how well the simulation system is capable of replicating this real-world system behavior during simulation execution under the same conditions as it has been collected or constructed.

Real-World System Interaction Causality Knowledge Specification

The idea for the interaction causality knowledge specification derives from Cockburn's use-case template approach and provides a qualitative means to specify the causal order of interactions over-time between the real-world systems $S_{ref_i}^{rw} \in R_w^{ref}$ [13]. In this way it enables a functional or process oriented manner to analyze the effects of missing system functional capabilities (F_{cap}) and interactions (E_d and I_d) on the overall simulated real-world behavior (Section 3.4). Therefore, in practice these specifications are developed from an analysis of the goals and functional capabilities of the overall real-world system and decomposition into its sub-functions. The set of all collected interaction mechanisms between the real-world systems $S_{ref_i}^{rw} \in R_w^{ref}$ define the *interaction causality knowledge set*, which is formally define as:

$$IC^{rw} = \left\{ IC_{chain_1}, IC_{chain_2}, ..., IC_{chain_n} \right\} \quad (5.6)$$

In *Expression 5.6* IC_{chain_i} is the i^{th} interaction mechanism or chain description that is defined as follows:

$$IC_{chain_i} = \left\langle \begin{array}{l} D_{seq}, I_{prim}, C_{pre}, C_{end}, E_{trig}, B_{seq}, \\ \left\{ FI_b \middle| b \in B_{seq} \right\}, \left\{ Con_b \middle| b \in B_{seq} \right\}, IB \end{array} \right\rangle \qquad (5.7)$$

In *expression 5.7* D_{seq} is the subset of real-world systems $S^{rw}_{ref_i} \in R^{ref}_{w}$ involved in the interaction chain. $I_{prim} = \left\{ I_{m_j} \middle| j \in T \right\}$ is the primary interaction chain, which is a linearly ordered set of interactions indexed by a time base *T*. Here $I_{m_j} = \left\langle S_m, R_m, f_{R_m} \right\rangle$ is the j^{th} interaction event exchanged containing the sending object $S_m \in D_{seq}$ and the receiving object $R_m \in D_{seq}$ for which the next condition should hold $S_m \notin R_m$. $f_{R_m} \in F_{cap\,R_m}$ is the invoked functional capability in the receiving object. C_{pre}, C_{end} and E_{trig} are respectively the sets of pre and end conditions proposition of this function interaction flow I_{prim}, and its triggering event. B_{seq} is the identifier set of all possible interaction chain branching variations and exceptions that could occur during the execution of the primary function interaction flow. $FI_b \subseteq IC_{list}$ is the indexed set containing all references to other interaction chains belonging to each branching variation. Con_b is the indexed set of all conditional propositions for the initiation of such a branching variation. $IB \subseteq I_{prim} \times Con_b$ represents the set of pairs that relates or links conditional proposition Con_b to a certain interaction event I_{m_j}. If $IB = \varnothing$ this specifies that an interaction flow doesn't have any branching variations.

Expression 5.6 is imparted in the unified fidelity framework referent template as physically linked list of specification templates for each interaction causality mechanism. This practical specification template for IC_{chain_i} contains the next fields:

- *Name*. Unique name to identify the specified interaction sequence.
- *Pre-Conditions*. Describes the necessary initial condition or states of the real-world systems at the start of the interaction mechanism.
- *End-Conditions*. Describes the final state or conditions of the real-world systems involved the interaction sequence.
- *Trigger Event*. The action starting the interaction sequence.
- *Primary Interaction Chain*. The main chain of interaction event exchanges between the systems from pre-condition to end-condition. The sequence lists the interactions in causal order and where possible specifies the duration and time intervals between the interactions.
- *Interaction Chain Variations*. Specifies any possible variation of the primary interaction chain.
- *Interaction Chain Exceptions*. Specifies the conditions that might stop the primary interaction chain completely or interrupt it and invoke a different interaction chain or overall real-world behavior.

- *Elicitation, Errors & Uncertainties*. Specifies interaction causality elicitation along with its associated errors and uncertainties and/or provides links to the referent *elicitation process knowledge specification* section (Section 5.5).

As an illustration consider the *Assure Safe Separation* function of the Airborne Separation Assurance System (ASAS) component each airspace user (*Figure 5-2*). This functional capability is realized in terms of a set of three sub-functions, which map to three system interaction chains. *Table 5-10* provides an excerpt of the interaction causality chain for the interactive behavior when a conflict should be resolved between two aircraft. The '=>' symbol in the variation and exception section indicates a reference to another branching detail interaction causality chain. Underlining of text represents a physical link to another referent part.

Interaction Chain Description		
Name	*Two Airspace User Separation Conflict Resolution (maps to the Assure Safe Separation Function) , id = 14*	
Pre Conditions	*Airspace users fly according flight-plan and track other users with an operational ASAS by transmission of ADS-B data through their transponders. Two airspace user's current tracks will cause a loss of safe separation within a certain time-span*	
End Conditions	*Both users have changed their original trajectory to their destination*	
Trigger Event	*One of the airspace user's ASAS detects the loss of safe separation with the other user*	
Primary Interaction Chain	Step	Interaction
	1	*Exchange ADS-B Data: transmit conflict notification & suggest resolution maneuver*
	…	…
	N	*Exchange ADS-B Data: Receive Maneuver is agreed and executed*
Variation	Step	Condition
	2	*Other airspace users transmit a conflict notification for the suggested resolution maneuver. => Multiple Airspace User Separation Conflict Resolution Chain (maps to the Assure Safe Separation Function)*
Exception	Step	Condition
	N	*No agreement on the resolution maneuvers can be achieved within a safe time span by both airspace users. => Air-Traffic Controller Separation Resolution Chain*
Error & Uncertainty	$P\left(IC_{chain_{14}}\right)=1$	

Table 5-10 Excerpt FASE Interaction Causality Description List

Non-casual behavior knowledge specification

Within the real-world there exist known logical relationships for and between the values of the system input (*U*), output (*Y*), state (*Q*) and parameter (*P*) variables. Relationships that always must hold for any behavior instance of a system. These relationships are called *non-causal relationships* [8]. Within the fidelity referent a distinction is made between following non-causal relationship types:

- *System variable range set* specifies the ranges for *U*, *Y*, *Q* and *P* of each system $S_{ref_i}^{rw} \in R_w^{ref}$. This set is formally defined as:

$$\bigcup_{i=1}^{n\left(R_w^{ref}\right)_{system}} \left\{ range(U), range(Y), range(Q), range(P) \middle| U \in S_{ref_i}^{rw} \wedge Y \in S_{ref_i}^{rw} \wedge Q \in S_{ref_i}^{rw} \wedge P \in S_{ref_i}^{rw} \wedge S_{ref_i}^{rw} \in R_w^{ref} \right\}_{S_{ref_i}^{rw}} \quad (5.8)$$

The range can define by a single value, a set of possible values for a discrete variable or a closed interval over $\Re$ for a continuous variable (Section 3.4.4). If the system variables elements are complex n-dimensional spaces, distributions and fields the range defines both the boundaries of the closed space over $\Re^n$ as well as the closed interval over $\Re$ for the associated variables. For n-dimensional parameter look-up tables range yields the set of possible axis values as well as the set of the associated grid-points.

- *System aggregate variable relationships set* specifies for each parent system $S^{rw}_{ref_i} \in R^{ref}_w$ the possible relationship between a set of subsystem variables values and some of $S^{rw}_{ref_i}$ its input or output variable(s). This set is formally defined as:

$$\bigcup_{i=1}^{n\left(R^{ref}_w\right)_{forks}} \left\{RL_{aggregate}\left(...\right)_1,...,RL_{aggregate}\left(...\right)_m\right\}_{S^{rw}_{ref_i}} \text{ with } S^{rw}_{ref_i} \in R^{ref}_w \qquad (5.9)$$

Where $RL_{aggregate}(...)_m$ is the m^{th} aggregate variable relationship of $S^{rw}_{ref_i}$, which is defined by the next Boolean expression:

$$RL_{aggregate}\left(t,U_c,Y_c,\left\{Y_d \middle| Y_d \in S^{rw}_{ref_d} \wedge d \in D\right\}\right)_m \qquad (5.10)$$

Inhere is $t \in T$, $U_c \in S^{rw}_{ref_i}$, $Y_c \in S^{rw}_{ref_i}$ and $D \in S^{rw}_{ref_i}$. Furthermore, $S^{rw}_{ref_d} \in R^{ref}_w$ is subsystem d of $S^{rw}_{ref_i}$.

- *System variable interrelationships set* specifies any non-causal relationship between U, Y, Q and P values of $S^{rw}_{ref_i} \in R_{ref}$. This set is formally defined as:

$$\bigcup_{i=1}^{n\left(R^{ref}_w\right)_{system}} \left\{RL_{system}\left(...\right)_1,...,RL_{system}\left(...\right)_k\right\}_{S^{rw}_{ref_i}} \text{ with } S^{rw}_{ref_i} \in R^{ref}_w \qquad (5.11)$$

Where $R_{bsystem}(...)_k$ is the k^{th} system variable interrelationship of $S^{rw}_{ref_i}$, which is defined by the next Boolean expression:

$$RL_{system}\left(t,U,Y,Q,P\right)_k \qquad (5.12)$$

In this Boolean expression is $t \in T$, $U \in S^{rw}_{ref_i}$, $Y \in S^{rw}_{ref_i}$, $Q \in S^{rw}_{ref_i}$ and $P \in S^{rw}_{ref_i}$. This expression and Boolean expression 5.10 can be simple and logical mathematical relationships but also more complex polynomial descriptions of relationships measured in the real-world including some correlation relationships. An example is a n-dimensional polynomial description of a aerodynamic lift-drag polar as function aircraft mach number and given configuration.

As in all previous sections, a template is used to specify and utilize this referent non-causal behavior knowledge in practice for each $S_{ref_i}^{rw} \in R_w^{ref}$. Besides the above behavior knowledge also the following items are specified for each non-causal relationship: *knowledge source references*, *data generation process* and *know* e*rror & uncertainty descriptions*. These additional items help track the source and assess the quality and reliability of the specified behavior knowledge.

The next excerpt from the FASE fidelity referent provides examples for each non-causal relationship categories (*Table 5-11*). Here underlining of text represents an actual link to other knowledge sections of the fidelity referent. The '->' symbol is used in the template to point to a list of subsystem variables part of a system aggregate variable relationship.

Non-Causal Behavior Knowledge Specification			
…			
Real-World System 1			
Name System:	Boeing B747-400		
System Variable Range Specification			
Name	Range	Data Source & Generation	Uncertainty & Error
V_{ias}	[0, 507]	*B747-400 Manual: Constructed from assumption no negative airspeed possible and aircraft Vmo*	*P(range(V_{tas})) = 1*
M_{tot}	[403599, 870000]	*B747-400 Manual: Constructed from aircraft empty and Mto weight*	*P(range(M_{tot})) = 1*
Cl_{basic}	table_clbasic.xml	*B747-200 NAS:A Derived from wind tunnel experiments*	*Aero Error & Uncertainty Spec*
…	…	…	…
System Aggregate Variable Relationships			
…			
Relationship 1			
System Variable:	M_{tot}		
Sub-System Variable:	Fuel System->M_{feul}, Airframe Structure->$M_{structure}$, Jet Engine->M_{engine}		
Boolean Expression:	$M_{tot} = M_{feul} + M_{structure} + 4\ M_{engine}$		
Data Source & Generation	*Basic Mechanical Engineering Principle: SME- ID3*		
Uncertainty & Error	$P\left(RL_{aggregate}\left(\mathrm{M}_{tot} = \mathrm{M}_{feul} + \mathrm{M}_{structure} + 4\ \mathrm{M}_{engine}\right)_1\right) = 1$		
…			
System Variable Interrelationships			
…			
Relationship 1			
System Variable:	$H_{terrain}$, H_{cg}		
Boolean Expression:	$H_{cg} \geq H_{terrain}$		
Data Source & Generation:	*Common SME knowledge: Impossible to fly under the ground*		
Uncertainty & Error:	$P\left(RL_{system}\left(\mathrm{H}_{cg} \geq \mathrm{H}_{terrain}\right)_1\right) = 1$		
Relationship 2			
System Variable:	$\vec{V}_{wind}, \vec{V}_{tas}, \vec{V}_{gs}$		
Boolean Expression:	$\vec{V}_{gs} = \vec{V}_{wind} + \vec{V}_{tas}$		
Data Source & Generation:	*From: Elements of Airplane Performance*		
Uncertainty & Error:	$P\left(RL_{aggregate}\left(\vec{V}_{gs} = \vec{V}_{wind} + \vec{V}_{tas}\right)_2\right) = 1$		

Table 5-11 Excerpt FASE Non-Causal Behavior Knowledge Specification

Behavior Samples Specification

Real-world system behavior samples B_{int} and B_{ext} specify respectively the external and internal system behavior knowledge in terms of a set of registered system variable trajectories or behavior instances (Section 3.4). These behavior samples provide a basis for the quantitative fidelity measurement aspect of the whole simulation fidelity assessment process. A behavior instance can originate from direct observation of a real-world system S_i^{rw} and/or are artificially generated by other simulations of the same system. There exist a whole range of different experimental frames or conditions for which system behavior instances can be registered. The possible range of experimental frames depends on the complexity of the system in terms of all conceivable combinations of initial system state, start time, parameter settings and system input.

Behavior sample based fidelity measurement is best performed when the behavior instances for the same real-world system S_i^{rw} in R_w^{ref} and S_{appx} have been observed under similar conditions and have comparable registrations of elements from *U, P, Q* and *Y* on a suitable time-base *T* (sample rates and aliasing effects) to avoid misalignment errors and uncertainties in the measurements. However, for reason discussed in Chapter 4 it is hard in real-life to exercise fully control over all internal (*P* and *Q*) and external (*U* and other unforeseen disturbances) system conditions. Similarly, it is not always possible to observe and record the whole real-world system and its interacting environment. This implies that real-world system behavior instances cannot always be elicited under the desired conditions or do not contain all the information that are of interest for the simulation application at hand. In such cases one has to rely on estimation techniques such as assumptions, extrapolation and interpolation to properly compare real-world and simulated behavior samples. Any *controllability* and *observability* limitations encountered during the observation or experimentation of behavior instances from the actual real-world and simulation system are thus potential sources for errors and uncertainty in simulation fidelity assessment. Therefore, the fidelity referent behavior samples specification requires that any information on their elicitation and generation process must be carefully documented along with quantitative or qualitative specification of error and uncertainty when possible. This can be done either directly in the template or using link to the elicitation process specification incase a more thorough specification is needed or available (Section 5.5).

Real-world system internal and external behavior samples have already been formally defined in Section 3.4.5 by expressions (*3.22*) and (*3.23*). Before moving on to the discussion of the fidelity referent behavior sample specification template a derived behavioral sample called *complex behavior sample* will be discussed first. In the real-world there exist many systems that are of stochastic nature. It is a well known fact that for such systems separate behavior instances are insufficient to fully characterize this kind of system behavior [8] [21] [47] [66] [80] [95] [129]. To properly characterize stochastic system behavior, a set of correlated behavior instances must be elicited and processed to provide estimates for certain stochastic variability quantifiers (average, variance, probability, etc.). Furthermore, in frequency domain analysis a spectrum of input signals is fed to a system to specify behaviors in terms of characteristic quantities, which directly relate to or can be derived from a certain set of behavior samples.

The mapping from a set of internal or external behavior instances into a set of derived behavior quantifiers for a real-world system is referred here as a *complex behavior* *sample*. Complex behavior samples for both internal and external system behavior are formally defined as follows:

$$\tilde{B}_{in} : (\omega, q)_1 \times (\omega, q)_2 \times ... \rightarrow Q_{complex} \tag{5.13}$$

$$\tilde{B}_{ext} : (\omega, \rho)_1 \times (\omega, \rho)_2 \times ... \rightarrow Y_{complex} \tag{5.14}$$

In (*5.13*) and (*5.14*) $Q_{complex}$ and $Y_{complex}$ are the resulting complex behavior quantifier sets. The exact structure and contents of these two sets depend on the real-world system and its behavior under consideration. Furthermore, $(\omega, q) \in B_{in}$ and $(\omega, \rho) \in B_{ext}$.

The union of all behavior samples of each real-world system $S^{rw}_{ref_i} \in R^{ref}_w$ form the fidelity referent behavior samples specification section and is formally defined as:

$$\bigcup_{k=1}^{n\left(R^{ref}_w\right)_{system}} \left\{B_{in}, B_{ext}, \bigcup \tilde{B}_{in}, \bigcup \tilde{B}_{ext}\right\}_{S^{rw}_{ref_i}} \tag{5.15}$$

The fidelity referent template that implements this union of behavior samples comprises the next elements to specify each internal or external behavior instance:

- *Observation Frame Name*. Unique name to identify the observation frame in which this behavior instance is collected.
- *Data Source Reference*. Specifies a reference to the data source in the referent 'Utilized knowledge and data source section' from which the behavior instance is collected. This helps to assess the reliability of the behavior instance and when necessary provides the opportunity to retrieve any additional information about the behavior instance and its collection process.
- *Data Collection Process Description*. Gives a description of how the data is collected and processed to generate the specified behavior sample. In other words it specifies whether the behavior sample is deduced from experimental observation with the actual real-world system S^{rw}_i or from another source such as simulation-generated data. Again this provides indicators for the applicability, quality and reliability of the behavior instance.
- *Real-World Initial and Boundary Conditions Description*. This part specifies the real-world system initial conditions ($p_1 \in P \wedge q_1 \in Q$) at $t_1 \in T$ for which the behavior instance holds. As discussed in Section 3.4 a different initial condition yields a different behavior instance. Any other boundary conditions such as specific assumptions made and issues regarding the controllability and observability of the real-world system or its environment in relationship to error and uncertainties should be specified here. This contains all information necessary to replicate the behavior instance by the simulation system as good as possible as well as for the assessment of effect of any errors and uncertainties that affect fidelity measurement reliability and credibility.
- *Real-World System Context*. References the other real-world systems from the topology map that were involved or interacting with the system for which this

behavior instance is specified. This gives an indication of the context in which the real-world system has been observed such as in a fully operational context or just as a stand-alone system. Plus it provides the links to the behavior instances of other real-world systems that were registered in conjunction with this behavior instance.

- *Observation Time Base*. Specifies the date, start times and end-times of the behavior instance time base (T).
- *Input Segment Specification*. Lists the registered elements of the applied input segment $\omega \in (T, U)$. For each recorded input variable its name, dimension, observation sample rate, known error and uncertainties are specified. Since the measured trajectories can be expressed in various formats (tables, figures, etc.), be store on different mediums (paper, electronic data-base, etc.) and could be very large, only a reference to the associated data source entry is specified instead of the trajectory values itself.
- *Output and State Segments Specification*. This section provides either the registered output segment $\rho \in (T, Y)$ or state segment $q \in (T, Q)$ resulting form the previously specified input segment. These segments are specified in the same manner as the input segment.

The complex behavior sample specification template contains the same elements as for the internal and external behavior instances. Instead of the input, output and state segment specification the next elements are specified for a complex behavior sample:

- *Behavior Instances List*. References the set of internal or external behavior samples in R_w^{ref}, which are used in the specific complex behavior mapping (*5.11* or *5.12*).
- *Complex Output and State Specification*. Here the complex output ($Y_{complex}$) or state ($Q_{complex}$), which result from the complex behavior mapping are specified. Depending on the complexity and size of both images its outcomes are directly specified in this referent section (name, dimension and value) or otherwise a reference to the belonging data source is provided.

The (in)formal complex behavior mapping description itself should be imparted in the 'Data Collection Process Description' part of the complex behavior sample specification template and linked to the elicitation process specification part of the fidelity referent (Section 5.5).

To illustrate both types of referent behavior sample specification consider *Table 5-12*, containing an excerpt from the CN235 Simulator Project (Appendix C).

Behavior Samples Specification	
…	
Behavior Sample 1	
Name System:	*CN235-220C*
Behavior Instances	
…	
Instance 1	
Frame Name:	*Ops312a.1 (Normal Take-Off)*
Data Source:	*CN235_OPT312S00R01.XLS on manufacture flight test data cd-rom*
Data Collection Process:	*See: Ops series Data Collection Process Specification Section*

Real-World Initial and Boundary Conditions			
Controllability & Observability Issues Error & Uncertainty:	*See: Ops series Data Collection Process Specification Section*		
Variable	Dimension	Value	Error & Uncertainty
THETA	*degrees*	*-0.0035*	*Unknown sensor error*
FLAP POS	*degrees*	*10*	*P(FLAP POS = 10) =1*

…

Real world System Context	
Real-World Sys 1:	*Atmosphere -> Air-Data and Wind Field*
Real-World Sys 2:	*Earth -> Runway and Earth Acceleration*
Observation Time Base	
Date:	*June 19th 1998*
Start Time:	*06:09:13.048*
End Time:	*06:10:23.503*

Input Segment Specification				
Variable	Dimension	Sample Rate	Significant Digits	Error & Uncertainty
DELTAL	*degrees*	*16 Hz*	**.0000*	*Unknown sensor error*
DELTR	*degrees*	*16 Hz*	**.0000*	*Unknown sensor error*

…

Output and State Segment Specification				
Variable	Dimension	Sample Rate	Significant Digits	Error & Uncertainty
VTAS	*knts*	*16 Hz*	**.0000*	*Unknown sensor error*
HGL1	*ft*	*16 Hz*	**.0000*	*Unknown sensor error*

…

Complex Behavior Sample 1	
Frame Name:	*Cross Wind Take-Off*
Data Source:	*Manufacture flight test data cd-rom*
Data Collection Process:	*See: Ops series Data Collection Process Specification Section*
Real-World Initial and Boundary Conditions	
Controllability & Observability Issues, Error & Uncertainty:	*See: Ops series Data Collection Process Specification Section*
Behavior Instance List	
Instance 1:	*Ops165a.2*
Instance 2:	*Ops165a.4*
Instance 3:	*Ops166a.2*
Instance 4:	*Ops166a.4*
Instance 5:	*Ops166a.6*

Complex Output and State Segment Specification			
Variable	Dimension	Value	Error & Uncertainty
AVERAGE TAKEOFF-TIME	*sec*	*20.3*	*Standard Dev: 2.3*

…

Table 5-12 Excerpt CN235 Simulator Referent Behavior Samples Specification

Qualitative Behavior Specification

As previously discussed it is not always possible to collect all behavior instances for every real-world system. However, given a specific observation frame(s) it is usually possible to qualitatively specify existing causal relationships between system variables over a period of time. Often it is also very well possible to observe or describe certain general trends in system behavior that are expected or must hold for certain changes in system variables over a period of time. These kinds of relationships are called by Birta *ordinary* and *change-in-value causal relationship* respectively [8]. Although these qualitative relationships provide a more course grained behavior specification than behavior samples, they do have definitely utility in fidelity assessment not only when actual behavior samples are unavailable. Especially during early simulation system development they will help to detect large fidelity discrepancies in the simulation model before more complex and time-consuming quantitative fidelity assessment methods based on behavior instances are applied. Following Birta's proposed notation the i^{th} ordinary causal relationship of a real-world system can formally be defined as:

$$RL_k^{oc}\left(t,U,Q,P,Y\right)_i \rightarrow RL_k^{oe}\left(t,U,Q,P,Y\right)_i \qquad (5.16)$$

Here RL_k^{oc} and RL_k^{oe} are Boolean expressions, which describe a certain real-world system behavioral condition. In RL_k^{oc} and RL_k^{oe} $t \in T$, $U \in S_{ref_i}^{rw}$, $Y \in S_{ref_i}^{rw}$, $Q \in S_{ref_i}^{rw}$ and $P \in S_{ref_i}^{rw}$ for $S_{ref_i}^{rw} \in R_w^{ref}$. If RL_k^{oc} holds it will result in the behavior effect RL_k^{oe}. Similarly change-in-value causal relationships are formally defined for changes in a real-world system its j^{th} parameter or l^{th} input variable as follows:

$$RL_k^{cvc}\left(t, p_j, p_j'\right) \rightarrow RL_k^{cve}\left(t,Y,Y'\right)_{p_j} \qquad (5.17)$$

$$RL_k^{cvc}\left(t, u_l, u_l'\right) \rightarrow RL_k^{cve}\left(t,Y,Y'\right)_{u_l} \qquad (5.18)$$

Where RL_k^{cvc} is Boolean expression, which either describes a change in value of a system parameter from p_j to p_j' over a period of time while the other system conditions remain the same with $p_j, p_j' \in P$. Like wise the same holds for a system input variable change from u_l to u_l' with $u_l, u_l' \in U$. If RL_k^{cvc} occurs the value change of system output vector from Y to Y' over time satisfies the Boolean expression RL_k^{cve}. The union of all aforementioned causal relationships of a real-world system forms the qualitative behavior reference knowledge of each $S_{ref_i}^{rw} \in R_w^{ref}$. All qualitative behavior references together form the fidelity referent qualitative behavior specification section. This section can now formally defined as follows:

$$\bigcup_{k=1}^{n\left(R_w^{ref}\right)_{system}} \left\{ \bigcup \left\{ RL_k^{oc}\left(..\right)_1 \rightarrow RL_k^{oe}\left(..\right)_1, .. \right\}, \bigcup \left\{ RL_k^{cvc}\left(..\right)_1 \rightarrow RL_k^{cve}\left(..\right)_1, .. \right\} \right\}_{S_{ref_k}} \qquad (5.19)$$

The practical fidelity referent template implementation of (*5.17*) is shown in *Table 5-13*. For each system variable involved in a relationship a reference to the belonging real-world system characteristics description section is included here. Each formal specification of a qualitative relationship is accompanied by textual description for clarity reasons. A specification and/or reference to data source and the elicitation process are provided to help track the origin and assess the reliability of the relationship specified. Furthermore, when possible information on the know error and uncertainty must be stated and/or referenced (Section 5.5.2). In *Table 5-13* excerpts from the CN235 simulation project are presented to provide some practical examples of both aforementioned qualitative causal relationships.

Qualitative Behavior Knowledge Specification

…

Real-World System 1

Name System:	CN235-220C

Ordinary Causal Relationships

…

Relationship 1

Observation Frame:	*Yield for every observation frame*
System Variables:	M_{fuel}, H_{cg}, $H_{terrain}$, V_{gs}
Formal Specification:	$\left(M_{fuel} = 0\right) \wedge \left(H_{cg} \geq H_{terrain}\right) \rightarrow \left(H_{cg} = H_{terrain}\right) \wedge \left(V_{gs} = 0\right)$
Textual Description:	*If the aircraft is flying above the ground and its fuel mass becomes zero then the aircraft ground speed should reduce to zero and the aircraft altitude will become equal to the terrain elevation.*
Data Source & Generation:	*Mechanical Engineering Principle: Without energy injection the aircraft kinetic and potential energy will be lost due to aerodynamic friction*
Error & Uncertainty:	$P\left(RL_1^{oc}\right) = 1$

Relationship 2

Observation Frame:	*Take Off*, *Climb*, *Cruise*, *Descent*, *Landing*
System Variables:	V_{tas}, V_{stall}, dH_{cg}/dt
Formal Specification:	$V_{tas} \leq V_{stall} \rightarrow dH_{cg} / dt < 0$
Textual Description:	*If the aircraft true airspeed is below the stall speed for any aircraft configuration then the aircraft altitude will decrease.*
Data Source & Generation:	*Common Aerospace Engineering Knowledge: SME_ID2*
Error & Uncertainty:	$P\left(RL_2^{oc}\right) = 1$

…

Change in Value Relationships

…

Relationship 1

Observation Frame:	*Cruise*
System Variable:	PLA, T_{eng}, V_{tas}, H_{cg}
Formal Specification:	$PLA' > PLA \rightarrow \left(T_{eng}' > T_{eng}\right) \wedge \left(V_{tas}' > V_{tas}\right) \wedge \left(H_{cg}' > H_{cg}\right)$
Description:	*If from a steady straight level flight the power lever angle increases then the engine thrust increases, which causes a new steady state straight flight with both an increased true airspeed and altitude. Altitude increase is due to a temporary instantaneous increase of flight path angle (T-D/W>0)*
Data Source & Generation:	*From: Elements of Airplane Performance*
Error & Uncertainty:	$P\left(RL_1^{cvc}\right) = 1$

Relationship 2

Observation Frame:	*Take Off*, *Climb*, *Cruise*, *Descent*, *Landing*
System Variable:	PLA, CLA, TQ_{eng}, N_p
Formal Specification:	$\left(PLA' = PLA\right) \wedge \left(CLA' > CLA\right) \rightarrow \left(TQ_{eng}' < TQ_{eng}\right) \wedge \left(N_p' > N_p\right)$
Description:	*If in any normal flight condition the condition lever angle is increased while the power lever position isn't changed the engine torque will decrease and the propeller rpm increases*
Data Source & Generation:	*From: CN235-220 Aircraft Operation Manual*
Error & Uncertainty:	$P\left(RL_2^{cvc}\right) = 1$

Relationship 3	
Observation Frame:	Climb, Cruise, Descent
System Variable:	Iyy, θ
Formal Specification:	$I_y' > I_y \rightarrow (\omega_\theta' < \omega_\theta) \wedge (\zeta_\theta' < \zeta_\theta)$ with $\tilde{B}_{phugoid} : (\omega, \rho)_{ext_{phugoid}} \rightarrow \{(\omega_\theta, \zeta_\theta) \mid \omega_\theta \in \Re \wedge \zeta_\theta \in \Re \wedge \theta \in (T, Y)\}$
Description:	*If aircraft inertia moment around the aircraft body y-axes increases then the period and damping of the phugoid motion of theta decreases*
Data Source & Generation:	*From: Lecture Notes Aircraft Stability and Control 2*
Error & Uncertainty:	$P(RL_3^{cvc}) = 1$
	…

Table 5-13 Excerpt CN235 Qualitative Behavior Knowledge Specification

5.5 Fidelity Referent Knowledge-Base: Elicitation Process Knowledge Specification Templates

The third and last part of the fidelity referent focuses on the specification of the elicitation process (Section 5.5.1) and error/uncertainties (Section 5.5.2) of the reference knowledge specified in the preceding section. As discussed previously (Chapter 4), availability of such knowledge is mandatory for the correct application of the fidelity referent and the interpretation or credibility judgment of the fidelity measurements based on the referent.

5.5.1 Elicitation Activities and Constraints Specification

Careful specification of the elicitation activities E_{rw} and their associated constraints set L_R^{elicit} provide quality indicators for the whole fidelity referent elicitation process, formally defined by *Expression 4.10*, and its product the real-world reference knowledge R_w^{ref} (Sections 4.4.2 and 5.4). Quality in this regard yields possible mistakes, error and uncertainty sources, systematic errors, elicitation thoroughness, usages of a standard or accepted procedures and appropriateness of the real-world knowledge for a certain application or fidelity measurement. Furthermore, the specification of the conducted elicitation activities enhances the traceability of the real-world knowledge and its stability.

The elicitation process activity specification comprises a high level description of the conducted elicitation activities, their relationships and general constraints (time, resources, etc) encountered during the development of the real-world reference knowledge. This description is completed with a structured set of the conducted activities or experiments. For each elicitation activity or experiment the following elements are specified (when applicable):

- *General Process Description*. A textual description how and what real-world knowledge has been collected, measured, pre-processed and added in the reference knowledge specification template (Section 5.4).

- *People Involved*. Listing of all individuals involved in this activity, which yields a series of participant ID links to *Developer and Validation Information Section* (Section 5.3.3).
- *Knowledge Sources Used*. Listing of all knowledge sources involved in this activity, which yields a series of links to the *Utilized Knowledge and Data Sources Section* (Section 5.3.5).
- *Real-World Reference Knowledge Part Coverage*. This element specifies or references those part(s) of the fidelity referent *Real World Knowledge Specification Section* (Section 5.4) that are the result of this elicitation activity.
- *Knowledge Collection*. Detailed description how the real world knowledge is collected. In here the following issues, when applicable, must be addressed: external influences (importance and magnitude), observation frame (definition, controllability, observability), initial conditions, boundary conditions and constraints, impact of collection process on the outcomes and the impact of the individuals involved (magnitude, experience, relevance).
- *Knowledge Preprocessing*. Detailed description of the methods used to pre-process the collected reference knowledge. In here the following issues, when applicable, must be addressed: possible introduction and propagation of error and uncertainty by the method itself, experience of the individuals involved, impact of assumptions and decisions made during preprocessing on the outcome.
- *Used Equipment*. A description of the possibly used (measurement) equipment in the knowledge collection and preprocessing that may affect the output. In here the following issues, when applicable and relevant, must be addressed with respect to the equipment performance: equipment precision (computational, measured, etc), bias and error, detectable changes, hysteresis and detection limit (upper and lower bound).

The table (*Table 5-14*) below gives a practical illustration of the elicitation activities and constraints specification. This illustration is an excerpt derived from the CN235 simulator project. Underlining again indicates a physical links or reference to other parts of the fidelity referent where additional information is specified.

Elicitation Activities and Constraints Specification	
Overall Elicitation Activities Description	*The real-world knowledge base is populated by the DUT team by means of cataloging, reviewing and merging knowledge that is made available to DUT by IAe. This requires close cooperation and communication with IAe team, to obtain appropriate data and solve any encountered issues (missing, incorrect, conflicting knowledge). Basically IAe is the major knowledge source mediator……..*
Overall Elicitation Constraints	*Since the level of indirection in obtaining real-world knowledge through IAe team, one is depending on IAe and what they make available or limited due to security reasons. Furthermore, expect communication misinterpretation and delays of knowledge. Real aircraft is not available for additional flight-testing when………*
Elicitation Activity Descriptions	
...	
Elicitation Activity 1	
Process Description	*Ops series Data Collection Process*
People Involved	
ID Participant:	*Participant_001*
ID Participant:	*Participant_005*
Knowledge and Data Sources Used	
Source:	*IAe CD-ROM: Flight Test Data Ops Series*
Source:	IAe Report:
Source:	IAe Report:

Source:	...
Referent Part Coverage	
Part:	*Behavior Sample Section: Long Handling Qualities, Short Period*
Part:	*Behavior Sample Section: Long Handling Qualities, Phugoid*
Part:	...
Knowledge Collection	
Observation frame Definition:	*The behavior samples have been collected according the FAA aircraft certification regulation in the context of the CN235-330M certification test program by a specifically internationally assigned flight test-team. For this purpose a real CN235-220M test aircraft has been instrumented…*
External Influences:	*None*
Controllability Issues:	*Local weather conditions are not controllable*
Observability Issues:	*Exact and current local weather conditions are not observable but general conditions are known.*
Initial Condition Issues:	*For the next test runs the moments of inertia are missing: Ops…..*
Boundary Condition Issues:	*None*
Constraints and Limitations:	*Behavior sample registrations have originally been performed for aircraft certification purposes instead of simulator validation. As a result not all available behavior samples required are available or for limited flight conditions or some essential variables are missing………*
Collection Process Issues:	*The sensors have been sampled at a rather low rate of 4 Hz….*

Knowledge Preprocessing Methods

Name	Description	Error & Uncertainty	Assumption
Interpolation	*A linear interpolation has been applied by IAe prior to delivering data to TUD to increase the behavior sample rate from 4 Hz to 16Hz*	*Error made is of second order*	*Smooth signal with low noise and frequency contents*
...	...	...	...

Used Equipment

Type or Name	Precision	Bias & Error	Hysteresis	Detection Limits
FORS 6 rate sensor	*0.01 deg/s*	*0.005 deg/s*	*Not Available*	*[-35.41 deg/s , +37.31 deg/s]*
...	...	...	...	...

...

Table 5-14 Excerpt CN235 Elicitation Activities and Constraints Specification

5.5.2 Referent Error & Uncertainty Specification

For various reasons discussed in the previous chapters, elicitation of real-world knowledge is an inherently erroneous and uncertain activity. Which means that the real-world reference knowledge specification R_w^{ref} (Section 5.4), no matter how precisely elicited has the possibility errors. These errors can either (partially) be known δR_w^{ref} (*Expression 4.11*) or (partially) unknown resulting in uncertainties $U_\delta\left(R_w^{ref}\right)$ (*Expression 4.12*). As formally specified by *Expressions 4.21* and *4.22* these reference knowledge errors and uncertainties contribute to the overall error and uncertainty of the practical simulation fidelity specification $F_{practical}$ (*Expression 4.23*). Therefore proper specification of these errors and uncertainties is an essential M&S engineering practice and an unavoidable part of fidelity assessment. However, literature shows that like fidelity itself the assessment of uncertainty in modeling and simulation has a long way to go but is gaining growing interest and research activities [30] [95] [96] [115] [142].

The major issue is that specification of errors and particularly uncertainty in modeling and simulation is neither a routine task nor a pure mathematical one. It depends on detailed knowledge of both the nature of the simuland and the simulation system, plus the measurement methods and procedures used. The utility of the error and uncertainty quoted for the results of simulation fidelity assessment therefore ultimately depend on the understanding, critical analysis and integrity of those who contribute to its quantification and qualification. Besides qualitative judgment there exist mathematical techniques that are adequate or promising for (at least some) parts of error/uncertainty assessment and specification include. These techniques that can be divided in either probabilistic (frequentist) and non-probabilistic techniques include: approximation theory, probability and stochastic theory, Bayesian theory, evidence theory, fuzzy set/logic theory, neural networks etc [30] [31] [67] [96] [97] [98] [115] [142] [146]. Although important a thorough discussion and research to their application is considered beyond the scope of this dissertation. Therefore, appendix B contains a summary of a selection of these possible mathematical methods.

Summarized, elicitation error specification is a measure for the acknowledged (partially) known deviations of all elements in the set R_w^{ref} with respect to R_w. During preprocessing of the reference knowledge fully known acknowledge errors can be accounted for or corrected. Obviously there is a chance that recognizable errors remain unacknowledged due to mistakes and blunders somewhere in the elicitation process.
Uncertainty in this regard are thus specific measures associated with a set or single element of R_w^{ref} , that define a quantity, value or range of values that could reasonably be attributed to the specified knowledge element in relationship to the real-world R_w. When evaluated and reported it qualitatively and/or quantitatively indicates the level of confidence that a specified real-world knowledge element actually represents the R_w or is within a bounded distance from the R_w.

The fidelity referent error and uncertainty qualification and quantification (EUQ&Q) section comprises a set of specification templates for each or a set reference knowledge parts of R_w^{ref} . They provide new and/or additional more detailed error and uncertainty information that is not specified elsewhere in the fidelity referent knowledge base structure. These specification templates contain the following constituent elements:

- *Real-World Reference Knowledge Part Coverage*. This element specifies or references those parts of R_w^{ref} (Section 5.4) to which this error and uncertainty specification belongs. This gives also the opportunity to specify total error and uncertainty in case when the local errors and uncertainties in separate reference knowledge elements do overlap or are correlated to each other.
- *Error Description*. A description of the known or identified error source(s), its origin (elicitation constraints, assumptions, limitations, etc), when possible the qualification and quantification of the error's magnitude and how this was determined, whether the error is resolvable and how, impact on the usability and reliability of the reference knowledge.
- *Uncertainty Description*. A description of the known or identified uncertainty source(s), its origin (elicitation constraints, assumptions, limitations, etc), its type (*aleatory* or *epistemic*), when possible the qualification and quantification of the uncertainty magnitude and what method(s) was used to determine this, whether the uncertainty is resolvable and what election efforts or experiments most be done to do so, impact on the usability and reliability of the reference knowledge.

Where, how and what kind of fidelity referent error and uncertainty sources can possibly be identified during the M&S development process is discussed in Chapter 0. The table at the next page gives an excerpt from the CN235 simulation project to illustrate how error and uncertainty knowledge can be specified in practice (*Table 5-15*). Underlining again indicates a physical links or reference to other parts of the fidelity referent where additional information is specified.

Reference Knowledge Error and Uncertainty Specification	
...	
EUQ&Q 1	
General Remarks:	*This section describes detected errors found in aileron registrations for symmetrical flight conditions and the impact of unknown local atmospheric conditions on these aircraft responses*
Referent Part Coverage	
Part:	*Behavior Sample Section: Long Handling Qualities, Short Period*
Part:	*Behavior Sample Section: Long Handling Qualities, Phugoid*
Part:	*...*
Error Description	
Sources & Origin:	*Not well calibrated or data processing error in aileron deflections*
Resolvability:	*None given the time and resources available to make test flights. However, application of simulated asymmetrical trim can be applied since they have hardly any effect on symmetric responses under investigation.*
Usability & Reliability Impact:	*Cannot be used since they cause no well asymmetrical trimmed flight state. Impact on the reliability of the rest of the flight results is considered limited since pilots have trimmed the aircraft properly according their debriefings.*

Qualification and Quantification		
Method	Description	Magnitude
Visual Inspection of Sample by SME	*Results showed both left and right aileron settings identical and almost constant outside the physical range, error is clear*	*Complete off physical possible range*
	...	

Uncertainty Description	
Sources & Origin:	*Local atmospheric conditions that haven't been measured but only global area-wide atmospheric conditions were known*
Type:	<~~Aleatory~~, Epistemic>
Resolvability:	*None given the time and resources available*
Usability & Reliability Impact:	*Negligible small impact and good usability*

Qualification and Quantification		
Method	Description	Magnitude
Logical reasoning by SME	*Evidence sources:* ** Global weather reports for the test flight show no sign for expecting significant local deviations from the excellent global atmospheric conditions* **Pilot and test engineers debriefings do not report any severe local atmospheric conditions encountered during test flights*	Less than 1% change of severe deviations due to not measured local atmospheric conditions

...

Table 5-15 Excerpt CN235 Referent Error and Uncertainty Specification

5.6 Summary

The existence and availability of a fidelity referent is an essential element in the practical assessment of simulation fidelity (Chapter 4). As formally defined by *Expression 4.13* a fidelity referent contains the following five high level constituent elements: the actual real world reference knowledge set R_w^{ref}, the reference knowledge error set δR_w^{ref}, the reference knowledge uncertainty set $u_{\delta}\left(R_{w}^{ref}\right)$, the reference knowledge election process set E_{rw} and its encountered limitations L_{R}^{elicit}.

In this chapter a practical realization of this fidelity referent paradigm has been developed along with additional underlying mathematical formulations for its constituent elements. This unified fidelity framework referent is realized in terms of a knowledge-base architecture composed of a structured set of generic and linked knowledge specification templates. These templates are grouped in three major subsets or knowledge area's that partly intersect:

- *Additive and Management Knowledge Area*. This area specifies the whole body of additive knowledge elements for carefully managing, using and maintaining the fidelity referent and it's contents (*Expression 4.13*). Such knowledge is essential for the proper application of a fidelity referent within the fidelity assessment activities throughout the whole simulation development and validation process as will be presented in Chapter 8.
- *Real-World Reference Knowledge Area*. This area specifies the actual real-world reference knowledge set R_{w}^{ref} that is used as input for the set of fidelity evaluator functions (Chapter 7) to assess the fidelity of a simulation system. The basis for this real-world reference knowledge template structure is the hierarchical object-oriented system specification paradigm discussed in Section 3.4. Using this approach the real-world reference is subdivided into two interrelated knowledge specification sets: structural and behavioral knowledge.
- *Elicitation Process Knowledge Area*. This area specifies all information regarding the elicitation (E_{rw}), limitations (L_{R}^{elicit}), errors (δR_{w}^{ref}) and uncertainties ($u_{\delta}\left(R_{w}^{ref}\right)$) of the information specified in the real-world reference knowledge area.

In both case-study applications (Appendix C) the developed architecture of structured knowledge templates proved to cover the most elementary and recurring specification elements required in any practical simulation fidelity assessment. Furthermore, they facilitated a means for the long overdue objective and precise measurement of simulation fidelity as desired by the simulation community (Chapter 0). Without no doubt these knowledge-base templates will not directly suite or cover every aspect of the almost infinite wide spectrum of simulation application problems that are addressed by this same simulation community. However, the unified fidelity framework referent knowledge-base templates are constructed such that they allow for easy tailoring and extending in order to fully suit any other specific application or problem domain. In this regard there remain several issues that require more attention and additional research to:

- Application and problem domain specific model and knowledge specification languages or methodologies. The outcomes from such studies can be used to refine, tailor and extend the in this thesis described fidelity referent templates. This will enhance the understandability and applicability of the current fidelity referent realization and its usage within the simulation fidelity assessment process for a larger public.

- Quantitative and qualitative methods for determining and facilitating real world reference knowledge errors and uncertainty for specific application or problem domains within the current unified fidelity referent knowledge base. Currently these errors and uncertainties are primarily specified in a coarse grained manner. An overview of methods that could be suitable for the assessment and specification of error and uncertainty in real-world reference knowledge is presented in Appendix B.
  Another important issue that requires attention in this context is the combination and specification of reference knowledge from multiple sources on the same aspect of the real world. Particularly, when such knowledge is inconsistent and conflicting.

- Development and implementation of dedicated automated tools that better support the specification, usage, visualization and management of knowledge within the fidelity referent knowledge-base architecture during the simulation system life cycle. Even in both case studies, relatively simple and small-scale simulation projects, involve a large amount of complex knowledge that has to be elicited, specified and managed in the knowledge-base templates. It proved to be too large for efficient and easy handling with the current, of the shelf, spreadsheet workbook implementation of this fidelity referent knowledge-base architecture. Availability of dedicated knowledge-base tools is therefore mandatory for successful cost-effective application and acceptance by the simulation community. Particularly if they can be integrated with other fidelity and simulation development tools to further cut development time and costs.
  A first possible step in the development of such tools is the translation of the in this Chapter discussed fidelity referent templates into an equivalent XML schema or data-type definitions (DTD). XML is standard extensible and structured markup language developed by the World Wide Web Consortium (W3C) for the creation and exchange of complex, structured data and documents. Due to their open and nonproprietary characters, XML based-documents are easily shared across many different computer platforms, tools and organizations.

6 Unified Fidelity Framework: Simulation System Knowledge Specification

6.1 Introduction

In chapter 4 a general applicable definition for simulation fidelity has been developed, which consists of a set of basic definitions, characterizations and formalisms for a unifying fidelity framework. Within this unified fidelity framework fidelity is formally defined by set of six constituent elements (*Expression 4.23*). The previous chapter discussed the first of these elements, the fidelity referent, in detail. This chapter focuses on the practical realization of the second element of this formal definition of fidelity, the total simulation system knowledge specification. Such a practical realization comprises the development of a knowledge-base architecture along with its underlying mathematical formulations for the specification of all available simulation system knowledge regarding its replication of the real-world. It is this knowledge that is used and compared with the fidelity referent to assess the level of fidelity within the unified simulation fidelity framework.

This chapter presents in Section 6.2 the high-level overview and relationships of the major building blocks that together form the simulation system knowledge-base. Next Sections 6.3 to 6.5 will present and develop each of these building blocks is in detail. Excerpts drawn from two aerospace simulation fidelity case studies are used in this chapter to illustrate each part of the developed simulation system knowledge-base template (Appendix C). A fully detailed presentation of both case-study simulation system knowledge-bases, implemented as a workbook of linked spreadsheets, is beyond the scope of this dissertation. Therefore, the majority of content of these knowledge-bases, although equally important in the complete fidelity characterization process, is deliberately summarized or omitted.

6.2 Simulation System Knowledge-Base: Structural Overview

Simulation system knowledge specification (S_{spec}) has formally been defined in section 4.4.2 (*Expression 4.3*). Similar to the fidelity referent the unified fidelity framework implementation of S_{spec}, formally defined by S_{appx} (Expression 4.14), constitutes a structured set of generic and linked specification templates, which together define a knowledge-base architecture. This architecture is hereafter called the simulation system knowledge-base.

The basic concept behind the design of the simulation system knowledge base is fidelity *Theorem 8* as developed in Section 4.3.2. This theorem states that the knowledge about the real-world system replication by a simulation system must be specified in similar problem domain knowledge terms and structured format as the real-world reference knowledge R_{ref}. A necessary condition to obtain consistent and comparable knowledge pairs for the practical fidelity assessment of the simulation system by means of the fidelity evaluator function set $\tilde{C}_{\Delta_{RS}}$ (*Expression 4.18*). Therefore, the fidelity referent knowledge-base design has been used as the bases for the unified fidelity framework simulation system knowledge-base architectural design as presented in this chapter. From a high-level view this architectural design contains specification templates that are grouped in the following major subsets:

- *Simulation Model Knowledge Specification* (Section 6.3), consisting of a series of templates for specifying the simulation model knowledge $S_{model} \in S_{appx}$ (*Expression 4.3*) including knowledge about the simulation model assumptions, data uncertainties, and the actual simulation system hardware and software implementation issues. Knowledge that supports the fidelity analysis, decision making process and reuse throughout the simulation system design and development phase from a structural perspective.
- *Simulation Execution Knowledge Specification* (Section 6.4), consisting of structured templates for specifying the series of performed simulation executions plus their associated pairs of configuration input settings $S_{config} \in S_{appx}$ and observed knowledge of the actual reproduction of the real-world $S_{rwr} \in S_{appx}$. Knowledge necessary for assessing the level of simulation fidelity from a behavioral perspective.
- *Complementary Knowledge Specifications* (Section 6.5), comprising a mixed set of templates for various complementary knowledge areas. First area is the specification of the set of simulation support systems $S_{support} \in S_{appx}$, which effect the simulation execution out come S_{rwr} (*Expression 4.4*). Next area specifies the elicitation process of all aforementioned specified simulation system knowledge along with the encountered limitations, errors and uncertainties in here (*Expression 4.17*). The last area is the additive and management area. An area specifying all additional knowledge necessary to properly manage, use, maintain and trace the simulation knowledge-base contents.

Figure 6-1 gives a graphical representation of the simulation system knowledge-base architecture. Arrows indicate the physical links to navigate between related knowledge-base sections.

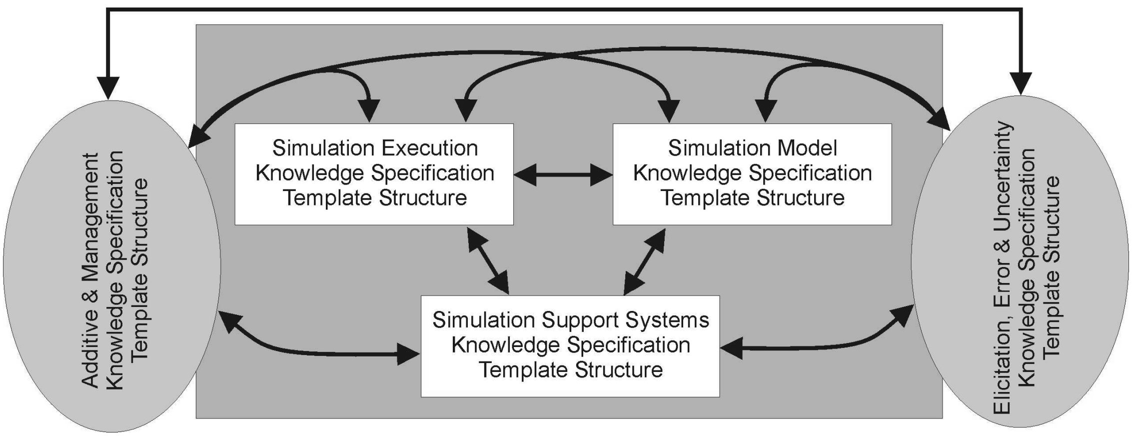


Figure 6-1 Simulation System Knowledge Base Structural Overview

6.3 Simulation System Knowledge-Base: Simulation Model Knowledge Specification Templates

A simulation model is the end product of the simulation development process (Section 3.3.1), which after being properly configured (S_{config_s}) by the user results in the replication the actual real-world over-time during a simulation execution $s \in S_{exec}$ (*Expression 4.4*). Therefore the simulation model forms the core of any simulation system. The knowledge specification for the simulation model comprises the following three elements:

- *Meta-Model Level Description*
- *Real-World System Realization Level Description*
- *Model, Parametric and I/O Data Uncertainty Description*

The contents and rationale for these three specification elements in the simulation system knowledge-base are discussed in the subsequent sections.

6.3.1 Meta-Model Level Description

In Section 3.2.3 a simulation model S_{model} has been defined as a set of n interconnected sub models M_{sub} capable of animating how the real world evolves over time. This set is formally defined as follows:

$$M_{model} = \left\{ M_{sub_1}, M_{sub_2}, \ldots, M_{sub_n} \right\} \tag{6.20}$$

In here each $M_{sub_i} \in M_{model}$ is responsible for the realization of a portion of this real-world replication by the simulation model during execution. Recalling this discussion and applying the composite component system formalism presented in Section 3.4.6 the next general definition for a simulation model can be given:

$$\begin{aligned} S_{model} = \Big(& T_{model}, U_{model}, Y_{model}, C_{model}, F_{cap_{model}} M_{model}, \left\{ I_m \,\middle|\, m \in M_{model} \cup \left\{ S_{model} \right\} \right\}, \\ & \left\{ E_m \,\middle|\, m \in M_{model} \cup \left\{ S_{model} \right\} \right\} \Big) \end{aligned} \tag{6.21}$$

Here T_{model} represents the simulation model time base. The multi-variable sets Y_{model} and U_{model} contain respectively the simulation model endogenous and exogenous I/O variables (Section 3.2.3). C_{model} is a multivariable set specifying the configurable variables of the simulation model such as parameters that are adjustable and sub models M_{sub}, which can be (de)selected or (de)activated. The functional capabilities of the simulation model, $F_{cap_{model}}$, are specified according the functional capability description as defined by *Expression 3.13* and *3.14*. In other words it defines the purpose(s) and application(s) the simulation model can be used or is intended for. Similar to the general composite component system formalism both I_m and E_m are respectively the set of sub model (*Expression 3.26*) that can possibly affect the m^{th} sub model in M_{model} along with the associated input output mapping set (*Expression 3.27*).

Expression 6.21 gives the most general formulation for a simulation model and formally defines S_{model} for any kind of *structural non-autonomous simulation systems* (Section 3.2.3). Using this expression it is also possible to formally define the simulation model of *structural autonomous simulation system* as a model S_{model} for which yield that $C_{model} = \varnothing$. Likewise, the simulation model of a *partial structural non-autonomous simulation system* can be defined as a S_{model} for which yields $C_{model} \cap M_{model} = \varnothing$. In other words the sub models of these simulation systems are pre-fixed.

The number of model component levels in *Expression 6.21* is by definition infinite. However for practical reasons in the unified fidelity framework the total number of these levels is limited to three (See *Figure 6-2*). These levels are chosen and named such to meet the current distributed and component-based simulation system architecture standards [11] [15] [17] [19] [26] [111] [156].

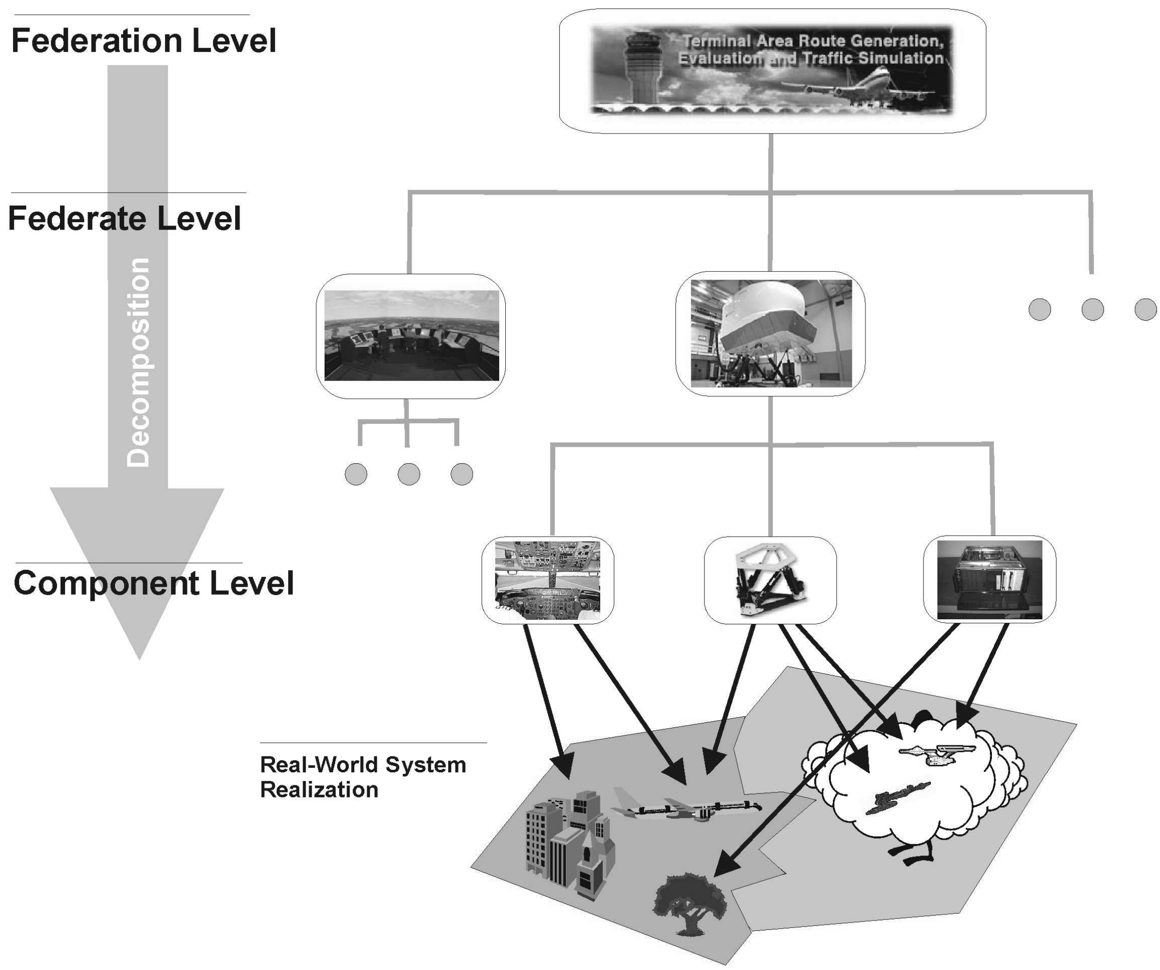


Figure 6-2 Unified Fidelity Framework Model Component Levels

The highest aggregation level of a simulation model is the federation model level (M^{fed}). A federation simulation model is composed of a set of *n* federate simulation models ($M^{fed}_{sub_i}$), which in their turn are composed of a set of *m* simulation model components ($M^{comp}_{sub_j}$). In case a federate simulation model isn't capable of joining or interoperating with other federates on a federation level that yields $M_{sub_i}^{fed} \notin M^{fed} \wedge \left(U_{model} = \emptyset \middle| U_{model} \in M_{sub_i}^{fed} \right)$, the simulation model is called *unitary*. Each $M_{sub_j}^{comp} \in M_{sub_i}^{fed}$ is responsible for the realization of a portion of this real-world system replication (Section 6.3.2).

The knowledge specification templates for the meta-model level description contain the following three constituent elements:

- *Overall Meta-Model Properties*. This area gives an overall textual and qualitative description of the structural composition of S_{model} in terms of its constituent (sub)models M_{sub_i}. Furthermore the following cardinal numbers can be specified to quantitatively characterize the complexity of the meta-model structure:
  - $n\left(S_{model}\right)_{M_{model}}$: The total number of (sub)models M_{sub_i} that populate S_{model}.
  - $n\left(S_{model}\right)_{M_{sub}^{fed}}$: The total number of federate (sub)models that populate S_{model}. If this number is one, this signifies a unitary simulation system.
  - $n\left(S_{model}\right)_{M_{sub}^{comp}}$: The total number of component models that populate S_{model}.
  - $n\left(S_{model}\right)_{M_{config}}$: The total number of (sub)models M_{sub_i} that can be (de)selected through C_{model}. This number is composed as follows:

$$n\left(S_{model}\right)_{M_{config}} = n\left(M_{config}^{fed}\right) + \sum_{i=1}^{n_{fed}} n\left(M_{config_i}^{comp}\right) \tag{6.22}$$

    In here $M_{config}^{fed} \subseteq M^{fed}$ is the subset of configurable model federates and $M_{config_i}^{comp} \subseteq M_i^{comp}$ is the subset of configurable model components of the i^{th} of the in total n model federates that make up the simulation model.
  - $n\left(S_{model}\right)_{interaction}$: The total number of interaction relationships between the (sub)models in S_{model}, which is the sum of the interaction set I_m cardinality of each (sub)model M_{sub_i}.

As an example the overall meta-model properties of the *Future Airspace Simulation Environment* are given (Appendix C):

Overall Structural Properties	
Total NOF Submodels	*11*
Total NOF Federates	*11*
Total NOF Components	*20*
Total NOF Configurable	*15*
Total NOF Interactions	*38*

Table 6-1 Overall Structural Meta-Model Specification Example

- *Meta-Model Topology Map*. This is a graphical representation of the meta-model composition of the simulation model in terms of federation, federate, unitary and component models, which is similar to the real-world system topology map (Section 5.4.1). Colored rectangles are used to visualize the simulation model sub models (*Figure 6-3*). A single arrowhead connection between two rectangles indicates a composition relationship. Double arrowhead connection is used to specify the interaction relationship and their direction between associated (sub)models. Rectangles with red borders are used to visualize (sub)models that can be (de)selected for each simulation execution through the simulation model configuration variable set C_{model}. All other (sub)models are pre-fixed during any simulation execution. To illustrate the meta-model topology map specification, an excerpt from the *Future Airspace Simulation Environment* project is presented in the figure below. A set of three dots signifies other equally important (sub)models but which are omitted in this excerpt for brevity.

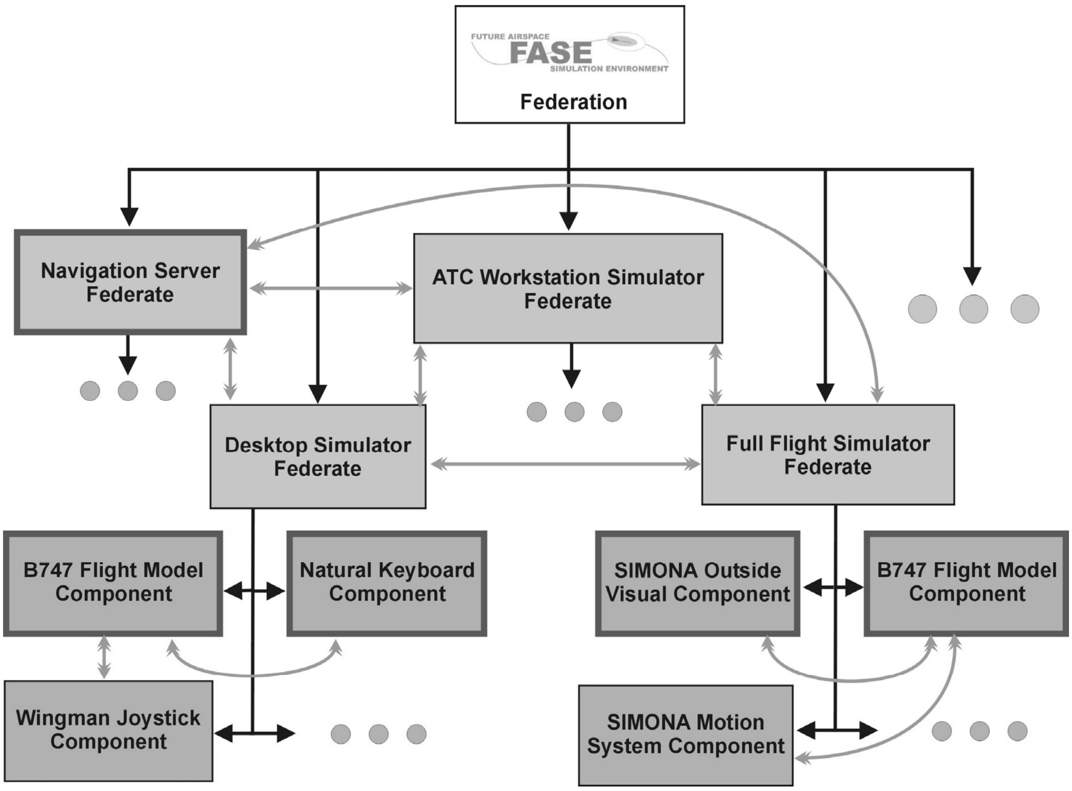


Figure 6-3 FASE Meta Model Topology Map

- *Meta-Model Characteristics Description List*.
The meta-model characteristics description list is an structured array of detailed descriptions for the (sub)models M_{sub} in a meta-model topology map (*Figure 6-3*). Like for the real-world system topology map (Section 5.4.1), in a automated for simulation model knowledge specification the detailed knowledge about the sub-model or interaction as discussed in this section should be shown up when pointed to in the meta-model topology map. For each model $M_{sub_i} \in M_{model}$ the following knowledge elements are specified:

    - *Name*. A unique name to identify the sub-model.

- *Model Type*. Specifies whether the sub-model is a *federation*, *federate*, *unitary* or *component* type.
- *Configurability*. A $n \in \mathbb{N}$ is used to indicate whether the sub-model is (non)selectable through the simulation model configuration settings (C_{model}); $n = 1$ means not-selectable and $n \geq 1$ means selectable plus the number of possible instances (*multiplicity*).
- *Input and Output Properties List*. This list describes all input U_{model} and output Y_{model} properties of the sub-model. For each of these properties besides their name also a description of what the property stands for, where applicable the mathematical symbol and dimension is specified.
- *Functional Capabilities List*. Description of all model's functional capabilities in relationship to other (sub)models in the topology map. These capabilities are specified according to the functional capability F_{cap} (*3.13*) and means-end hierarchy F_{cap_c} (*3.28*) specification formalisms as outlined in Section 3.4.

- *Topology Characteristics List*. Specifies the model's sub-model reference set M_{model} and also the reference to the model's parent model in the total meta-model hierarchy. In case of a model *component* type the link(s) to the associated real-world realization description, in terms of the replicated real world (sub)system models $S^{rw}_{model_k} \in M_{comp_i}$, are specified instead (See section 6.3.2). The second part of this list is the specification of all the model's interactions (E_m and I_m) with the other models in the meta-model topology map. For each interaction a textual description of the interaction mechanism and reference to the other involved model component is given along with the observable properties (name, dimension, input or output identifier, practical hard & software realization) associated with the interaction. Furthermore, a specification of what and how the external input/output from other model components is processed and fed to the internal input/output interface of each $S^{rw}_{model_k} \in M_{comp_i}$ (See Section 6.3.2).

  The hard & software realization characteristics are necessary items in the assessment of possible sources and bottlenecks of unacceptable fidelity mismatch issues during simulation system development and validation process (Chapter 9). These effects are most significant and typical for hardware-in-the-loop and distributed simulation system architectures [47] [51] [119]. In here model components are interconnected and interoperate through a sensor/stimulator and/or communication network systems whose performance characteristics (latency, bandwidth, sample rates, etc.) can lead to discrepancies in the real-world replications of each model component.

Table 6-2 gives an excerpt of the meta-model characteristics description for the Future Airspace Simulation Environment (FASE) as depicted in *Figure 6-3*. Underlining of text again represents links to other referent sections.

Identification and General Properties	
Name:	*B747 Flight Model*
Model Type:	*~~Federation~~, ~~Federate~~, ~~Unitary~~, Component*
Configurability:	*0*

Input Output Properties	
…	
Property 1	
Name:	*Scaled Control Collum Position*
Description:	*The control collum deflection scaled between <-1,1> of the min and maximum deflection of the control input device.*
Symbol:	S_e
Dimension:	*[-]*
Type:	*~~Output~~, Input*
…	
Functional Capabilities	
…	
Capability 1	
Goal:	*Replication of the aircraft flight and ground operation dynamics*
Inputs:	*Scaled Control Collum Position, Scaled Control Wheel Position, …*
Outputs:	*Aircraft Position, Aicraft Attitude, Aircraft Ground Speed, …*
Sub Functional:	*F_{cap} 6 DOF Equations of Motion, …*
…	
Topology Composite Component Characteristics	
Parent Model:	*Desktop Simulator*
Sub 1	
Name:	*Rolls-Royce RB211-524H Jet Engine System Model*
Multiplicity:	*{4}*
Topology Interactions Characteristics	
…	
Interaction 1	
Name:	*Flight Control Deflection*
Description:	*A flight control device deflection causes an input signal that drives the replicated B747 flight control system* to simulate the B747 elevator, rudder and aileron deflections during the simulated flight execution.
Involved Model:	*Wingman Joystick*
IO Variable 1:	*Scaled Control Column Position { B747 flight control system .S_e} (input, <-1,1>, [-])*
IO Variable 2:	*Scaled Control Wheel Position { B747 flight control system .S_r} (input, <-1,1>, [-])*
…	…
H&S Realization:	*The interaction with the Wingman Joystick Component is realized by means of a software implemented connection (C++) via the local memory of the desktop*

Table 6-2 Meta-Model Characteristics Description

6.3.2 Real-World System Realization Description

As discussed in the previous section the *model component* is the lowest level in element in meta-model, which is responsible for the actual replication of the whole or portion of the real-world (*Figure 6-2*). The real-world system realization description comprises the specification of the structural composition of the modeled real-world by each $M_{comp_i} \in S_{model}$ in terms of a set of *m* hierarchical ordered and interacting real-world (sub)system models $S^{rw}_{model_k}$. Each real-world (sub)system model $S^{rw}_{model_k}$ is a simulation system's counterpart representation for a single actual real-world system $S^{rw}_i \in S^{rw}$ (*Expression 5.1*). Similar to $S^{rw}_{ref_i}$, which specifies all available reference knowledge for S^{rw}_i in the fidelity referent knowledge-base, $S^{rw}_{model_i}$ is the placeholder for the available knowledge about the representation of the same S^{rw}_i within the simulation system knowledge-base (Section 5.4). Using the general formal specification for a simulation model (*Expression 6.21*) this means that for each $M_{model} \in M_{comp_i}$ yield:

$$M_{model} = M_{model}^{rw} = \left\{ S_{model_1}^{rw}, S_{model_2}^{rw}, ..., S_{model_m}^{rw} \right\} \tag{6.23}$$

Like for each $S_{ref_k}^{rw} \in R_w^{ref}$ the object-oriented system knowledge specification paradigms of Section 3.4 serve as the basis for each $S_{model_k}^{rw} \in M_{model}^{rw}$ to specify its structural composition, relationships and behavioral knowledge of the replicated real-world system S_i^{rw}. Depending on the simulation model *component* a portion of its constituent real-world (sub)system models $S_{model_k}^{rw}$ might be configurable through the simulation model configuration settings (C_{model}). This yields that M_{model}^{rw} can be divided into two complementary subsets:

$$M_{model}^{rw} = M_{config}^{rw} \cup M_{non\text{-}config}^{rw} \tag{6.24}$$

where M_{config}^{rw} is the subset of configurable real-world (sub)system models and $M_{non\text{-}config}^{rw}$ the subset of non-configurable real-world (sub)system models.

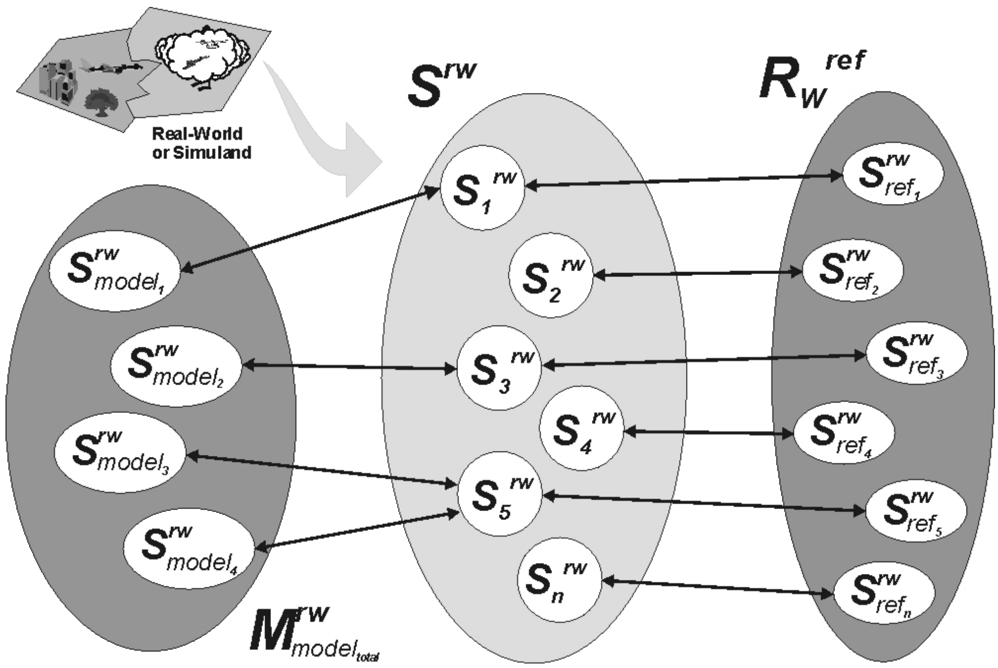


Figure 6-4 Directed Graph for the Real-World, Referent and Simulation Model

Usually a simulation model consists of more than one model component therefore the total set of real-world systems that can possibly be replicated by the simulation model is given by (figure *Figure 6-2*):

$$M_{model_{total}}^{rw} = \bigcup_{i=1}^{n_{fed}} \left(\bigcup_{j=1}^{k_{comp}} M_{model_j}^{rw} \right) \tag{6.25}$$

In *Expression 6.25* n_{fed} is the number of simulation model federates and k_{comp} the number of model components of the simulation federate *i*. Unlike the fidelity referent, the total real-world (sub)system model set, by definition, doesn't have a *one-to-one correspondence* relation with the real-world system set (*Figure 6-4*). This means that some real-world systems are not represented by the simulation model or it may contain multiple models for the same real-world system.

Obviously the lack of certain real-world system representations in the simulation model affects the realism of the simulation and may induce problems in the usage of its results. However, multiple model representations for the same real-world system, which are simultaneously used during simulation execution, may also affect the realism of the simulation outcome. In literature such multiple model representations are called manifold representations [51]. Following the mathematical notations used in this thesis, manifold representations can be formally defined as a collection of *k* sets $M^{rw}_{manifold_k} \subseteq M^{rw}_{model_{total}}$:

$$M^{rw}_{manifold} = \left[\left\{ M^{rw}_{manifold_1} \right\}, \left\{ M^{rw}_{manifold_2} \right\}, \ldots, \left\{ M^{rw}_{manifold_k} \right\} \right]$$

$$where \qquad (6.26)$$

$$\left\{ \forall M^{rw}_{model_i} \in M^{rw}_{manifold_k}, S^{rw}_k \in S^{rw} : M^{rw}_{model_i} \mapsto S^{rw}_k \right\}$$

Manifold representations defined by *Expression 6.24* can occur in both distributed and unitary simulation systems. For instance various visual databases for the same terrain representation or dead-reckoning representations of the same vehicle state coexist in different federates [118] [126]. As an example of unitary simulations consider *Figure 4-8* where both a directly interacting computed and physical flight-deck acceleration state coexist for the same aircraft.

The above discussed real-world system realization part of the simulation model is organized in the unified fidelity specification template in three information areas: *Overall Structural Properties*, *Real-World System Model Topology Map* and *Real-World System Model Characteristics and Realization Description List*.

Overall Structural Properties

This template is the counterpart for the overall structural properties fidelity referent template developed in section 5.4.1. Similar to that template this template gives an overall description, both qualitative and quantitative, of the structural composition of the total replicated real-world by the simulation model in terms of its constituent (sub)system models $S^{rw}_{model_k} \in M^{rw}_{model_{total}}$ and their interactions. Except for additional types of cardinal numbers this specification template is identical to the *overall structural properties fidelity referent template* but now applying to the replicated real-world structure. For brevity this template is not discussed in detail and illustrated here again.

The simulation model specific cardinal numbers that have to be specified here are the following:

- $n\left(M^{rw}_{model_{total}}\right)_{config}$: The total number of configurable real-world system models in $M^{rw}_{model_{total}}$, which is defined as:

$$n\left(M^{rw}_{model_{total}}\right)_{config} = \sum n\left(M^{rw}_{config_i}\right) \quad \text{for} \quad \forall M^{rw}_{config_i} \subseteq M^{rw}_{model_i} \in M^{rw}_{model_{total}} \quad (6.27)$$

- $n\left(M^{rw}_{manifold}\right)$: The total number of manifold representations in $M^{rw}_{model_{total}}$.

These cardinal numbers characterize the configurability and thus reusability of the simulation model for various application purposes. Furthermore, they give high-level information of the flexibility in structural fidelity and associated issues during simulation system development and validation.

Real-World System Model Topology Map

The constitution of the real-world system realization by the simulation model can graphically be displayed in a similar fashion as the *topology map* imparted in the fidelity referent knowledge-base (Section 5.4.1). The rationale for creating such a real-world system model topology map is exactly the same, easy visual inspection and navigation through its structural composition. It comprises the graphical representation of the interaction and composite-component relationships between all real-world (sub)systems models $S^{rw}_{model_k} \in M^{rw}_{model_{total}}$. Except for an additional configurability symbol, the real-world system model topology map symbols and color-coding are identical to the one imparted in the fidelity referent knowledge-base. A configurable system model is indicated with a black asterisk in the ellipse.

Real-World System Model Characteristics and Realization Description List

The real-world system model characteristics and realization description comprises an array of detailed specification of each $S^{rw}_{model_k} \in M^{rw}_{model_{total}}$. The internal system knowledge formalism given in Section 3.4.5 is used as the basis for this real-world system model knowledge specification. This specification template can be divided in two subsections, which are discussed next.

First subsection is the simulation model knowledge-base counterpart of the *Real-World System Characteristics Description* template for $S^{rw}_{ref_i}$ in the fidelity referent knowledge-base (Section 5.4.1). Except for the omission of the *Reality Class* and *System Type* fields and the next additional knowledge elements both templates are identical:

- *Real-World System Reference*. A reference or name to the real-world system $S^{rw}_i \in S^{rw}$ (*Expression 5.1*) that is represented by $S^{rw}_{model_k}$. This element directly translates to a link to the associated reference knowledge $S^{rw}_{ref_i} \in R^{ref}_w$ within the fidelity referent knowledge-base.

- *Component Model Reference*. A link to the parent *meta-model* component model to which the real-worlds system model belongs.
- *Configurability*. A $n \in \mathbb{N}$ is used to indicate whether the system model is (non)selectable through the simulation model configuration settings (C_{model}); $n = 1$ means not-selectable and $n \geq 1$ means selectable plus the number of possible instances (*multiplicity*).
- *Manifold Representations*. A listing of references to other system models, part of the simulation model, replicating the same real-world system $S_i^{rw} \in S^{rw}$.
- *Parameter Configurability and Range*. In the representational and behavioral properties list two additional knowledge areas are added for the system model parameter set $P_{model} \in S_{model_k}^{rw}$ specification. The parameter set P_{model} can be divided in two disjoint subsets:

$$P_{model} = P_{config} \cup P_{non-config} \qquad (6.28)$$

  In *Expression 6.27* P_{config} contains all system model parameters that are constant for every simulation execution while $P_{non-config}$ specifies the set of system model parameters that are configurable through the simulation model configuration settings (C_{model}). First a Boolean value is used to specify whether the parameter is configurable or not. Second the parameter range, *range*(P_{model}), is specified, which is a constant value in case of non-configurable parameter. This range is specified in a similar fashion as in the non-causal behavior knowledge area of the fidelity referent knowledge-base (Section 5.4.2).
- *Interaction Type and Physical Data Exchange*. In the interaction specification two additional fields are added. The first element specifies whether the data part of an interaction between two system models $S_{model_k}^{rw}, S_{model_j}^{rw} \in M_{model_{total}}^{rw}$ is exchanged *locally*, i.e. within the same model component, or *externally*. Next it is specified how the interactions in terms of simulated data exchange between each other is physically realized (See also 6.3.1).

Since the previous specification template is almost identical to the *Real-World System Characteristics Description* template presented and practically illustrated in Section 5.4.1, this template is not stated and illustrated here again.

The second specification subsection for $S_{model_k}^{rw}$, called *Real-World System State Transition and Output Realization Description*, builds upon the internal system knowledge specification formalisms of state transition (Δ) and output (Λ) function (*Expression 3.18* and *3.19*). These two formalisms specify for each $S_{model_k}^{rw}$ the internal model structure and working that will realize the behavior replication of the associated real-world system during simulation execution. Prima facie, Δ and Λ specify the actual hardware and/or software implementation of the functional capabilities set $F_{cap_k} \in S_{model_k}^{rw}$ along with it's underlying mathematical relationships, algorithms, assumptions, performance characteristics and limitations. Depending on the number, type and complexity of these functional capabilities, Δ and Λ can be broken done in a set partially and/or fully autonomous elements that implement a specific behavioral part of $S_{model_k}^{rw}$.

These two sets are defined as follows:

$$\Delta = \left\{ \hat{\Delta}_1, \hat{\Delta}_2, ..., \hat{\Delta}_n \right\} \quad \text{and} \quad \Lambda = \left\{ \hat{\Lambda}_1, \hat{\Lambda}_2, ..., \hat{\Lambda}_m \right\} \tag{6.29}$$

There exist many possible types of simulation model implementations for Δ and Λ, which heavily depend on the problem and application domain at hand. Likewise a wide variety and mixture of languages and formats can be used to specify both these two elements. Therefore, it is difficult to provide a single standard specification template (See also Section 3.4.5). However, it is quite well possible to develop a set of generic knowledge area, which such a specification template for each pair $\left(\Delta, \Lambda\right) \in S^{rw}_{model_k}$ should posses. These knowledge areas form the basis for the unified fidelity framework *Real-World System Realization Description* template and specify the following items for each element $\hat{\Delta}_i \in \Delta$ and $\hat{\Lambda}_j \in \Lambda$:

- *Realization Identifier*. A unique name to identify the behavioral representation part.
- *Realization Type*. An identifier specifying whether the realization is a physical model or computer model representation of the real-world (Section 3.2). Furthermore it must be indicated if the representation is deterministic or stochastic (Section 3.4.5).
- *Realization Goal Description*. A description of what part of the real-world system functional capabilities this realization tries to replicate. This in relationship to the other real-world system models $S^{rw}_{model_k} \in M^{rw}_{model_{total}}$ and model components (M_{comp_i}) that are part of simulation model.
- *Realization Assumptions and Limitations*. A specification of the assumptions or simplifications that apply to this realization. The implications in terms of limitations on the real-world system replication should be stated as well. These assumptions and their implications can address many factors, including effects on the replication of the real-world in other parts in the simulation system [102]. Any specified model error and uncertainty in this regard should be referenced here (Section 6.3.3).
- *Realization Time-Base Specification*. This section involves the specification of the simulation time-base using the following identifiers:

    - *Time-base type*: Either a continuous ($T_{\mathfrak{R}}$) or discrete ($T_{\mathfrak{I}}$) time-base (Section 3.4.3).

    - *Time speed-up factor (T_{speed_up})*: A value quantifying the scaling factor for the simulated time-base with respect to the real time-base, defined as follows:

$$\frac{l\left(\omega_{\langle t_1, t_2 \rangle}\right)_{sim}}{\left(t_{RT_2} - t_{RT_1}\right)_{\omega_{\langle t_1, t_2 \rangle}}} = T_{speed\_up} \tag{6.30}$$

Here $l(...)_{sim}$ is the length of a state transition segment (*Expression 3.4*) resulting from the realization during simulation execution. The difference between t_{RT_1} and t_{RT_2} defines the period in wall-clock or real-time it takes for the realization to produce this segment. A $T_{speed\_up} = 1$ indicates that the realization runs at real-time. $T_{speed\_up} > 1$ and $T_{speed\_up} < 1$ means respectively faster and slower execution than real-time.

- *Minimum timeframe size* ($T_{frame_{\min}}$): The smallest timeframe, in respect to the simulated time-base, with which the realization is capable of producing simulated state updates or generates events.

Both time speed-up factor and minimum timeframe size are influenced by the system model hardware performance and algorithm processing times.

- *Input, State, Parameter and Output Specification*. Specifies the subset of input, state, parameter and output variables of $S^{rw}_{model_k}$ that are part of this realization replication of real-world system behavior. When applicable, the variable update timeframe (T_{frame}) must be specified, defined as:

$$T_{frame} = n \cdot T_{frame_{\min}} \; with \; n \in \Im . \tag{6.31}$$

- *Realization Implementation Description*. This section describes the actual implementation and internal working of the real-world system replication. In case of a computer model its model type (*discrete or continuous time or discrete event*) is stated here, along with physical laws, databases, numerical and computational algorithms such as integration or table interpolation routines, software implementation and protocols etc. These descriptions can include references to simulation system design documents or other information sources on the implementation details. For physical models the physical shape, dimensions and working are described, along with its performance characteristics. The exact contents and used language in this knowledge section depends on the nature and type of the implementation and is therefore not further pre-specified in the unified fidelity framework simulation model knowledge-base. This area can be tailored to suite a particular problem or application domain at hand.

As an example of a *Real-World System Realization Description* consider the next excerpt from the FASE case-study, which describes the real-world system model realization of a primary radar system (*Table 6-3*). Like for the fidelity referent knowledge-base underlining of text represents a link to other parts of the simulation system knowledge-base where additional information is found.

Real-World System Model Characteristics & Realization Description List	
…	
System Model 1	
Real-World System Reference	*Primary Air Surveillance Radar*
Characteristics Description	
…	
State Transition and Output Realization List	
…	

Realization 1	
Name	*Primary Radar Plot Generation*
Type	*~~Physical Model~~, Computer Model* *~~Stochastic~~, Deterministic*
Goal Description	*Presentation of Detected Aircraft Entities to the Air Traffic Controller (functional capability 4)*
Assumptions, Simplifications and Limitations	
Assumption Description	Limitation Description
Perfect Ground Clutter Filtering	*No representation of false echo's and removal desirable targets behavior of MTI circuitry systems*
…	…
Time-Base	
Type	*~~Continuous Time~~, Discrete Time, ~~Discrete Event~~*
Speed-up Factor	*1*
Minimum Timeframe	*0.1 [s] or 10 [Hz]*
Input, State, Output and Parameter Variable	
Variable Name	Update Timeframe
Radar Plot	*10 [s] or 0.1 [Hz]*
Antenna Azimuth Angle	*0.1 [s] or 10 [Hz]*
…	…
Implementation	
… Implemented as a C++ class (source files: FASEPrimaryRadar.h & FASEPrimaryRadar.cxx) …	

Table 6-3 Excerpt FASE Primary Radar System Model Characteristics and Realization Description

6.3.3 Model, Parametric & I/O Data Uncertainty & Error Source Knowledge

Limitations and constraints placed on the simulation model development process are potential sources for imperfections and uncertainty in the replication of the real world by the simulation system. In other words sources that effect the quality of the internal realization of the real-world system models and thus the attainable level of fidelity during simulation execution. As discussed in section 4.3.3 *precision* is one of the eight descriptors for characterizing fidelity. Therefore, specifying any possible knowledge regarding these imprecision error and uncertainty sources in the simulation model is essential. Within the unified fidelity framework simulation model knowledge-base the following three related imprecision error and uncertainty source category are defined:

- *Structural Imprecision Error and Uncertainty*: These are fidelity error and uncertainty sources in the simulation model its structure, underlying concepts and logic. Examples of such sources include incomplete and ambiguous knowledge, necessary simplification of real-world system complexity, discretization of continuous real-world processes and systems, aggregation of separate real-world system behaviors.
- *Data Imprecision Error and Uncertainty*: These are fidelity error and uncertainty sources in the simulation model parametric data. Sources in this category include imprecision and accuracy of parameters due to instrument error, round-off errors, data conditioning process errors. Furthermore, these sources include parameter data gaps, inconsistencies between data from different sources, and data adequacy problems.
- *Estimation Imprecision Error and Uncertainty*: These are fidelity error and uncertainty sources in the simulation model due to system identification and inference processes [80]. These sources include simplifications for mathematical

treatment such as linear regression model versus a parabolic model, ambiguous chose of the optimum principle in a parameter estimate or limited data recording length, replications or coverage i.e. number of measured data points.

Another important aspect in this context, as expressed by the fidelity descriptor *sensitivity* (Section 4.3.3), is the effect or sensitivity of the simulation execution outcomes to the above-mentioned imprecision error and uncertainty sources of a simulation model. The common approach to analyze these effects is called sensitivity analysis [22] [67] [146]. In here the basic idea is to systematically change the model input, parameter data or structure over a certain range of interest and observing the effects upon simulation execution outcome. As easy as this may seem sensitivity analysis of complex and large-scale models remain an important problem and expensive [146]. However, the results of sensitivity analysis are useful to assess and estimate the level of simulation fidelity with respect to the simulation model's capability to predict the yet unknown real-world system knowledge [156]. Furthermore, this knowledge forms the start point for tracing the causes of deficiencies in the desired level of simulation fidelity in replicating known or measured real-world system knowledge. Therefore any performed sensitivity analysis results should be specified here as-well.

The specification template for the simulation model, parametric and I/O data imprecision uncertainty and error source knowledge is almost similar to the *error and uncertainty specification template* in the referent knowledge-base (Section 5.5.2). Therefore only the differences are discussed here, which encompass the following additional specification elements:

- *Error and uncertainty Qualifier Area*: This area specifies the imprecision error or uncertainty source category being *model*, *data* or *estimation*.
- *Sensitivity Analysis Knowledge Area*: Here the details of the possibly preformed sensitivity analysis are specified. This includes the description of the used analysis technique itself, what and with what range changes have been made to the model, input or parametric data, the quantitative and qualitative results accompanied with the drawn conclusions or known implications regarding the simulation fidelity.

6.4 Simulation System Knowledge-Base: Simulation Execution Knowledge Specification Templates

Except for sensitivity analysis, the simulation model knowledge specification S_{model} is a 'static' knowledge specification process in the sense that it doesn't require any simulation execution. The simulation execution knowledge part of the simulation system knowledge-base specifies all available knowledge regarding the real-world replication over time during simulation execution.

Any simulation system, except for structural autonomous simulation systems for which yields that the $S_{config_s} = \varnothing$, will provide some degree of freedom to configure the simulation model (Section 3.2.3 and 6.3). That means that the final structural composition of the real-world system replication by a simulation model and thus also the associated behavior varies with each simulation system configuration setting S_{config_s}

during simulation execution (*Expression 4.4*). Therefore, the total knowledge specification for each i^{th} simulation execution $s_{exec_i} \in S_{exec}$ is determined by the pair of S_{config_s} and S_{rwr_s} (*Expression 4.3)*. However, due to elicitation limitations L_s^{elicit} (*Expression 4.14*) only a portion of all possible simulation executions can be or will be performed: $\tilde{S}_{exec} \subseteq S_{exec}$. For the same reasons is $\tilde{S}_{rwr_s}$ the approximated knowledge specification of the observed simulated real-world system S_{rwr_s} during simulation execution.

The simulation execution knowledge specification therefore comprises a list of the performed simulation executions $\tilde{S}_{exec}$ (*Figure 6-5*). And for each of these simulation executions it is required to specify S_{config_s} and S_{rwr_s}. In the next two subsections the specification format and templates for S_{config_s} and $\tilde{S}_{rwr_s}$, as used in the unified fidelity framework, are presented.

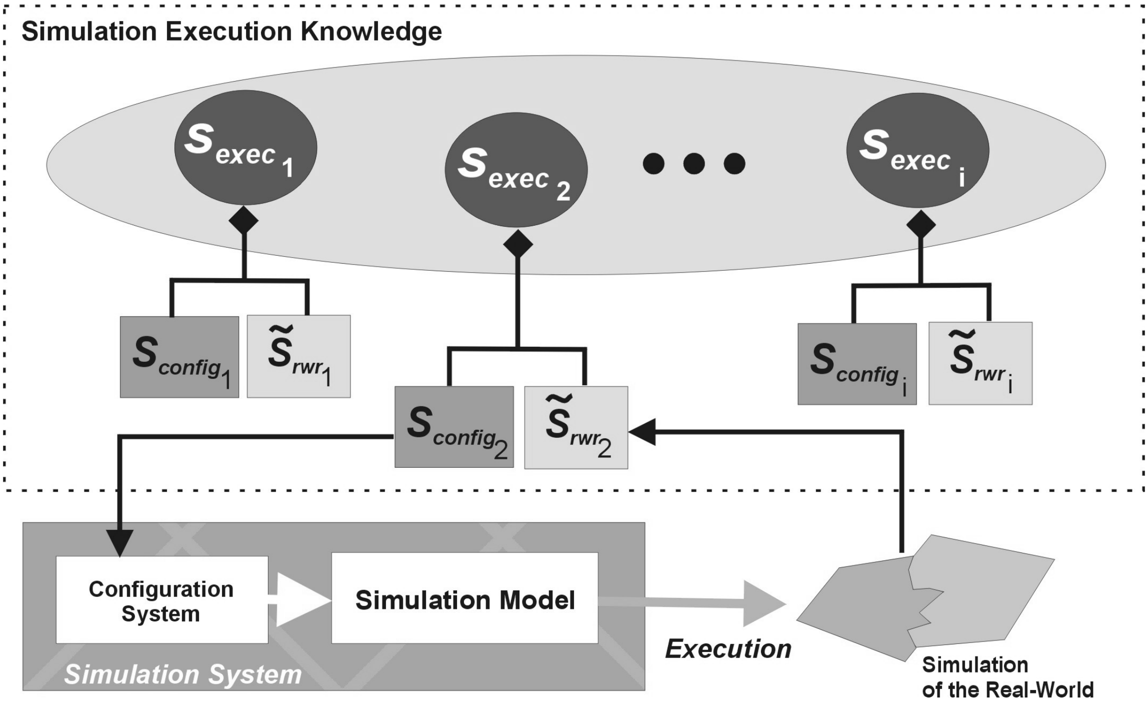


Figure 6-5 Simulation Execution Knowledge Definition

6.4.1 Simulation System Configuration Setting Specification

The degree of freedom in configuring a simulation system varies from simulation system to simulation system. Sometimes for relative simple simulation systems only a certain initial condition in the form of a for instance a vehicle position can be configured, while for distributed simulation systems based on for instance HLA complete simulation model components in the form of federates can be selected. This degree of freedom is determined by the required simulation execution scenarios that are necessary to full-fill the user-needs (Section 9.2). Therefore, unified fidelity framework the specification format and template for S_{config_s} is a generic one that covers the most common and essential configuration setting elements. Like other parts of the simulation system knowledge specification the specification template, when necessary, can be tailored to fully fit the simulation system at hand. The unified S_{config_s} specification template addresses the following configurable elements:

- *Simulation Execution Identifier*. A unique name or id of the performed simulation execution to which this configuration setting applies.
- *Simulation Model Configuration*. Specifies the union of all the selected configurable federate models (M^{fed}_{config}), model components (M^{comp}_{config}) and real-world system models (M^{rw}_{config}) in C_{model} (Section 6.3.1). When applicable the number of instances of both prefixed and selected sub-models must be stated along with their id's. This defines the complete simulation model structural configuration, which is referred to in the unified fidelity framework as the *executable* simulation model:

$$S^{exec}_{model} \subseteq S_{model} \tag{6.32}$$

- *Parameter Settings*. Specifies for each $S^{rw}_{model_k} \in S^{exec}_{model}$ the values for the configurable parameters $p \in P_{config} \subseteq P$. These parameter values can range from a single value, for instance an aircraft payload mass, to loading complex parameter set from file such a specific aerodynamic data set.
- *Initial Condition Settings*. Specifies for each $S^{rw}_{model_k} \in S^{exec}_{model}$ the initial values for the state variables $q_{t_0} \in Q$ at the start ($t = t_0$) of the simulation execution (Section 3.4.5). This can but doesn't necessary have to a steady initial state.
- *Situational and Operational Context Description*. Specifies a description of the real-world situational and operational context in which the replicated real-world systems operate together to achieve the simulation execution objectives. This includes elements such as mission statement and planning description, pilot briefings and experiment execution description. The exact content and format used to specify a simulation context is at the discretion of simulation scenario developer and is determined by the application and problem domain (Section 9.2.2).
- *Behavioral Input and Scripting Specification*. Specifies recorded and/or scripted input samples that are (re)played during the simulation execution. These samples are mostly used in the simulation model for the representation of random and artificial behavior (failures, environmental conditions, human behavior and controls, traffic samples etc). Format can range from a series of scripted events or events on a timeline documented in a script file to a simply a reference to a behavior sample that is loaded.
- *Termination Condition Specification*. Specifies the simulation execution termination conditions for which yields that the execution objectives have been accomplished.

To illustrate this simulation system configuration setting specification again an excerpt of the FASE case-study is provided (*Table 6-4*). See also *Figure 6-3*. This example configuration setting concerns a free-flight scenario with the simulation execution objective of studying airborne separation assurance system (ASAS) logic and cockpit displays of traffic information (CDTI) in relationship to the pilot's ability to detect and solve separation loss conflicts. In the next table underlining represents links to other parts of the simulation system knowledge specification. The used symbol "::" signifies the scope to which an system type and/or an instance belongs too. Since all FASE case-study simulation model federates and model components use configuration files to store and load parameter and initial conditions for the replicated real-world systems, references to the used files are included in this simulation system configuration specification. Furthermore, in support of the HLA distributed simulation technology it is indicated in the FASE configuration specification whether selectable real-world systems are locally or remotely owned [15] [16].

<table>
<tr><th colspan="5">Simulation System Configuration Specification</th></tr>
<tr><td>Execution ID:</td><td colspan="4">(FF-P-AC-1) Free-Flight Pilot-in-Loop ASAS and CDTI Experiment 1 run 4</td></tr>
<tr><th colspan="5">Selected Structural Composition</th></tr>
<tr><th>Federate Name</th><th>Instance ID</th><th>Model Component</th><th>Local Owned Real-World Systems Representations</th><th>Local Remote Real-World Systems Representations</th></tr>
<tr><td rowspan="2">FASE B747-400 Desktop Simulator</td><td>LRSATS 1</td><td>WingmanJoystick::LRJS1</td><td>B744::SIA267</td><td>B744::KLM390
B735:: EZY544
MD11:: MPH978
A330:: LH5908
...</td></tr>
<tr><td>LRSATS 3</td><td>WingmanJoystick::LRJS2</td><td>B744::KLM390</td><td>B744::SIA267
B735:: EZY544
MD11:: MPH978
A330:: LH5908
...</td></tr>
<tr><td>FASE Pseudo-Pilot Aircraft Simulator</td><td>LRSATS2</td><td>N.A.</td><td>B734::EZY544
MD11::MPH978
A330:: LH5908
...</td><td>B744::SIA267
B744::KLM390</td></tr>
<tr><td>...</td><td>...</td><td>...</td><td>...</td><td>...</td></tr>
<tr><th colspan="5">Parameter and Initial Condition Specification</th></tr>
<tr><th colspan="2">System Type and Instance ID</th><th colspan="2">Parameter Settings</th><th>Initial Condition</th></tr>
<tr><td colspan="2">LRSATS1::B744::SIA267</td><td colspan="2">B744.xml</td><td>SIA267.xml</td></tr>
<tr><td colspan="2">LRSATS2::A330:: LH5908</td><td colspan="2">A330.xml</td><td>LH5908.xml</td></tr>
<tr><td colspan="2">...</td><td colspan="2">...</td><td>...</td></tr>
<tr><td colspan="2">LRSATS1::Atmosphere</td><td colspan="2">Meteo1.xml</td><td>N.A.</td></tr>
<tr><td colspan="2">LRSATS2::Atmosphere</td><td colspan="2">Meteo1.xml</td><td>N.A.</td></tr>
<tr><td colspan="2">LRSATS3::Atmosphere</td><td colspan="2">Meteo1.xml</td><td>N.A.</td></tr>
<tr><th colspan="5">Situational and Operational Context Description</th></tr>
<tr><td colspan="5">Both the aircraft B744::SIA267 and B744::KLM390 are operated by a single pilot who will fly the aircraft according a specified flight-plan. The trajectories of both these flight-plans are constructed such that after 15 minutes a loss of separation will occur. The aircrafts simulated controlled by the pseudo-pilot will also fly according their flight-plan. The pseudo-pilot aircraft trajectories will not cause any loss of separation if all aircraft fly their original flight-plan. They are only there to limit maneuver space as in a real airspace, for both the pilots in solving the separation loss conflict of their aircraft respectively.</td></tr>
<tr><th colspan="5">Behavioral Input and Scripting Specification</th></tr>
<tr><th>System Instance ID</th><th>Input Variable</th><th>Type</th><th colspan="2">Input File Reference</th></tr>
<tr><td>N.A.</td><td>N.A.</td><td><Script, Sample></td><td colspan="2">N.A.</td></tr>
<tr><th colspan="5">Termination Condition Specification</th></tr>
<tr><td colspan="5">IF
All occurred separation conflicts after t= 15 [min] for both the B744::SIA267 and B744::KLM390 aircraft are solved.
AND
Both the B744::SIA267 and B744::KLM390 aircraft are back on their original flight-plan trajectory.
THEN
The simulation execution is terminated</td></tr>
</table>

Table 6-4 Excerpt FASE Simulation System Configuration Specification

6.4.2 Real-World System Replication Knowledge Specification

This section presents the templates to specify the real-world system replication knowledge S_{rwr_s} as observed and registered during simulation execution (*Figure 6-5*). S_{rwr_s} contains both the real-world structural and behavioral data for each replicated real-world system $S^{rw}_{model_k} \in S^{exec}_{model}$, which are discussed next.

Real-World Structural Composition and Relationships

The simulation execution real-world structural composition and relationships in S_{rwr_s} is formed by a subset of the total real-world system model structure $M^{rw}_{model_{total}}$ (*Expression 6.25*). This set of knowledge is the simulation execution replicated counterpart (*Figure 6-4*) for the structural knowledge and data available for the actual real-world system in the fidelity referent knowledge base (Section 5.4.1). To be comparable with this referent knowledge-base the simulation execution real-world structural composition and relationship knowledge, like the simulation model knowledge specification, is specified in similar terms. Therefore, the specification templates comprise the next information areas that, except for the explicitly indicated differences, are identical to those discussed in Section 5.4.1:

- *Overall Structural Properties*: Specifies the overall structural properties of the total replicated real-world system during simulation execution. Like in the simulation model knowledge specification the total number of manifold representations is specified as an additional and important cardinal number (*Expression 6.26*).
- *Real-World System Model Topology Map*: Specifies the *topology map* of the total replicated real-world system during simulation execution.
- *Real-World System Model Characteristics Description List*: Specifies for each real-world system model $S^{rw}_{model_k}$ participating in the simulation execution its characteristics. This is practically realized by means of a list containing two pointers for each $S^{rw}_{model_k}$. One refers to the simulation model real-world system model characteristics knowledge area. The other one refers to the associated simulation system configuration S_{config_s}, where the parameter settings etc. for $S^{rw}_{model_k}$ be found.

For brevity the above templates are not stated and illustrated in detail here again.

Real-World Behavioral Data

The simulation execution real-world behavioral data in S_{rwr_s} is the replicated counterpart for the behavioral data available for each real-world system $S^{rw}_{ref_i} \in R_{ref}$ in the fidelity referent knowledge-base (*Figure 6-4*). Which means that the behavioral knowledge and data specification in S_{rwr_s} can be divided in the same four behavioral knowledge specification areas as for R_{ref} (Section 5.4.2). Likewise the mathematical equivalent (*Expressions 5.6* to *5.19*) for these knowledge areas developed in that same section also hold for this knowledge area in S_{rwr_s}. The major difference is that they now apply to the set S_{model}^{exec} and its constituent elements $S_{model_k}^{rw}$ instead. Except for explicitly indicated differences, the following simulation execution real-world behavioral data specification templates are identical to the those presented in Section 5.4.2:

- *Real-World System Model Interaction Causality Knowledge Specification*: Specifies the observed set of interaction chains (*Expression 5.6*) of the total replicated real-world system during simulation execution.
- *Non-casual Model Behavior Knowledge Specification*: Specifies for each $S_{model_k}^{rw} \in S_{model}^{exec}$ the observed set of variables ranges (*Expression 5.8*), aggregate variable relationships (*Expression 5.9*), and variable inter-relationships (*Expression 5.12*) during simulation execution.
- *Model Behavior Sample Specification*: Specifies for each $S_{model_k}^{rw} \in S_{model}^{exec}$ both the registered normal and complex behavior sample sets (*Expression 5.15*) during simulation execution.
- *Qualitative Model Behavior Specification*: Specifies the observed ordinary causal *(Expression 5.16*) and change-value causal relationships (*Expression 5.17 and 5.18*) for each $S_{model_k}^{rw} \in S_{model}^{exec}$ during simulation execution.

For brevity the above templates are not stated and illustrated in detail here again.

6.5 Simulation System Knowledge-Base: Complementary Knowledge Specification Templates

The previous sections discussed the core knowledge areas and associated templates of the simulation system knowledge-base, necessary for simulation fidelity assessment. In the next sections three complementary knowledge elements and specification templates will be presented:

- *Simulation Support System Specification*
- *Elicitation Process Knowledge Specification*
- *Additive and Management Knowledge Specification*

These elements provide complementary knowledge to properly use, maintain and manage the simulation system knowledge-base throughout the whole simulation development and validation process.

6.5.1 Simulation Support System Specification

Chapter 3 informally presented the simulation system as being composed of a simulation model, a configuration system and simulation support systems (*Figure 3-3*). The first two systems have already been discussed and specified in detail the previous section of this chapter. In order to complete the knowledge specification of the simulation system any information about the support systems $S_{support}$ should be specified. Theoretically simulation support systems can effect the simulation execution (*Expression 4.4*). However, in real-life, a proper simulation system design usually results in simulation support systems that have no or insignificant effects on the real-world system replication

S_{rwr_s} during a simulation execution. In case they do then the simulation system design needs to be reconsidered and is usually adjusted. Despite the important role in the overall operational usage of a simulation system, the unified fidelity framework simulation system knowledge-base doesn't therefore include any particular support system specification template. This is a practical assumption for most simulation systems without affecting the general applicability of S_{spec} in simulation fidelity assessment.

6.5.2 Elicitation Process Knowledge

The formal definition for S_{spec} comprises a high-level quintuple similar to the fidelity referent (*Expression 4.13 and 4.17*). Besides the specification of the actual elicited simulation system knowledge S_{appx} also the elicitation process E_{sim} must be specified. This includes also the set of constraints L_S^{elecit}, which apply to the elicitation of the simulation system knowledge along with the associated possible error δS_{appx} and uncertainty $U_\delta\left(S_{appx}\right)$ sources due to these limitations. It is stressed here that these knowledge elicitation limitations, errors and uncertainties differ from the simulation model and parameter data elicitation limitations, errors and uncertainties encountered during the simulation system development (Section 6.4). To illustrate the differences consider the next figure.

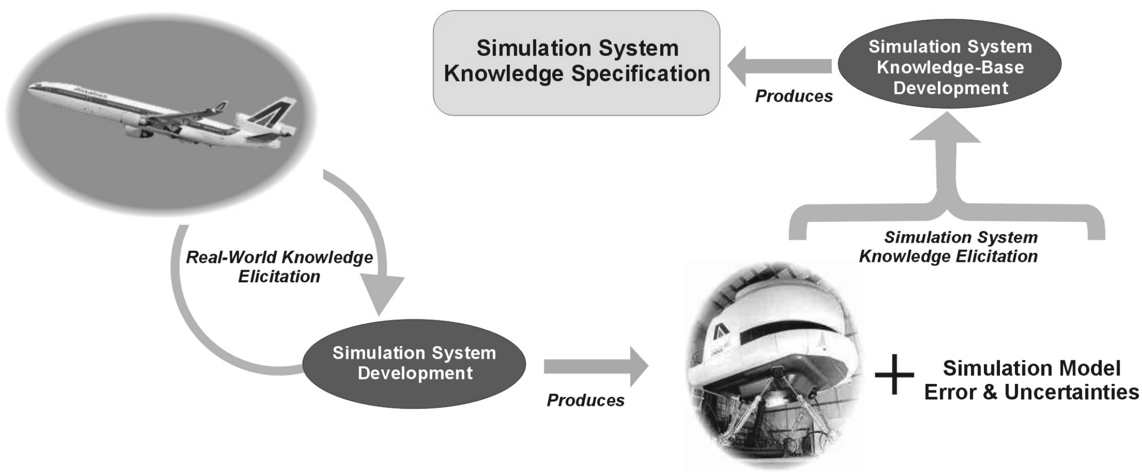


Figure 6-6 Simulation Development vs. Simulation Knowledge Elicitation

Since the simulation system elicitation process knowledge is similar to that of the fidelity referent elicitation process knowledge, the same templates are used as the ones developed in section 5.5 for the fidelity referent knowledge-base. Therefore, these templates are not stated and illustrated here again.

6.5.3 Additive & Management Knowledge

Similar to the fidelity referent knowledge-base the simulation system knowledge-base should also provide a well-defined body of additive knowledge elements for managing and maintaining the elicited simulation system knowledge S_{appx} throughout the simulation system live-cycle (Section 5.3). These additional elements provide essential information about the quality level, applicability, suitability and traceability of the simulation system knowledge-base contents. Which is necessary and useful information for fidelity assessment activities performed during the simulation system development and validation process. Except for the omission of the *Applicability and Status Information* area, the specification templates are identical to the ones developed and incorporated in the fidelity referent knowledge-base (Section 5.3). For brevity these four knowledge areas are summarized below and are not further illustrated here in detail again:

- *Identification and Management Information*: Specifies the simulation system knowledge-base identification, point of contact and management details such as revision history.
- *Developer and Validation Agent Information*: Lists all people whom were involved in the development and validation of the simulation system knowledge-base.
- *Used Knowledge and Data Sources*: Lists all knowledge and data sources that have been utilized in the simulation system knowledge-base development such as simulation system design documents etc.
- *Utility Knowledge and Data*: Specifies any utility knowledge and data necessary to fully understand or use the specified simulation system knowledge S_{appx} such as axes systems, mathematical notations, etc.

6.6 Summary

By definition not only real-world reference knowledge (R_{ref}) but also any knowledge regarding the replication of this real-world by the simulation system (S_{spec}) is required to be able to properly characterize and measure the level of simulation fidelity (Chapter 4). In this chapter a knowledge-base architecture has been developed to practically specify this simulation system knowledge (S_{appx}) within the unified fidelity framework. The simulation system knowledge-base architectural design centers round a set of structured generic and linked knowledge specification templates and is complemented with mathematical formulations underlying its constituent elements. Since the knowledge about the real-world system replication by a simulation system must be specified in similar terms and structured format as the real-world reference knowledge, the fidelity referent knowledge-base design has been chosen as the bases for the simulation system knowledge-base architectural design (Chapter 5). The commonalities and differences between both knowledge-base architectures have been presented in this chapter along with their rationale. The simulation system specification templates are grouped in the following major subsets:

- *Simulation Model Knowledge Specification*: In summary area provides the practical means for the static *model structure fidelity* qualification and quantification of each constituent simulation model component plus indicators to solve simulation behavioral fidelity issues up front or during simulation testing. It includes knowledge templates for simulation model assumptions, data uncertainties, the actual hardware and software.
- *Simulation Execution Knowledge Specification*: Summarized this area specifies a set of registered replications of the real-world during simulation execution plus the belonging simulation system configuration setting. In other words these

templates contain all available knowledge necessary for assessing the level of simulation fidelity from a dynamic *behavioral perspective*.

- *Complementary Knowledge Specifications*: Summarized this area specifies the complementary knowledge to properly use, maintain and manage the simulation system knowledge-base throughout the simulation development and validation process. It addresses simulation support system, elicitation, additive and management knowledge specification.

The union of the simulation system configuration setting and the simulation model specification provides the complete static structural representation of the real-world that can be compared with the real-world structural knowledge specification part of the fidelity referent knowledge base (Section 5.4.1). Similarly, the specification of the registered real-world behavior during simulation contains the information about the behavioral representation of the real-world by the simulation system. This knowledge can be compared with the actual real-world behavioral data specification part of the fidelity referent knowledge base (Section 5.4.2). Both the fidelity referent and the simulation system knowledge-base are exploited in Chapter 7 for the development of a possible set of fidelity evaluator functions that qualify and quantify the level of simulation fidelity.

Both case-study applications (Appendix C) proved that the developed architecture of structured knowledge templates cover the most elementary and recurring specification elements required in any practical simulation fidelity assessment. Obviously, these knowledge-base templates will not directly suite or cover every aspect of the wide spectrum of simulation application problems. However, like the fidelity referent, the simulation system knowledge-base templates have been constructed such that they allow for easy tailoring and extending in order to fully suit any other specific application or problem domain. In this regard there are several issues that require attention and additional research:

- Application and problem domain specific model and knowledge specification languages to refine, tailor and extend the in this chapter described templates. This will improve the understandability and applicability of the current realization within the simulation fidelity assessment process for a larger public.
- For the same reasons as given in section 5.6 the development and implementation of dedicated automated tools that support the simulation knowledge specification throughout the simulation system life cycle is mandatory. The focus should be on the integration with other fidelity and simulation development tools to facilitate a cost-effective application of the complete unified fidelity framework approach.
- Research to sensitivity analysis methodologies and tools to solve the current issues and reduce expenses for fidelity assessment (Section 6.3.3).

7 Unified Fidelity Framework: Fidelity Metrics and Measurement Methods

7.1 Introduction

As formally defined by in chapter 4 it is the set of fidelity evaluator functions that actually measure the difference between the real-world system reference knowledge and replicated real-world system knowledge. This knowledge is respectively specified in the fidelity referent and the simulation system knowledge-base structure, which have extensively been discussed in the previous two chapters. Fidelity evaluator functions come in a wide-range of different flavors. Moreover, their type, complexity and number that have to be used to populate a practical fidelity evaluator function set $\tilde{C}_{\Delta_{RS}}$ (*Expression 4.19*) varies with the simulation application and problem domain at hand (Chapter 8). Therefore, the objective of this chapter is to present a basic taxonomy with the most common and elementary fidelity evaluator functions that can be used for simulation fidelity qualification and quantification. These fidelity evaluator functions directly derive from the unified fidelity framework referent and simulation system knowledge-base specification elements. This taxonomy is not fully exhaustive but serves as a starting point for developing more exhaustive and application or problem specific fidelity evaluator functions taxonomies. Such development is a study itself, which is beyond the scope of this thesis. Most of these evaluator functions have been applied in the two aerospace simulation case studies (Appendix C).

Despite their varying nature, fidelity evaluator functions can be classified in either quantitative or qualitative methods (Section 4.3.2). Furthermore, the basis for all evaluator functions is formed by the eight fidelity characterization concepts developed in Section 4.3.3. These concepts further classify the fidelity evaluator functions in areas in which the difference between structural and behavioral knowledge pairs contained within the referent and simulation system knowledge-base can qualified and/or quantified. Section 7.2 will focus on the various classes of quantitative structural fidelity evaluator functions. Quantitative behavioral fidelity evaluator functions are the topic of Section 7.3. Finally Section 7.4 will touch upon subject matter expert based fidelity metrics and measurement methods for both structural and behavioral knowledge pairs.

7.2 Quantitative Structural Oriented Metrics and Methods

Structural oriented fidelity metrics and methods focus on the assessment of the deviation between the structural composition and relationships of the simulation system representation of the simuland with respect to the actual real world. In practice this means the comparison between the specified structural knowledge section pairs found in the fidelity referent knowledge base and the simulation system specification knowledge base. The purpose of the in this section presented metrics and methods is to objectively characterize and quantify the simulation fidelity in terms of resolution or the level of detail used in a simulated representation of the real-world (Section 4.3.3). These resolution metrics and methods are further divided in two subclasses, system-level and property-level. System-level metrics and methods attempt to characterize the difference in the level of aggregation, complexity and completeness of the represented hierarchy of real-world systems and interactions. The property-level metrics and methods try to characterize the difference in complexity and completeness for each represented real-world system in terms of their functional capabilities, parameter, input, output and state properties. Both classes are discussed in the next two subsections.

As formalized in sections 6.3 and 6.4 the configuration settings fed to the simulation model determine the final structural composition of the real-world system as replicated during each simulation execution. Therefore, the majority of the metrics discussed in this section can be applied to assess both the structural simulation fidelity during simulation execution and the fidelity capabilities of the total simulation model. For reasons of brevity and readability these resolution metrics are discussed in this section from the simulation execution perspective. However, all here presented resolution metrics and methods are, unless explicitly stated otherwise, applicable to both simulation system fidelity assessment activities (Chapter 9).

7.2.1 System-Level Resolution Quantification

System-level resolution metrics are the most coarse-grained and top-level form of measuring the difference between the real-word and its simulated counterpart. These metrics describe from a system level perspective, what real-world systems and their interactions are left out with respect to the real world structural hierarchy in the fidelity referent (Section 5.4.1). This can also be assessed qualitatively by means of visually comparing the real-world system reference topology map and the real-world system model topology map. Although such qualitative evaluation certainly complements the simulation system fidelity assessment, the use of the quantitative resolution metrics is preferred since they better facilitate the required more formal and mathematical rigorous treatment of simulation fidelity.

Total Real-World System Scope Difference

The first resolution evaluator function operates on the referent real-world system set R_w^{ref} and the executable real-world system model set $M_{exec}^{rw} \subseteq M_{model_{total}}^{rw}$ in S_{appx}. This two-dimensional fidelity evaluator function comprises the next two metrics:

$$
\begin{aligned}
c_{\Delta_{res\_1}}(R_w^{ref}, S_{appx})_{d_1} &= \left(R_w^{ref} \setminus M_{exec}^{rw}\right) \\
c_{\Delta_{res\_1}}(R_w^{ref}, S_{appx})_{d_2} &= \left(M_{exec}^{rw} \setminus R_w^{ref}\right)
\end{aligned}
\qquad (7.1)
$$

The first metric in *expression 7.1*, labeled d_1, specifies the set of all real-world systems not present in the scope of the executable simulation model. This metric provides the lowest level evaluation of which and how many real-world systems in total haven't been represented during simulation execution. In case the second metric results in a non-empty set this could indicate that the simulation system models non-existent imaginary or material real-world system or that the fidelity referent knowledge is incomplete. Their proportional inverse scaled equivalents are defined using their set cardinalities (Section 4.4):

$$(d_1)^{-1} = 1 - \frac{n\left(R_w^{ref} \setminus M_{exec}^{rw}\right)}{n(R_w^{ref})_{system}}$$

$$(d_2)^{-1} = 1 - \frac{n\left(M_{exec}^{rw} \setminus R_w^{ref}\right)}{n(R_w^{ref})_{system}} \tag{7.2}$$

A value of one for each element in *expression 7.2* signifies a perfect conformance with the fidelity referent in terms of the total scope of the represented real-world systems. Values lower than one indicate a lesser conformance to this referent.

Partial Real-World System Scope Difference and Cardinality

In case the first output element in *expression 7.2* has a value other than one, it is known some real-world systems are not represented. However, their locations within the real-world structural hierarchy are not known. Since such knowledge is essential for judging suitability and validity of a simulation system more detailed structural fidelity evaluations are necessary. This requires fidelity evaluator functions that assess the conformance to the referent for particular structural area's and decomposition levels of interest. Due to their hierarchical composition this is something that intuitively follows from both the referent and the simulation system knowledge specification. Basically, this means that a similar fidelity evaluator function as in *expression 7.1* is applied but now to subsets of the referent real-world system set $R_{w_i}^{ref} \subseteq R_w^{ref}$ and the executable real-world system model set $M_{exec_i}^{rw} \subseteq M_{exec}^{rw}$ that cover the same real-world area A_i of interest. This is more clearly illustrated in the next Venn diagram.

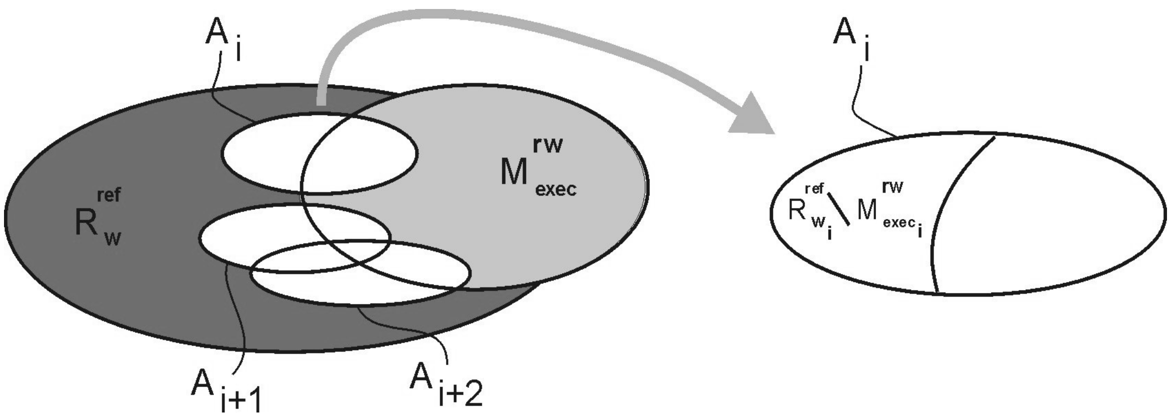


Figure 7-1 Partial Real-World System Scope Difference Venn diagram

The fidelity evaluator function for assessing a partial real-world system scope is formally expressed by the next one-dimensional metric:

$$c_{\Delta_{res\_2}}(R_w^{ref}, S_{appx})_{d_1} = \left(R_{w_i}^{ref} \setminus M_{exec_i}^{rw}\right) \tag{7.3}$$

Expression 7.3 can be extended to a multi-dimensional vector for as many as necessary real-world area's A_i of interest. The associated proportional inverse scaled equivalent is defined using set cardinalities and looks as follows:

$$\left(d_1\right)^{-1} = 1 - \frac{n\left(R_{w_i}^{ref} \setminus M_{exec_i}^{rw}\right)}{n(R_{w_i}^{ref})} \tag{7.4}$$

Again a value of one signifies a perfect conformance with the fidelity referent but now for partial scope of represented real-world systems. Values lower than one indicate a lesser conformance to a fidelity referent area of interest.

Difference in Maximum and Average Decomposition Depth

To gain insight in the real-world system decomposition levels that are represented by the executable simulation model compared to the fidelity referent, the maximum and average decomposition depth metrics can be used. These metrics build upon the specified maximum and average branch length in both the referent and the simulation system knowledge specification (*Expressions 5.4 and 5.5*). The fidelity evaluator function containing both metrics is the following:

$$\begin{aligned} c_{\Delta_{res\_3}}(R_w^{ref}, S_{appx})_{d_1} &= B_{length_{max}}^{ref} - B_{length_{max}}^{exec} \\ c_{\Delta_{res\_3}}(R_w^{ref}, S_{appx})_{d_2} &= B_{length_{ave}}^{ref} - B_{length_{ave}}^{exec} \end{aligned} \tag{7.5}$$

In here the superscript *ref* and *exec* refer to the respectively the branch lengths of R_w^{ref} and M_{exec}^{rw}. The proportional inverse scaled equivalents are defined by:

$$\begin{aligned} \left(d_1\right)^{-1} &= 1 - \frac{d_1}{B_{length_{max}}^{ref}} \\ \left(d_2\right)^{-1} &= 1 - \frac{d_2}{B_{length_{ave}}^{ref}} \end{aligned} \tag{7.6}$$

Again a value of one signifies identical branch lengths. Values lower than one indicate a lesser conformance to a fidelity referent area of interest. In case a value lager than one is returned either indicates that the simulation system models non-existent imaginary or that the material real-world system or that the fidelity referent knowledge is incomplete.

Difference in Decomposition Width and Bifurcation

Both the maximum and average decomposition depth metrics give a top-down characterization of the structural hierarchy completeness. Another characterization of structural completeness of importance is the difference in decomposition width and bifurcation evaluation. Together, they indicate the degree of aggregation is applied between the interactive behaviors of separate composing subsystems within the executable simulation model. Usually, such aggregation involves approximations and assumptions of interactive behaviors of the separate subsystems. This may result in obfuscation or even a loss of certain desired interactive behaviors in the overall represented parent system [119]. The difference in decomposition width and bifurcation build upon the specified total number of leafs and forks in the structural hierarchy of both the referent and the simulation system knowledge specification (Section 5.4.1). The fidelity evaluator function containing both metrics is give by:

$$
\begin{aligned}
c_{\Delta_{res\_4}}(R_w^{ref}, S_{appx})_{d_1} &= n\left(R_w^{ref}\right)_{forks} - n\left(M_{exec}^{rw}\right)_{forks} \\
c_{\Delta_{res\_4}}(R_w^{ref}, S_{appx})_{d_2} &= n\left(R_w^{ref}\right)_{leafs} - n\left(M_{exec}^{rw}\right)_{leafs}
\end{aligned}
\tag{7.7}
$$

The proportional inverse scaled equivalents are simply calculated by:

$$
\begin{aligned}
\left(d_1\right)^{-1} &= 1 - \frac{d_1}{n\left(R_w^{ref}\right)_{forks}} \\
\left(d_2\right)^{-1} &= 1 - \frac{d_2}{n\left(R_w^{ref}\right)_{leafs}}
\end{aligned}
\tag{7.8}
$$

In case an element of *expression 7.8* returns a value lager than one either indicates that the simulation system models non-existent imaginary or that the material real-world systems or that the fidelity referent knowledge is incomplete.

Total and Partial Real-World Interaction Scope Difference

Even if all subsystems are present at decomposition level, it doesn't have to imply that every interaction relationship between each subsystem is present in the executable simulation model. To evaluate differences in interaction relationships the basic resolution metrics of total interaction scope difference and cardinality can be used, which looks as follows

$$
c_{\Delta_{res\_5}}(R_w^{ref}, S_{appx})_{d_1} = \bigcup_{i=1}^{n\left(R_w^{ref}\right)_{system}} I_i^{rw} \setminus \bigcup_{j=1}^{n\left(M_{exec}^{rw}\right)_{system}} I_j^{exec}
\tag{7.9}
$$

In here $I_i^{rw} \in S_{ref_i}^{rw}$ and $I_j^{exec} \in S_{model_j}^{rw}$ are the system interaction sets as defined by *expression 3.27* belonging to either the real-world reference system and the real-world system model.

The proportional inverse scaled equivalent of *expression 7.9* is defined using both set cardinalities (*Expression 5.3*):

$$
\left(d_1\right)^{-1} = 1 - \frac{n\left(R_w^{ref}\right)_{interaction} - n\left(M_{exec}^{rw}\right)_{interaction}}{n(R_w^{ref})_{interaction}}
\tag{7.10}
$$

A value of one for each element in *expression 7.10* means a perfect conformance with the fidelity referent in terms of the total scope of the represented real-world interactions.

A value lower than one signifies that some real-world interactions are not represented and the executable simulation model has thus lesser conformance to the referent. Values larger than one indicates either modeled but non-existent imaginary interactions or an incomplete fidelity referent. Like for the total real-world system scope difference (*Expression 7.2*) their locations within the real-world structural hierarchy are not known. This essential fidelity knowledge can be evaluated by applying an evaluator function similar to *expression 7.9* to subsets of the referent real-world system set $R^{ref}_{w_i} \subseteq R^{ref}_{w}$ and the executable real-world system model set $M^{rw}_{exec_i} \subseteq M^{rw}_{exec}$ that cover the same real-world area A_i of interest (*Figure 7-1*). These resulting so-called partial real-world interaction scope difference metrics can now be used to assess the conformance to the referent for a particular structural sub area and decomposition level of interest.

7.2.2 Property-Level Resolution Quantification

Property-level resolution metrics and methods are a more low-level form of measuring the executable simulation model resolution than the previously discussed system-level resolution metrics. These metrics characterize for each represented real-world system in the executable simulation model the difference in completeness in terms of their multiplicity, functional capabilities, parameter, input, output and state properties.

Real-World System Multiplicity Difference

A real-world reference system class can have multiple instances in with the real world. The possible numbers of these instances are specified in the fidelity referent with a multiplicity set of integers $N^{rw}_{multi_i}$ (Section 5.4.1). To quantify the conformance of the number of $n^{exec}_{multi_i}$ instances for the i^{th} corresponding real-world system model within the executable simulation model the following multiplicity difference evaluator function is defined:

$$c_{\Delta_{res\_6}}(R^{ref}_{w}, S_{appx})_{d_1} = \begin{Bmatrix} n^{exec}_{multi_i} \in N^{rw}_{multi_i} \Rightarrow d_1 = 0 \\ \\ n^{exec}_{multi_i} \notin N^{rw}_{multi_i} \Rightarrow d_1 = k \cdot \left| n^{rw}_{multi_{closest}} - n^{exec}_{multi_i} \right| \\ with \\ k \in \mathbb{N} \wedge n^{rw}_{multi_{closest}} \in N^{rw}_{multi_i} \end{Bmatrix} \tag{7.11}$$

In *expression 7.11* is $n^{rw}_{multi_{closest}}$ the element in $N^{rw}_{multi_i}$ with the closest distance to $n^{exec}_{multi_i}$. Furthermore, by definition the multiplicity of a represented real-world system must always be one or larger and k can be chosen freely. An example of the multiplicity difference fidelity evaluator function output for a given set N^{rw}_{multi} is graphical illustrated in *Figure 7-2* at the next page. The proportional inverse scaled equivalent is given by:

$$\left(d_1\right)^{-1} = 1 - \frac{d_1}{k \cdot d_1 + 1} \tag{7.12}$$

A value of one in *expression 7.12* means that multiplicity of the i^{th} real-world system model is located within the multiplicity set of the corresponding reference real-world system. Values other than one signify that the multiplicity is outside this reference multiplicity set. In practice e*xpressions 7.11* and *7.12* have to be applied to each element $S^{rw}_{model_i} \in M^{rw}_{exec}$. The resulting vector with the outcomes then characterizes the multiplicity difference of the complete simulation model.

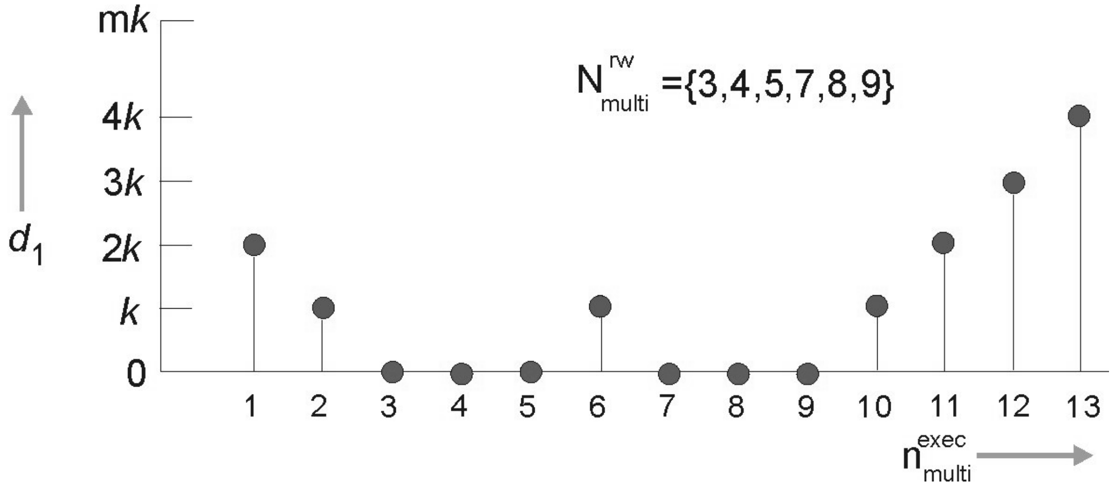


Figure 7-2 Example Multiplicity Difference Evaluator Function

Representational and Behavior Property Scope Difference

When a real-world system is represented in the executable simulation model this not necessary implies that the corresponding real-world system model replicates all its representational and behavioral properties (Section 3.4.5). For various reasons only a portion of the variable properties might be present. To characterize the difference in input (*U*), output (*Y*), state (*Q*) and parameter (*P*) property scope between the i^{th} real world reference system and the corresponding real-world system model the following eight-dimensional fidelity evaluator function is used:

$$
\begin{aligned}
&c_{\Delta_{res\_7}}(R^{ref}_w, S_{appx})_{d_1} = \left(U^{ref}_i \setminus U^{exec}_i\right) \quad c_{\Delta_{res\_7}}(R^{ref}_w, S_{appx})_{d_2} = \left(U^{exec}_i \setminus U^{ref}_i\right)\\
&c_{\Delta_{res\_7}}(R^{ref}_w, S_{appx})_{d_3} = \left(Y^{ref}_i \setminus Y^{exec}_i\right) \quad c_{\Delta_{res\_7}}(R^{ref}_w, S_{appx})_{d_4} = \left(Y^{exec}_i \setminus Y^{ref}_i\right)\\
&c_{\Delta_{res\_7}}(R^{ref}_w, S_{appx})_{d_5} = \left(P^{ref}_i \setminus P^{exec}_i\right) \quad c_{\Delta_{res\_7}}(R^{ref}_w, S_{appx})_{d_6} = \left(P^{exec}_i \setminus P^{ref}_i\right)\\
&c_{\Delta_{res\_7}}(R^{ref}_w, S_{appx})_{d_7} = \left(Q^{ref}_i \setminus Q^{exec}_i\right) \quad c_{\Delta_{res\_7}}(R^{ref}_w, S_{appx})_{d_8} = \left(Q^{exec}_i \setminus Q^{ref}_i\right)
\end{aligned}
\tag{7.13}
$$

In here $\left(U^{ref}_i, Y^{ref}_i, Q^{ref}_i, P^{ref}_i\right) \in S^{rw}_{ref_i}$ and $\left(U^{exec}_i, Y^{exec}_i, Q^{exec}_i, P^{exec}_i\right) \in S^{rw}_{model_i}$. When an even numbered metric in *expression 7.13* results in a non-empty set this could indicate that the simulation system models non-existent real-world system representational and behavioral properties or that the fidelity referent knowledge is incomplete. The proportional inverse scaled equivalents for *expression 7.13* are defined using their set cardinalities as given by *Expression 7.14.*

$$
\begin{aligned}
&(d_1)^{-1} = 1 - \frac{n\left(U_i^{ref} \setminus U_i^{exec}\right)}{n\left(U_i^{ref}\right)} && (d_2)^{-1} = 1 - \frac{n\left(U_i^{exec} \setminus U_i^{ref}\right)}{n\left(U_i^{ref}\right)} \\
&(d_3)^{-1} = 1 - \frac{n\left(Y_i^{ref} \setminus Y_i^{exec}\right)}{n\left(Y_i^{ref}\right)} && (d_4)^{-1} = 1 - \frac{n\left(Y_i^{exec} \setminus Y_i^{ref}\right)}{n\left(Y_i^{ref}\right)} \\
&(d_5)^{-1} = 1 - \frac{n\left(P_i^{ref} \setminus P_i^{exec}\right)}{n\left(P_i^{ref}\right)} && (d_6)^{-1} = 1 - \frac{n\left(P_i^{exec} \setminus P_i^{ref}\right)}{n\left(P_i^{ref}\right)} \\
&(d_7)^{-1} = 1 - \frac{n\left(Q_i^{ref} \setminus Q_i^{exec}\right)}{n\left(Q_i^{ref}\right)} && (d_8)^{-1} = 1 - \frac{n\left(Q_i^{exec} \setminus Q_i^{ref}\right)}{n\left(Q_i^{ref}\right)}
\end{aligned}
\tag{7.14}
$$

A value of one for an element in *expression 7.14* signifies a perfect conformance with the real-world reference system. A value lower than one indicates a lesser representational and behavioral conformance. Applying *expressions 7.13* and *7.14* to each element $S_{model_i}^{rw} \in M_{exec}^{rw}$ results in the overall characterization of the difference in representational and behavioral scope of the complete executable simulation model.

Functional Capability Scope Difference

The last property-level resolution evaluator function is the functional capability scope difference. This fidelity evaluator function simply quantifies the difference in functional capabilities (*Expression 3.13*) between the i^{th} real world reference system and the corresponding real-world system model present in the simulation model (Section 3.4.4). The two-dimensional fidelity evaluator function for this task comprises the following two metrics:

$$
\begin{aligned}
&c_{\Delta_{res\_8}}(R_w^{ref}, S_{appx})_{d_1} = \left(F_{cap_i}^{rw} \setminus F_{cap_i}^{exec}\right) \\
&c_{\Delta_{res\_8}}(R_w^{ref}, S_{appx})_{d_2} = \left(F_{cap_i}^{exec} \setminus F_{cap_i}^{rw}\right)
\end{aligned}
\tag{7.15}
$$

Where $F_{cap_i}^{rw}$ and $F_{cap_i}^{exec}$ are respectively the functional capability set of the i^{th} pair of corresponding real-world reference system $S_{ref_i}^{rw}$ and system model $S_{model_i}^{rw}$. Another more complex and possible useful metric could be developed based on the means-end hierarchy of overall and nested functional capabilities (Section 3.4.6). However this isn't further exploited within this thesis.

Similar to the previous fidelity evaluator function the proportional inverse scaled equivalents for *expression 7.15* are defined using their set cardinalities:

$$\left(d_1\right)^{-1} = 1 - \frac{n\left(F_{cap_i}^{rw} \setminus F_{cap_i}^{exec}\right)}{n\left(F_{cap_i}^{rw}\right)} \qquad \left(d_2\right)^{-1} = 1 - \frac{n\left(F_{cap_i}^{exec} \setminus F_{cap_i}^{rw}\right)}{n\left(F_{cap_i}^{rw}\right)} \qquad (7.16)$$

For the first element in *expression 7.16* a value of one signifies a perfect conformance with the real-world reference system. Values lower than one mean lesser conformance. A value other than one for the second element indicates either that the simulation system models non-existent real-world system functional capabilities or that the fidelity referent knowledge is incomplete.

7.3 Quantitative Behavioral Oriented Metrics and Methods

Behavioral oriented fidelity metrics and methods focus on the assessment of the difference between the simulation system behavioral representations of real world with respect to the actual real world behaviors. The basis for these metrics and methods is the pair-wise comparison between the specified behavioral knowledge found in the corresponding fidelity referent knowledge base and the simulation system specification knowledge base sections. The objective of the in this section presented metrics and methods is to objectively characterize the simulation fidelity in terms of error and accuracy of the behavioral replication of a real-world system (Section 4.3.3). These error and accuracy metrics and methods are catalogued and discussed in this section according the four sets of behavioral knowledge specification areas (Section 5.4.2): real-world system non-causal behavior, behavior samples, ordinary causal and logical relationships, and interaction causality. Although all of these behavioral metrics are commonly used for assessing the behavioral fidelity during simulation execution some can be applied to assess the behavioral fidelity capabilities of the total simulation model in the earlier stage of simulation system development (Chapter 9). For reasons of brevity and readability the here presented behavioral accuracy metrics are discussed from the simulation execution perspective.

7.3.1 Real-World System Non-Causal Behavior Accuracy

Non-causal behavior accuracy metrics evaluate how well the simulation system's behavioral replication of a real-world system conforms to the actual non-causal relationships that hold for such a real-world system. These metrics provide a global form of measuring the difference between the real-world system behavior and its simulated counterpart, which has utility in assessing the accuracy of the predictive capabilities of a simulation system.

System Variable Range Error and Accuracy

The system variable range set (Expression 5.8) provides the known physical range in which the value of a real-world system input, output, parameter, and state variable must reside. The two-dimensional evaluator function for this purpose evaluates both the frequency of the violations and the total degree of violation. This total degree of violation is a cumulative metric, which sums the minimal distance between the observed value of a simulated real-world system variable and the range specified by the fidelity referent for every violation during simulation execution (*Figure 7-3*).

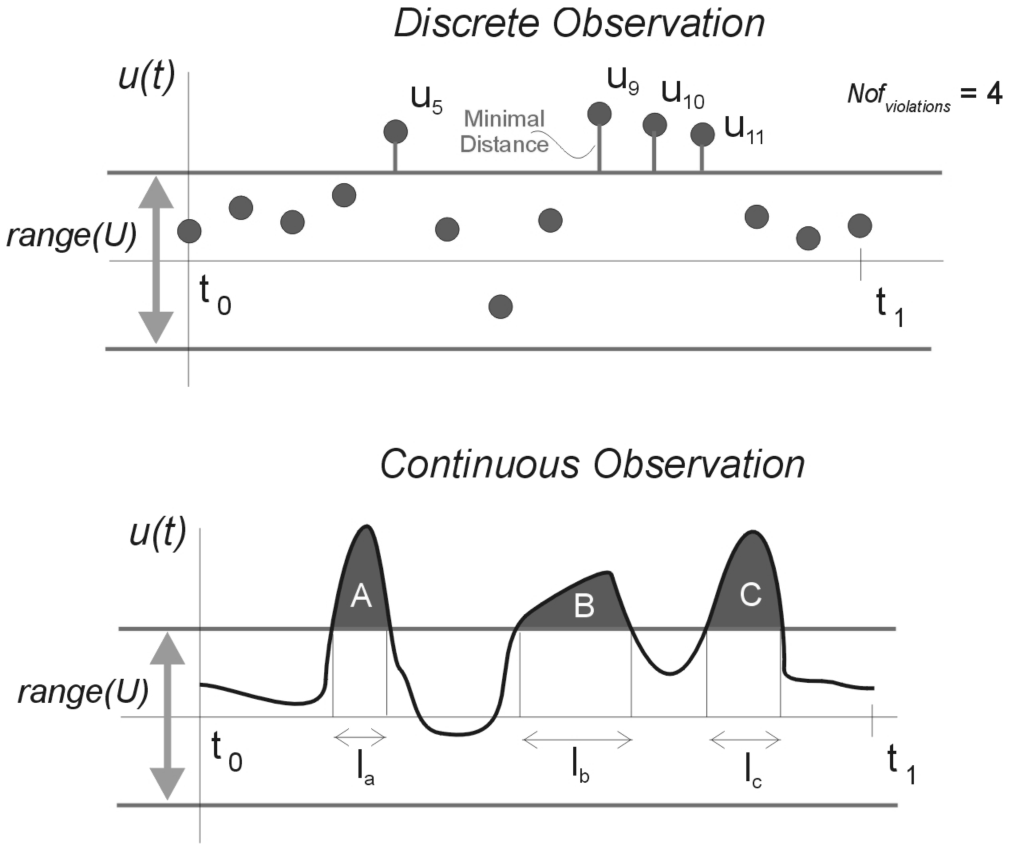


Figure 7-3 2D Examples of Range Error and Accuracy

Formally the system variable range evaluator function applied to an input variable of the i^{th} real-world system looks as follows:

$$c_{\Delta_{ac\_1}}(R_w^{ref}, S_{appx})_{d_1} = \sum_{t=t_0}^{t_1} f_{dist}\left(range_{u_{s_i}}\left(U_s^{rw}\right), u_{s_i}^{exec}\left(t\right)\right)$$

$$c_{\Delta_{ac\_1}}(R_w^{ref}, S_{appx})_{d_2} = \frac{Nof_{violations}}{n\left(\omega_{\langle t_0, t_1 \rangle}\right)} \qquad (7.17)$$

Here is $\omega\langle t_0, t_1 \rangle$ the observation sequence (Section 3.4.3) and $Nof_{violations}$ the observed number of discrete violations within this timeframe. The function f_{dist} is defined as:

$$f_{dist}\left(range_{u_{s_i}}\left(U_s^{rw}\right), u_{s_i}^{exec}\left(t\right)\right) = \begin{cases} 0 & \forall u_{s_i}^{exec}\left(t\right) \in range_{u_{s_i}}\left(U_s^{rw}\right) \\ \left|U_{s_{closest}}^{rw} - u_{s_i}^{exec}\left(t\right)\right|, & \forall u_{s_i}^{exec}\left(t\right) \notin range_{u_{s_i}}\left(U_s^{rw}\right) \end{cases} \qquad (7.18)$$

In here is $range_{u_{s_i}}\left(U_s^{rw}\right)$ the range of input variable $u_{s_i} \in U_s^{rw}$ of the i^{th} real-world reference system and $u_{s_i}^{exec}\left(t\right)$ the corresponding observed value at time stamp t during simulation execution. Furthermore, $U_{s_{closest}}^{rw}$ is the element in $range_{u_{s_i}}\left(U_s^{rw}\right)$ with the closest distance to $u_{s_i}^{exec}(t)$. In case the observation is performed on a continuous time base, the first output element of *expression 7.17* becomes an integral that calculates the area or an n-dimensional volume outside the specified variable range (*Figure 7-3*). The second element then becomes the total percentage of time the simulated variable violates the specified real-world system variable range within the observation time frame. The proportional inverse scaled equivalent of *expression 7.17* is given by:

$$(d_1)^{-1} = 1 - \frac{d_1}{d_1 + 1} \qquad (d_2)^{-1} = 1 - \frac{d_2}{d_2 + 1} \tag{7.19}$$

Here a value of one means that the range is not violated. A value smaller than one indicates a certain degree of violation. This is in accordance with the definition of accuracy developed in section 4.3.3. Therefore *Expression 7.19* is considered to be a measure of system variable range accuracy. As a final remark, *Expression 7.17*, *7.18* and *7.19* can and must also be applied to the real-world system output (*Y*), parameter (*P*) and state (*Q*) variables. For brevity reasons these are not presented in this thesis.

Aggregated System Variables Error and Accuracy

Within the structural hierarchy of an executable real world system model set the aggregate variable relationships between a represented real world system model and its child subsystems must reflect those specified in the fidelity referent. Each aggregate variable relationship is formally defined by a Boolean expression over time with a prototype given by *Expression 5.10* in section 5.4.2. During simulation execution it is evaluated how well the executable real world system model preserves this relationship by means of the next fidelity evaluator function:

$$c_{\Delta_{ac\_2}}(R_w^{ref}, S_{appx})_{d_1} = \left\{ \begin{array}{l} p_{RL_{aggregate}}\left(t_{exec}\right)_m : RL_{aggregate}\left(...\right)_m^{exec} \\ is\ true \Rightarrow d_1 = 0 \\ \\ p_{RL_{aggregate}}\left(t_{exec}\right)_m : RL_{aggregate}\left(...\right)_m^{exec} \\ is\ false \Rightarrow d_1 = \dfrac{n\left(\omega_{\langle t_0, t_1\rangle}\right) - n(T_{p_{RL}})}{n\left(\omega_{\langle t_0, t_1\rangle}\right)} \end{array} \right\} \tag{7.20}$$

Inhere $p_{RL_{aggregate}}\left(t_{exec}\right)_m$ is the proposition function on the m^{th} aggregate variable relationships expression $RL_{aggregate}\left(...\right)_m$ in the fidelity referent whose truth is tested for the corresponding real-world system model variables during simulation execution. Furthermore, $T_{p_{RL}}$ is the truth set containing the time stamps for which the proposition function is true [78]. If the executable simulation model conforms to this aggregate variable relationships expression for the whole observation time sequence $\omega\langle t_0, t_1\rangle$ the fidelity evaluator function returns zero. In the event that during this observation time frame the aggregate variable relationships expression proposition is violated the fidelity evaluation function returns a normalize ratio of the number occurred violations. For observations with continuous time-bases this ratio becomes the total percentage of time the aggregate variable relationships expression proposition is false. Obviously, this fidelity evaluator function must be applied to all aggregate variable relationships specified in the fidelity referent to acquire the complete picture. The proportional inverse scaled equivalent of *expression 7.20* is given by:

$$\left(d_1\right)^{-1} = 1 - \frac{d_1}{d_1 + 1} \tag{7.21}$$

Here a value of one means that the executable simulation model accurately represents the aggregate variable relationship. A value smaller than one indicates a certain degree of mismatch. Similar to *Expression 7.19* also this expression is considered to be a measure of accuracy.

Often aggregate variable relationships are directly implemented within the simulation model, which means that the proposition of expression 7.17 can also be evaluated by inspection without the execution of the simulation model. Providing that the simulation model correctly implements the system aggregate relationship the outcome of the fidelity evaluator function will be zero.

Interrelated System Variables Error and Accuracy

The last non-causal behavior fidelity evaluator function assesses the preservation of non-causal real-world system variables (*U*, *Y*, *Q* and *P*) interrelationships by the simulation system during simulation execution. These real-world system variables interrelationships are also specified in the fidelity referent by means of a set of Boolean expressions (*Expressions 5.11* and *5.12*) and are evaluated during simulation execution by means of the next fidelity evaluator function:

$$c_{\Delta_{ac\_3}}(R_w^{ref}, S_{appx})_{d_1} = \left\{ \begin{array}{l} p_{RL_{system}}\left(t_{exec}\right)_k : RL_{system}\left(\ldots\right)_k^{exec} \\ is\ true \Rightarrow d_1 = 0 \\ \\ p_{RL_{system}}\left(t_{exec}\right)_k : RL_{system}\left(\ldots\right)_k^{exec} \\ is\ false \Rightarrow d_1 = \dfrac{n\left(\omega_{\langle t_0, t_1\rangle}\right) - n(T_{p_{RL}})}{n\left(\omega_{\langle t_0, t_1\rangle}\right)} \end{array} \right\} \tag{7.22}$$

In *expression 7.22* is $p_{RL_{system}}\left(t_{exec}\right)_k$ the proposition function on the k^{th} system variable interrelationships expression $RL_{system}\left(\ldots\right)_k$ in the fidelity referent whose truth is tested for the corresponding real-world system model variables during simulation execution. This evaluation works identical as for the previously discussed aggregate variable relationships expression proposition and the proportional inverse scaled equivalent for *Expression 7.22* is also identical to *Expression 7.21*. Therefore this discussion is not repeated here again.

7.3.2 Real-World System Behavior Sample Accuracy

The most common known and applied method to determine the behavior difference between a real-world system and its simulated counterpart, is the quantitative comparison of their corresponding behavior samples. The objective of this method is to quantitatively assess the accuracy of the simulation system capabilities for replicating known real-world behavior samples. Within the fidelity referent and simulation system knowledge specification there exist a set of three types of behavior samples: internal, external and complex (*Expression 5.15*). These internal and external behavioral samples can be evaluated with each other by means of a fidelity evaluator function with the following generic structure:

$$
\begin{aligned}
c_{\Delta_{ac\_4}}(R_w^{ref}, S_{appx})_{d_1} &= \left\{ \left(q_{ref}\left(t_1\right) - q_{exec}\left(t_1\right) \right)_{q_k} \quad \cdots \quad \left(q_{ref}\left(t_1\right) - q_{exec}\left(t_1\right) \right)_{q_k} \right\} \\
c_{\Delta_{ac\_4}}(R_w^{ref}, S_{appx})_{d_2} &= \left\{ f_{diff}\left(\omega_{ref}, \omega_{exec} \right)_{u_1} \quad \cdots \quad f_{diff}\left(\omega_{ref}, \omega_{exec} \right)_{u_n} \right\}
\end{aligned}
\qquad (7.23)
$$

Here the first evaluator function output is a vector that specifies the difference between each corresponding elements of both behavior sample initial conditions $q(t_1)$. The second output is vector with *trajectory difference function* f_{diff}, which quantifies the difference in trajectory for each corresponding input vector element $\omega\left(t\right)_{u_i}$. Such a function can be of any form but maps the difference between two trajectories into a single real value [156]. Depending on the type of behavior sample, the third output either contains the quantified difference for each real-world system output $\rho\left(t\right)_{y_i}$ or the state $q\left(t\right)_{q_i}$ vector element. This is formally defined as follows:

$$
\begin{aligned}
c_{\Delta_{ac\_4}}(R_w^{ref}, S_{appx})_{d_3} &= \left\{ f_{diff}\left(\rho_{ref}, \rho_{exec} \right)_{y_1} \quad \cdots \quad f_{diff}\left(\rho_{ref}, \rho_{exec} \right)_{y_m} \right\} \\
&\mathit{or} \\
c_{\Delta_{ac\_4}}(R_w^{ref}, S_{appx})_{d_3} &= \left\{ f_{diff}\left(q_{ref}, q_{exec} \right)_{q_k} \quad \cdots \quad f_{diff}\left(q_{ref}, q_{exec} \right)_{q_k} \right\}
\end{aligned}
\qquad (7.24)
$$

For gaining proper correspondence judgments for the real-world system output and state responses it is important that the quantified difference (*Expression 7.23*) between both the initial conditions and input segments are zero or at least as small as possible. More than for structural resolution and non-causal behavior accuracy this requires a careful design of the test experiments within experimental frames. Even with such experimental design controllability, observability and resource constraints could make it difficult or impossible to obtain an appropriate and complete set of behavioral test data as desired. In case limited real-world behavioral data is available for certain behavioral sample accuracy assessments, this will introduce additional uncertainty and errors. The evaluation of the impact of these errors and uncertainties varies from case to case and requires tailored fidelity evaluator functions for each specific case. This is a complex and special issue, which is beyond the scope of this thesis. Without affecting the applicability of the evaluator function in the remaining of this section it is assumed that appropriate and complete set of behavioral data is available. For some general initial approaches to assess this issue the reader is referred to [67].

*Expression 7.2*3 has to be applied to all initial conditions and input trajectories that are part of a complex behavior sample (Section 5.4.2). Furthermore, for complex behavior samples *expression 7.24* must be applied to either the complex behavior state $Q_{complex}$ or output $Y_{complex}$ vector (*Expression 5.13* and *5.14*), instead to each separate real-world system state or output trajectory.

As already discovered in the early stages of this simulation fidelity project, there are more than one possible ways to characterize the difference of behavior sample trajectories [122] [125]. In practice it is therefore sometimes necessary to apply several fidelity evaluator functions with different f_{diff} to the same behavior samples. When necessary or for convenience the results of these fidelity evaluator functions can also be aggregated into an overall accuracy quantification for a trajectory [154]. The next paragraphs will discuss possible and commonly used instance of these *trajectory difference function* f_{diff}.

Absolute Maximum and Minimum Accuracy

The most straightforward trajectory difference functions are the absolute maximum and minimum error, which return the absolute maximum and minimum error between a arbitrary reference (z_{ref}) and simulated (z_{exec}) trajectory variable within the trajectory domain $[t_1,t_2]$ (*Figure 7-4*). These both functions are formally defined as:

$$f_{diff}^{max}\left(z_{ref}, z_{exec}\right) = \max_{t\in[t_1,t_2]}\left|z_{ref}\left(t\right) - z_{exec}\left(t\right)\right|$$

$$f_{diff}^{min}\left(z_{ref}, z_{exec}\right) = \min_{t\in[t_1,t_2]}\left|z_{ref}\left(t\right) - z_{exec}\left(t\right)\right| \quad (7.25)$$

Expression 7.25 defines an error bandwidth in which the absolute behavioral error of the simulated trajectory resides in. The proportional inverse scaled equivalent of *expression 7.25* i.e. the maximum and minimum accuracy, is calculated with the use of the absolute maximum value of the reference trajectory:

$$\left(d_1\right)_{\max}^{-1} = 1 - \frac{f_{diff}^{max}\left(z_{ref}, z_{exec}\right)}{f_{diff}^{max}\left(z_{ref}, z_{exec}\right) + \frac{1}{k}} \qquad \left(d_1\right)_{\min}^{-1} = 1 - \frac{f_{diff}^{min}\left(z_{ref}, z_{exec}\right)}{f_{diff}^{min}\left(z_{ref}, z_{exec}\right) + \frac{1}{k}} \quad (7.26)$$

In *expression 7.25* scaling factor k is defined as $k = \max_{t\in[t_1,t_2]}\left|z_{ref}\left(t\right)\right|$. The lower the values of *expression 7.25* the more accurately the simulated trajectory replicates the corresponding reference trajectory in the fidelity referent.

Integrated Absolute and Squared Accuracy

Other often-used trajectory difference functions are the integrated absolute and squared error metrics. The integrated absolute error is calculate by means of integrating the absolute difference between a reference (z_{ref}) and simulated (z_{exec}) trajectory variable over its domain [t_1,t_2]. This function results in the area encapsulated between both the trajectories (*Figure 7-4*). The square error is the time-weighted integral of the squared difference between z_{ref} and z_{exec}. These both functions are formally defined as:

$$f_{diff}^{abs}\left(z_{ref},z_{exec}\right)=\int_{t_1}^{t_2}\left|z_{ref}\left(t\right)-z_{exec}\left(t\right)\right|\cdot dt$$

$$f_{diff}^{square}\left(z_{ref},z_{exec}\right)=\int_{t_1}^{t_2}\left(z_{ref}\left(t\right)-z_{exec}\left(t\right)\right)^2 dt \tag{7.27}$$

It is also possible to create time-weighted versions of the above expression [91].

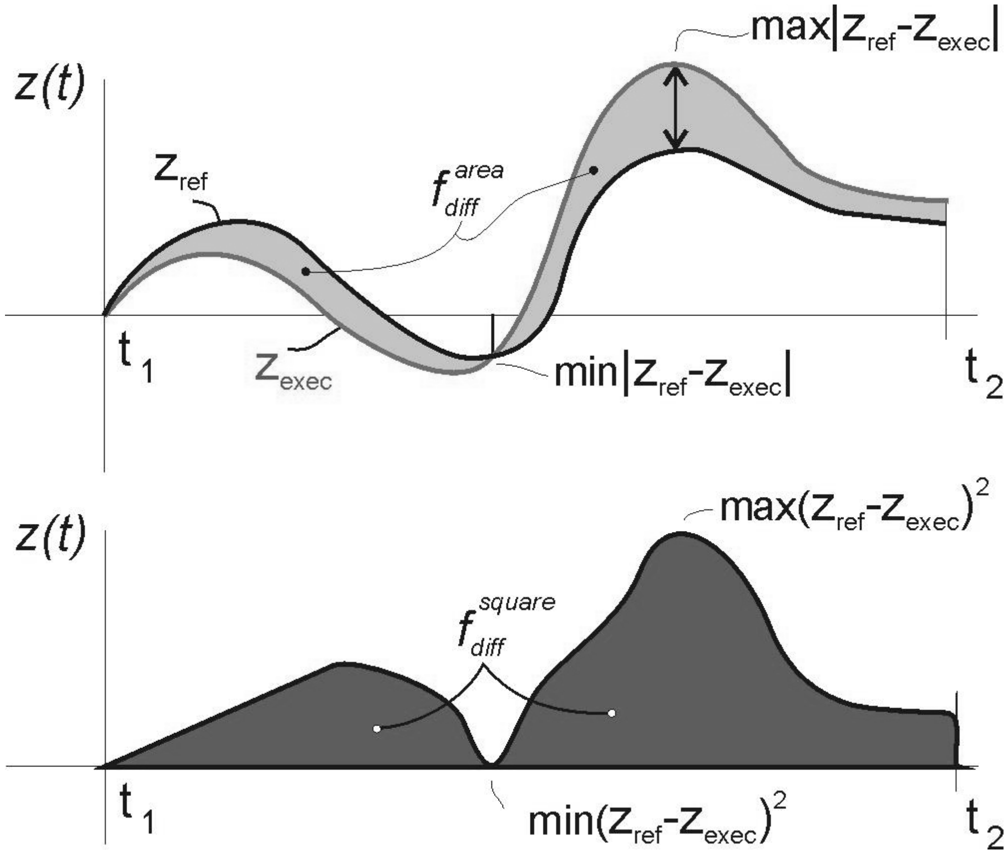


Figure 7-4 Absolute Maximum, Minimum, Integrated and Squared Error

The proportional inverse scaled equivalent of *expression 7.26* is calculated with the use of the time integrated absolute and squared value of the reference trajectory according *expression 7.28*. A value of one for *expression 7.28* indicates a perfect match of the simulated trajectory with the reference trajectory. The smaller the values of *expression*

7.28 the less accurately the simulated trajectory replicates the corresponding reference trajectory in the fidelity referent.

$$\left(d_1\right)_{abs}^{-1} = 1 - \frac{f_{diff}^{abs}\left(z_{ref}, z_{exec}\right)}{f_{diff}^{abs}\left(z_{ref}, z_{exec}\right) + \dfrac{1}{\int_{t_1}^{t_2} \left|z_{ref}\left(t\right)\right| \cdot dt}}$$

$$\left(d_1\right)_{square}^{-1} = 1 - \frac{f_{diff}^{square}\left(z_{ref}, z_{exec}\right)}{f_{diff}^{square}\left(z_{ref}, z_{exec}\right) + \dfrac{1}{\int_{t_1}^{t_2} z_{ref}\left(t\right)^2 \cdot dt}} \tag{7.28}$$

Asymptotic, Steady-State, Derivative and Other Set-Point Accuracy

A common practice to analyze real-world system behaviors, particular in area of control and dynamic system engineering, is processing behavior instances into a set of specific characteristic quantities that are used to evaluate and compare system behaviors [80] [99] [100] [156]. This practice is an alternative and/or complimentary method for directly assessing the difference between a referent and simulated trajectory over-time by means of the previously discussed trajectory difference functions. The general trajectory difference function for this kind of applications looks as follows [155]:

$$f_{diff}\left(z_{ref}, z_{exec}\right) \underset{def}{=} d_z\left(f_{diff}^{select}\left(z_{ref}\right), f_{diff}^{select}\left(z_{exec}\right)\right) \tag{7.29}$$

Here $f_{diff}^{select}\left(z\right)$ is a function that selects or generates the characteristics quantities of interest from a trajectory *z*. The difference between these quantities of the referent and simulated trajectory is determined by the metric d_z. Usually, such practices utilize special input trajectories to drive both the real-world system and the simulation system in order to compare certain characteristics quantities of the state and output vector trajectories. Examples of these input trajectories include pulse, step, ramp, sine and square wave, event sequence and stochastic spectra. The next $f_{diff}^{select}\left(z\right)$ functions are commonly used and applied (*Figure 7-5*):

- *Asymptotic Value:* This function returns the asymptotic value for a trajectory *z(t)*. Practically, this means the value at a large time stamp t_2:

$$f_{diff}^{select}\left(z\right) = \lim_{t \to t_2} z\left(t\right) \quad \text{with } t_1 \ll t_2 < \infty \tag{7.30}$$

- *Steady State Value:* This function returns the steady-state value of a trajectory *z(t)* in case it is steered from an initial steady-state to the next:

$$f_{diff}^{select}\left(z\right) = z\left(t_s\right) \text{ with } t_s : \dot{z}\left(t\right) = 0 \ \ \forall t \in \left[t_s, t_2\right] \wedge t_s > t_1 \tag{7.31}$$

- *Steady State Settling Time:* This function is related to the previous one and returns the transient time from an initial steady state at t_1 to the next:

$$f_{diff}^{select}(z) = t_s\text{-}t_1 \;\text{ with }\; t_s : \dot{z}(t) = 0 \;\; \forall t \in [t_s, t_2] \wedge t_s > t_1 \qquad (7.32)$$

- *Derivative Value:* This function determines for a specific instance in time (t_k) the time derivative of the trajectory *z(t)*:

$$f_{diff}^{select}(z) = \dot{z}(t_k) \qquad (7.33)$$

- *Maximum Overshoot:* This function determines the maximum overshoot for step input responses with respect to the final steady-state value at time stamp t_s (*Expression 7.31*):

$$f_{diff}^{select}(z) = \max(z(t)) - z(t_s) \qquad (7.34)$$

- *Rise-time:* This function determines the minimal time for a state or output responses to a step or ramp input trajectory to reach a certain value *k*:

$$f_{diff}^{select}(z) = t_{min} \;\text{ with }\; t_{min} : z(t_{min}) = k \wedge z(t) \neq k \;\; \forall t < t_{min} \qquad (7.35)$$

- *Time to Double or Time to Half:* This function determines the time needed for a continues increasing trajectory *z(t)* to double or half its amplitude. Particular half-life is important to characterize the damping of periodic oscillating trajectories.

$$\begin{aligned} f_{diff}^{select}(z) &= t_{double} - t_1 \;\text{ with }\; t_{double} : z(t_{double}) = 2 \cdot z(t_1) \wedge t_1 < t_{double} \\ f_{diff}^{select}(z) &= t_{half} - t_1 \;\text{ with }\; t_{half} : z(t_{half}) = 2 \cdot z(t_1) \wedge t_1 < t_{half} \end{aligned} \qquad (7.36)$$

Since all the above $f_{diff}^{select}(z)$ functions return a single real value, the trajectory difference function (*Expression 7.29*) and associated proportional inverse scaled equivalent are similar and look as follows:

$$\begin{aligned} f_{diff}(z_{ref}, z_{exec}) &= f_{diff}^{select}(z_{ref}) - f_{diff}^{select}(z_{exec}) \\ (d_1)^{-1} &= 1 - \frac{\left| f_{diff}(z_{ref}, z_{exec}) \right|}{\left| f_{diff}(z_{ref}, z_{exec}) \right| + \dfrac{1}{f_{diff}^{select}(z_{ref})}} \end{aligned} \qquad (7.37)$$

A value of one in *expression 7.37* indicates a perfect match of the characteristic quantity of the simulated trajectory with the reference trajectory. The smaller the values of e*xpression 7.37* the less accurately the simulated trajectory replicates the characteristic quantity of the corresponding reference trajectory.

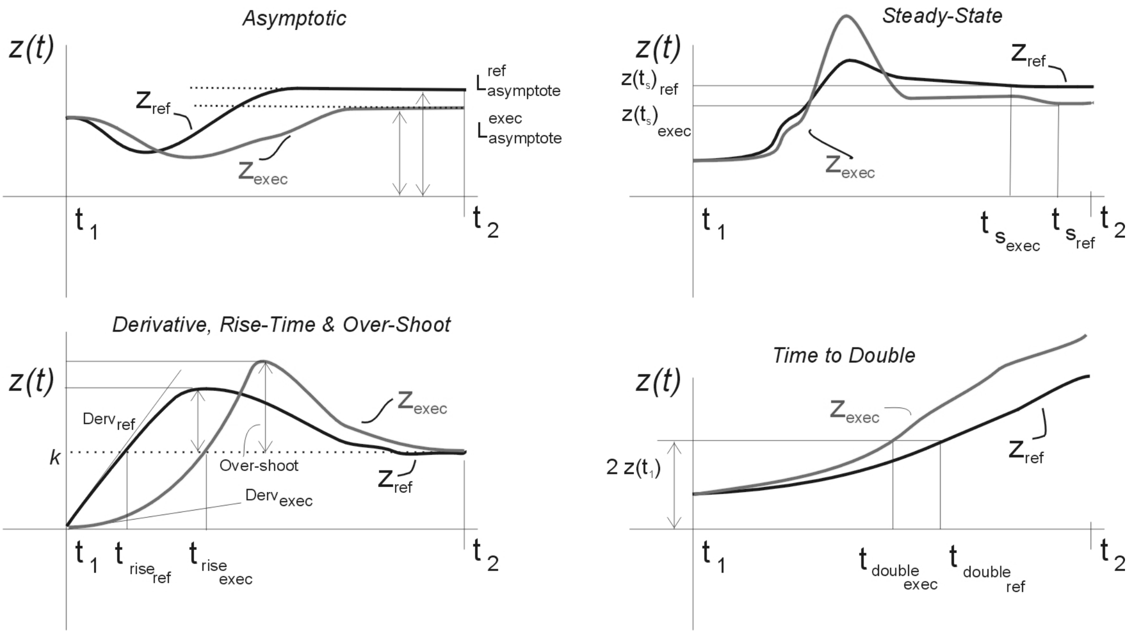


Figure 7-5 Definitions of Trajectory Characteristic Quantities

Temporal Accuracy

Real-world behavior samples are always specified with respect to a time-base. Therefore, assessing the temporal or time representational accuracy of the replicated real-world system is an important issue. Within the real-world system realization description section of the simulation system knowledge (Section 6.3.2) already two temporal metrics were developed: *time speed-up factor* and the *update-time frame* (*Expression 6.30* and *6.31*). Both metrics directly provide a quantification of the difference in simulated time versus the real-world physical time, which progresses continuously. Their associated proportional inverse scaled equivalent to express their accuracy are defined as:

$$\left(d_1\right)^{-1}_{speed\_up} = \begin{Bmatrix} \dfrac{1}{T_{speed\_up}} & \forall T_{speed\_up} \geq 1 \\ T_{speed\_up} & \forall T_{speed\_up} < 1 \end{Bmatrix}$$

$$\left(d_1\right)^{-1}_{frame} = 1 - \frac{1}{k \cdot T_{frame} + 1} \qquad (7.38)$$

Where k is an arbitrary scaling factor, which is usually equal to one. In case $\left(d_1\right)^{-1}_{speed\_up}$ is equal to one the real-world system is replicated in real-time. The smaller the value is the less accurate the simulated time is with respect to the physical time. Likewise, a $\left(d_1\right)^{-1}_{frame}$ less than one indicates a less accurate or more discrete replication of the physical time by the simulation system.

A response time delay with respect to the real world is another form of temporal errors that occur in simulation systems [51] [122] [125]. These errors occur when a real-world system model reproduces the same behavioral state or output response that the real-world system would under the same identical conditions and with the same input trajectory, but with a phase error along the time-base. Such errors result from both behavioral approximation and hard/software latencies. Phase errors can be calculated using *Expression 7.29* with a selector function $f_{diff}^{select}(z)$ that returns the time stamp of a characteristic predetermined point in the output or state trajectory *z(t)*. The trajectory phase difference function and associated proportional inverse scaled equivalent then becomes:

$$f_{diff}^{phase}\left(z_{ref}, z_{exec}\right) = f_{diff}^{select}\left(z_{ref}\right) - f_{diff}^{select}\left(z_{exec}\right) = t_{ref}^{phase} - t_{exec}^{phase}$$

$$\left(d_1\right)^{-1} = 1 - \frac{f_{diff}^{phase}\left(z_{ref}, z_{exec}\right)}{t_{ref}} \tag{7.39}$$

In real-life all real-world system behavior evolves with respect to the same physical time-base and events between systems occur in a specific time-stamped order on this global time-base. Depending on the chosen simulation model architecture (mathematical and logical relationships, hard/software, etc.) it is not uncommon that there exist multiple local representations of the same physical time-base within different real-world system models. This could cause serious behavioral sample errors if local time-bases of interacting real-world system models advance in an uncorrelated manner. An issue of specific concern in parallel and distributed simulation systems that is further complicated within these kinds of simulation systems by possible different latencies between the various interacting real-world system models. One of the erroneous results could be that the causal order of events and other functional relationships [51], which are usually implemented as time-stamped data streams between interacting model components [128], do not correspond to those causal relationships persistent in the real-world behavior samples. A simple trajectory difference function for this purpose is the one that returns the set of such observed temporal related errors:

$$f_{diff}^{temporal}\left(z_{ref}, z_{exec}\right) = \left\{e_1^{temporal}, e_2^{temporal}, \ldots\right\} \tag{7.40}$$

Using this expression the causal accuracy of the behavior sample registered during simulation execution is given by the following proportional inverse scaled equivalent:

$$\left(d_1\right)_{abs}^{-1} = 1 - \frac{n\left(\left\{e_1^{temporal}, e_2^{temporal}, \ldots\right\}\right)}{n\left(\omega_{\langle t_1, t_2\rangle}\right)} \tag{7.41}$$

In here $n\left(\omega_{\langle t_1, t_2\rangle}\right)$ is the number of discrete time-stamps in the simulated trajectory under the assumption that the number of temporal related errors is less than the number of data points in the trajectory. *Expression 7.41* states that the less temporal related errors the more accurate the referent trajectory is reflected during simulation executions.

There exist a whole range of other temporal related issues from both the modeling as well as the simulation system hard/software perspective, which effect error and accuracy of the simulated behavior samples with respect to the real-world. Most of these temporal related issues and metrics are extensively discussed in modeling and simulation literature and communities such as Simulation Interoperability Standards Organization, Society of Computer Simulation and AIAA Modeling and Simulation [51] [57] [58] [111] [118] [125] [139] [152]. The reader is referred to these and other publications for more detail discussion. One last worth temporal issue worth to mention here is the effect of large differences in update rates for various interaction real-world system models. Here dead-reckoning algorithms are utilized to extrapolate the input vector for time frames when there is no new but required input data available from another real-world system model. Obviously this could cause errors, often discontinuities along the variable axis, in the behavior samples of a real-world system model applying dead reckoning.

Frequency Domain Accuracy

Two typical categories of trajectory difference functions are those that derive from frequency domain analysis of dynamic systems [35] [91] [100] [125] [151] [152] [156]. The first category is based on a $f_{diff}^{select}(z)$, which generates the frequency contents of a deterministic or stochastic trajectory z in terms of the distribution of energy over different frequencies. This resulting characteristics quantity is commonly known as the power spectral density function. Much extensive literature and tools are available for calculating or estimating the power spectral density from various types of trajectories. Therefore, the reader is referred to the above literature references for more details on these methods. For now it is just sufficient to know that a power spectral density function of trajectory z is always calculated and plotted in a diagram with logarithmic axis for the frequency domain of interest, as depicted in the next example figure taken from [91].

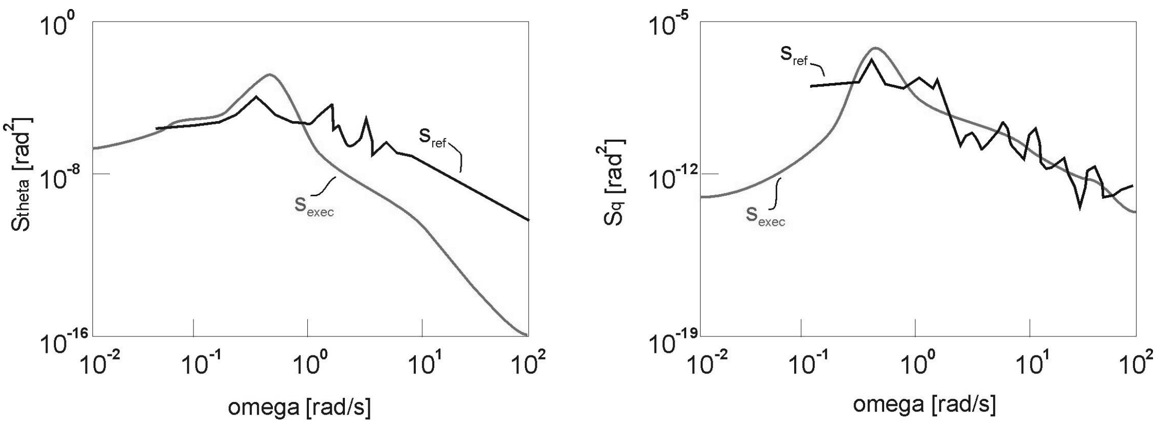


Figure 7-6 Power Spectral Densities of Aircraft Variables Due to Turbulence

Once the real world and the simulated behavioral samples are transformed into power spectral density functions they can be compared for their differences in the frequency range of interest. To assess the accuracy of the power spectral density function resulting from the simulation system, difference quantifier functions similar to those as given by *Expression 7.25* to *7.28* can be used.

The second category of frequency domain analysis based $f_{diff}^{select}(z)$, uses complex behavior samples in the form of real-world system state and output responses to a spectrum of sine shaped input trajectories of varying frequency. These behavior samples are then transformed in to a complex behavior quantifier commonly known as frequency response or Bode plots [99] [100]. A Bode plot comprises two diagrams as function of the input frequency. One diagram specifies the absolute value of the resulting state or output response modulus and the other specifies the phase lead/lag with respect to the input trajectory.

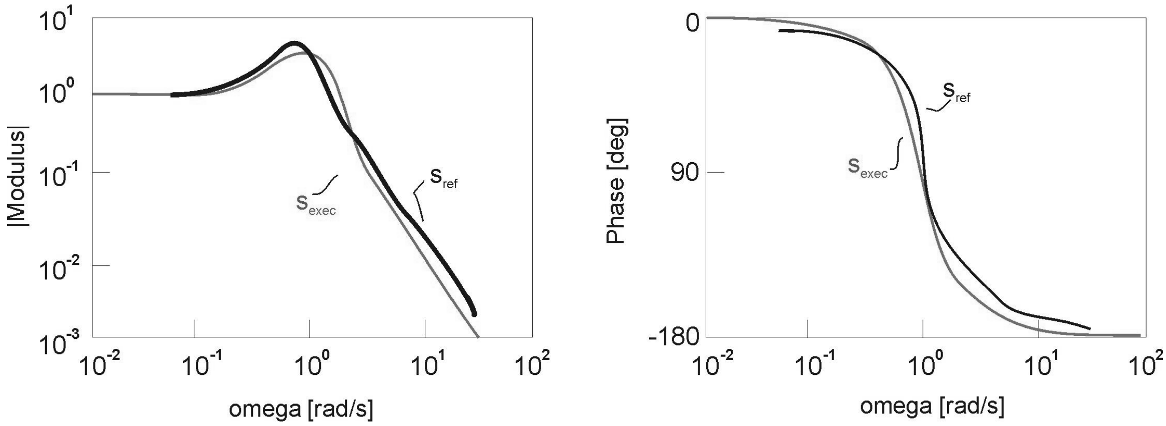


Figure 7-7 Bode Plot for a Second Order Dynamic System Simulation

Figure 7-7 gives a typical example of a Bode plot for a simulation of a second-order dynamic system [99]. To assess the accuracy of the Bode plots resulting from the simulation system, difference quantifier functions similar to those as given by *Expression 7.25* to *7.28* can be used. Some often-used $f_{diff}^{select}(z)$ functions for accuracy assessment that select characteristics quantities from these plots include: asymptotic values (*Expression 7.30*), derivatives (*Expression 7.32*), peak-value, and band-with. These last two characteristic functions are defined for the modulus plot as follows using the steady-state gain *K*:

$$f_{diff}^{select}\left(bode_{\mathrm{mod}}\left(\omega_{freq}\right)\right) = \max\left(bode_{\mathrm{mod}}\left(\omega_{freq}\right)\right)$$

$$f_{diff}^{select}\left(bode_{\mathrm{mod}}\right) = \omega_{bandwidth} \quad \text{with } \omega_{bandwidth} : bode_{\mathrm{mod}}\left(\omega_{bandwidth}\right) = \tfrac{1}{2}\sqrt{2}\cdot K \qquad (7.42)$$

Statistical Accuracy of Stochastic and Complex Behavior Samples

Another important and very often applied class of fidelity evaluator functions for complex behavior samples is the use of statistics. As already discussed in section 2.4.6 there exists a broad range of literature on application of statistical techniques to model and simulation systems to assess the behavioral accuracy of stochastic or random systems as well as on standard literature on general statistical methods [79] [89] [90] [137]. It is far beyond the scope of this thesis to list and discuss all these statistical techniques in here. Instead several classical and mainstream statistical methods are highlighted here and it is briefly demonstrated how these can be integrated as fidelity evaluator functions within the unified fidelity framework. Basically there exist two approaches for assessing statistical accuracy of stochastic and complex behavior samples.

The first approach is based on a characteristic quantity that can be derived for each behavior sample in the complex behavior sample. Usually, these quantities are certain measures of performance or effectiveness of a dynamic system [4] [5], such as for instance the number of delayed flights or average delay in an air-traffic control simulation during peak-hour [32] [118]. Both the reference and simulated complex output or state vector in those cases consist of a sample of an actual population. These samples can then be compared for differences and accuracy in statistical measures of central tendency and dispersion, like mean, median, standard deviation, variance, etc. As an illustration, consider a function $f_{diff}^{select}\left(z_{complex}\right)$ that generates the characteristic statistical quantities of mean μ_z and variance σ_z^2 from a complex behavior quantifier set $z_{complex}$. The complex behavior sample difference functions then become:

$$f_{diff}^{statistic}\left(z_{ref},z_{exec}\right)_{d_1} = f_{diff}^{select}\left(z_{complex}^{ref}\right) - f_{diff}^{select}\left(z_{complex}^{exec}\right) = \mu_z^{ref} - \mu_z^{exec}$$

$$f_{diff}^{statistic}\left(z_{ref},z_{exec}\right)_{d_2} = \left(\sigma_z^2\right)_{ref} - \left(\sigma_z^2\right)_{exec} \tag{7.43}$$

The associated proportional inverse scaled equivalent to assess the accuracy of the simulated mean μ_z and variance σ_z^2 is given by *expression 7.37*. Other more thorough statistical techniques that have and could also be applied for developing $f_{diff}^{select}\left(z_{complex}\right)$ functions include, but are not limited to, hypothesis and confidence intervals, t-test, analysis of variance, bootstrap, regression and nonparametric techniques like Mann-Withney and Kolmogorov-Smirnov [4] [22] [67] [129].

The second approach is the time-series analysis approach, which directly calculates statistical properties such as time mean and variance from a stochastic behavior sample. These two time mean and variance are defined and generated by the following $f_{diff}^{select}\left(z\right)$ functions for a stochastic behavior sample *z(t)*:

$$f_{diff}^{select}\left(z\right) = \mu_z = \lim_{T\to\infty}\frac{1}{2\cdot T}\int_{-T}^{T} z\left(t\right)\cdot dt$$

$$f_{diff}^{select}\left(z\right) = \sigma_z^2 = \lim_{T\to\infty}\frac{1}{2\cdot T}\int_{-T}^{T} z^2\left(t\right)\cdot dt - \mu_z^2 \tag{7.44}$$

In [91] it is demonstrate how these integrals can be approximated from discrete and finite stochastic behavior samples. In here the usage of power spectral density function estimates for a stochastic behavior sample plays an important roll. This not only allows for the practical assessment of the difference in the mean and variance of both a reference and simulated stochastic behavior sample, but also comparison based on the power spectral densities themselves. Other characteristic and important statistical properties for stochastic behavior samples include auto-product, auto covariance, cross product and cross variance [90] [91]. To achieve a better estimate of these kind of statistical properties usually an ensemble of stochastic behavior samples i.e. complex behavior sample are elicited. For each behavior sample the above statistical properties can be determined, whose results can than be combined into another statistical quantity, such as an expected value of the mean or variance over the whole ensemble, to estimate the whole stochastic system or process properties. Such an approach and also any other statistical method require a careful experimental design for both the reference and the simulation system in order to obtain well-conditioned stochastic data and reduce statistical uncertainties [89].

7.3.3 Real-World System Qualitative Behavior Accuracy

Qualitative behavior metrics determine how accurately the simulation system behavior samples meet the qualitative behavior knowledge as specified in the fidelity referent (Section 5.4.2). The in this referent section described ordinary and change-in-value behavioral relationships describe observation frames for which a certain qualitative real-world system behavior holds. These frames provide the basis for developing simulation executions in order to elicit the corresponding simulated behavior samples, which can then be evaluated against the specified qualitative reference behavior. The outcomes of the in here used metrics provide a coarse-grained measure, in qualitative sense, for the behavioral difference between the real world and its simulated counterpart.

Both the ordinary and the change-in-value behavioral relationships are described by means of a set of Boolean cause effect expressions (*Expressions 5.16* until *5.19*). Like for several previously discussed non-causal behavioral relationships, proposition functions are also utilized as the bedrock for fidelity evaluator functions in this area. The fidelity evaluator function for an ordinary causal relationship is defined as:

$$c_{\Delta_{ac\_5}}(R_w^{ref}, S_{appx})_{d_1} = \left\{ \begin{array}{l} p_{RL_{ordinary}}\left(t_{exec}\right)_k : RL_k^{oc}\left(...\right)_i^{exec} \rightarrow RL_k^{oe}\left(...\right)_i^{exec} \\ is\ true \Rightarrow d_1 = 0 \\ \\ p_{RL_{ordinary}}\left(t_{exec}\right)_k : RL_k^{oc}\left(...\right)_i^{exec} \rightarrow RL_k^{oe}\left(...\right)_i^{exec} \\ is\ false \Rightarrow d_1 = 1 \end{array} \right\} \quad (7.45)$$

Inhere is $p_{RL_{ordinary}}\left(t_{exec}\right)_k$ the proposition function on the k^{th} system its i^{th} ordinary causal relationship $RL_k^{oc}\left(...\right)_i \rightarrow RL_k^{oe}\left(...\right)_i$ in the fidelity referent whose truth is tested for the corresponding real-world system model variables during simulation execution. Similarly, the fidelity evaluator function for change in-value causal relationships $RL_k^{cvc}\left(...\right)_i \rightarrow RL_k^{cve}\left(...\right)_i$ is defined by *expression 7.46.*

$$c_{\Delta_{ac\_6}}(R_w^{ref}, S_{appx})_{d_1} = \begin{cases} p_{RL_{chiv}}\left(t_{exec}\right)_k : RL_k^{cvc}\left(...\right)_i^{exec} \rightarrow RL_k^{cve}\left(...\right)_i^{exec} \\ is\ true \Rightarrow d_1 = 0 \\ \\ p_{RL_{chiv}}\left(t_{exec}\right)_k : RL_k^{cvc}\left(...\right)_i^{exec} \rightarrow RL_k^{cve}\left(...\right)_i^{exec} \\ is\ false \Rightarrow d_1 = 1 \end{cases} \tag{7.46}$$

The proportional inverse scaled equivalents of both *expression 7.45* and *7.46* are identical and are simply given by the next expression:

$$\left(d_1\right)^{-1} = 1 - d_1 \tag{7.47}$$

Here a value of one means that the executable simulation model accurately represents either the ordinary or the change-in-value causal behavioral. A value of zero indicates a certain degree of mismatch in the simulated ordinary or the change-in-value causal behavioral relationships with respect to the referent.

7.3.4 Real-World System Interaction Causality Accuracy

The last category of real-world system behavioral accuracy metrics are those evaluating how accurately the simulation system represents the specified reference interaction causality mechanisms. As discussed in section 5.4.2 the total interaction causality set (*Expression 5.6*) consists of interaction chain descriptions (*Expression 5.7*), which specify a chain of causal order interactions or event between multiple real-world systems. Due to the complexity of an interaction chain description its fidelity evaluator function is not so straightforward and is composed of several logical steps. The first step comprises assessing whether for a given pre-condition $C_{pre_i}^{ref}$ the simulation model can reproduce the belonging i^{th} reference interaction chain $IC_{chain_i}^{ref} \in IC_{ref}^{rw}$ when triggered with the same trigger event $E_{trig_i}^{ref}$. In case this interaction chain isn't present in the simulation model or observed during simulation execution, the fidelity evaluator function will return one to indicate the maximum difference with the real-world. Formally this looks as follows:

$$c_{\Delta_{ac\_7}}(R_w^{ref}, S_{appx})_{d_1} = 1 \ \ if \ \ IC_{chain_i}^{exec} \notin IC_{ref}^{rw} \ \ for \ \ C_{pre_i}^{exec} = C_{pre_i}^{ref} \ \wedge E_{trig_i}^{exec} = E_{trig_i}^{ref} \tag{7.48}$$

If the i^{th} reference interaction chain $IC_{chain_i}^{ref}$ is present in the simulation model and for the given pre-condition $C_{pre_i}^{ref}$ and same triggering event $E_{trig_i}^{ref}$ the same reference end condition $C_{pre_i}^{ref}$ is achieved plus any variations and exceptions are also correctly reproduced, the fidelity evaluator function will return zero. In that case the simulated interaction chain is identical to the reference one:

$$c_{\Delta_{ac\_7}}(R_w^{ref}, S_{appx})_{d_1} = 0 \ \ if \ \ IC_{chain_i}^{exec} \equiv IC_{chain_i}^{ref} \tag{7.49}$$

In all other cases the simulated interaction chain is present but contains several errors in either the primary interaction chain, its variations and/or exceptions. To express the degree of difference with the reference interaction chain the next compound expression is used within the fidelity evaluator function:

$$c_{\Delta_{ac\_7}}(R_w^{ref}, S_{appx})_{d_1} = d_i^{IC} = \tfrac{1}{3}\left(2 \cdot \frac{n\left(I_{prim}^{ref}\right) - n\left(I_{prim_{corr}}^{exec}\right)}{n\left(I_{prim}^{ref}\right)} + \frac{n\left(B_{seq}^{ref}\right) - n\left(B_{seq_{corr}}^{exec}\right)}{n\left(B_{seq}^{ref}\right)}\right) \quad (7.50)$$

Here I_{prim}^{ref} is the reference primary interaction chain of $IC_{chain_i}^{ref}$ and $I_{prim_{corr}}^{exec}$ the set of corresponding and correctly represented interactions during simulation execution. Similar B_{seq}^{ref} is the possible set of all branching variations and exceptions and $B_{seq_{corr}}^{exec}$ its correctly simulated counterpart. As can be seen in *expression 7.50* the impact of the primary interaction chain error on the overall measured difference d_i^{IC} is larger than for the secondary exceptions and variations.

The proportional inverse scaled equivalents of both *expression 7.48* and *7.50* are identical and are simply given by the next expression:

$$\left(d_1\right)_i^{-1} = 1 - d_i^{IC} \quad (7.51)$$

In expression 7.51 a value of one indicates that the i^{th} $IC_{chain_i}^{ref}$ is correctly represented by the simulation system. A lower value indicates a lesser degree of correspondence the to this reference interaction chain description.

Using the previous result, the total interaction causality accuracy of the simulation system can now be defined as the proportional inverse scaled equivalent sum of the separate measured interaction causality chain description differences:

$$\left(d_1\right)_{total}^{-1} = 1 - \frac{\sum_{i=1}^{n(IC_{ref}^{rw})} d_i^{IC}}{n(IC_{ref}^{rw})} \quad (7.52)$$

This expression returns a value of one in case all interaction chain descriptions are fully and correctly represented by the simulation system. A value smaller than one is returned if interaction chain descriptions are incomplete replicated or missing. The smaller this value the less accurate the replication of real world by the simulation system.

7.4 Subject Matter Expert Based Metrics and Methods

As argued in section 4.3.2 and formalized by fidelity *theorem 5* practical fidelity assessment has an inherent subjective and qualitative element. This yields that besides the more objective and quantitative fidelity evaluation metrics and methods discussed in the previous two sections also qualitative fidelity evaluator functions of more subjective nature can or have to be applied in order to fully assess the level of fidelity. The common approach in qualitative fidelity evaluation is the deployment of one or subject matter experts (SME) who will rate the level of structural and behavioral simulation fidelity in one or several different real-world areas of interest. Despite that SME opinions are frequently used in simulation fidelity evaluations there exist very little information and formal or standard procedures to properly conduct such SME evaluations within simulation system validation activities, even not for those case where SME opinions are mandatory [35]. Since many consider SME based evaluations as a special and uncultivated area requiring considerable more research, this thesis will here only touch upon the basic application principles of SME based fidelity evaluation within the unified fidelity framework [22] [86] [109].

7.4.1 SME Roles in Human Realizations of Fidelity Evaluator Functions

Subject matter experts can play many roles within the model and simulation enterprise, which not all do relate to fidelity assessment [22]. The first SME role in fidelity assessment is serving as a direct knowledge source in de development and validation of the formal real-world reference knowledge-base as discussed in chapter 5. Usage of SME for this purpose is also proposed by Metz when no other real-world reference data is readily available [86]. Furthermore SME play also a similar role in the validation of the simulation system knowledge-base developed in chapter 6. The second SME role is that of specifying and verifying the required level of simulation fidelity for a specific application purpose (See Section 8.2)

The third role is that of fidelity evaluator within the conceptual model and simulation testing stages of simulation system development (Chapter 90). In formal sense the SME expert then serves as a human realization of one or more fidelity evaluator functions given by *Expression 4.5*. Like for the fidelity evaluator functions discussed in the previous sections the objective of SME evaluation is to compare a portion of the real-world reference knowledge with the simulation system knowledge and generating a qualitative and/or quantitative rating of their differences.

In the strictest sense SME based fidelity evaluation means that a SME is provided with a subset of formally specified real-world reference and simulation system knowledge and is asked to rate the magnitude of any structural or behavioral differences. A classical method in this area is the Turing-test [22] [145]. This tests presents the SME with two blind knowledge-sets respectively drawn from the fidelity referent and the simulation system knowledge specification and is asked to differentiate between the two. If he can, he is asked to describe the differences in natural language.

A lesser strict SME based fidelity evaluation is when the SME is only provided with a subset of formally specified simulation system knowledge. In this case a SME must rate any structural or behavioral differences using its own knowledge of the real world as the reference. Obviously, this increases the change of additional errors related to the SME his perception, interpretation and appreciation of the real-world under evaluation [109]. Classically evaluation methods in this area comprise reviews, inspections and walkthroughs of the specified simulation system knowledge to identify and quantify or qualify any differences [22].

The less strict SME based fidelity evaluations are those in which the SME is placed in the situation where he can directly interact with and observe the simulation system during execution. Now the SME has to generate the fidelity evaluation on its own perception, interpretation and appreciation of both the real-word and the simulation system representation of this real-world. This will even more increase change of additional fidelity judgment errors. The classical methods in this type of SME fidelity evaluation, particularly found in the training simulation application domain, are human-in-the-loop simulation test and face validation reviews [1] [22] [35] [44] [58].

7.4.2 Addressing SME Usage Issues in Qualitative Fidelity Evaluation

SME usage is and has always been controversial and topic for debate. It outcomes are by many considered to be highly subjective and lacking sufficient reliability due to human misinterpretations, bias, perception errors, etc. One of the reason for this is that usually SME based assessments are conducted in ad hoc and unsystematic ways [109]. Without going into too much detail, the top-level approach for SME based fidelity evaluations within the unified fidelity framework is presented. This generic method provides a more formal and systematic process to perform SME based fidelity evaluations and comprises the next eight steps:

1. *Need and Objectives for SME Evaluation Analysis*: Since quantitative fidelity evaluator function and methods are preferred over SME-based evaluation, ascertain that there is a true need for SME-based evaluation. When SME-evaluation is indeed necessary determine and specify its objectives in the context of the whole fidelity evaluation process.
2. *Design SME Evaluation Tasks and Criteria*: Once the fidelity evaluation areas are know specify the exact SME tasks in terms what and how he or she must conduct the evaluation. This involves the development of a set of questionnaires with evaluation criteria and associated rating scales. The use of well-designed rating scales (crisp, fuzzy or nominal), which quantitatively rate the degree of difference between certain real-world aspects and their simulated counterparts, provides a more formal and quantitative means for SME-based fidelity evaluation [2].
3. *Develop Briefings for each SME Evaluation Task*: Next develop sound briefings to brief the SME upfront each fidelity evaluation task they have to perform. This will increase the performance and help guide the SME through his evaluation task such that the evaluations are given from the right perspective, are traceable, well reported and comparable with evaluation results of other SME.
4. *Select SME-Evaluation Results Processing Methods*: To reduce bias and filter out any outliers due to SME misinterpretations, perception errors etc. select or develop results processing method that produces rigorous and objective results. Usually, statistical methods are used for this purpose, which includes the assessment of the SME pool size that is required for each evaluation tasks to obtain statistically reliable results [60] [86] [140].
5. *Select and Assign SME to Appropriate Task(s)*: Once the complete SME-base fidelity evaluation plan has been completed, one can start the selection and assignment of SME to the appropriate tasks. Criteria to select appropriate SME are given in references [22] and [109]. In order to properly manage and schedule all SME evaluation tasks the information about the SME is best document according the *Referent Developer and Validation Information* template developed in section 5.3.3. Here also a SME task competence rating must be provide and is used to weight the fidelity evaluation task results among SME in

the same pool. More priority can than be given to the fidelity evaluation ratings of the SME who are considered more knowledgeable than the other when the same fidelity evaluation task ratings are combined and compared among each other [2].

6. *Brief SME and SME Tasks Execution*: Brief the SME according plans developed in step 3. Monitor SME performance during the evaluation tasks and when necessary assist or steer the SME in solving any unforeseen evaluation difficulties.
7. *Debrief SME and Analyze Individual Results*: Debrief the SME to obtain any additional important information and remarks regarding the fidelity of the simulation execution that are not covered in the questionnaires. Furthermore, store and analyze the SME fidelity evaluation results for correctness, clarity and completeness.
8. *Overall Statistical Analysis and Weighted Aggregation of Scores*: Finally perform the in step four select statistical analysis method on the pool of the same fidelity evaluation ratings and with the help of the SME weight develop a reliable overall SME fidelity rating for this particular area of the real-world replication. Address any found inconsistencies or contradictions among SME results. Next carefully specify and store the results with all other fidelity evaluator function results.

7.5 Summary

In this chapter a basic taxonomy of the most common and elementary fidelity evaluator functions has been presented, which can be used within the unified fidelity framework. Although not fully exhaustive this taxonomy clearly demonstrated how fidelity evaluator functions, as specified by the formal definition of pragmatic fidelity (Section 4.4.2), can be implemented in real simulation practice to measure the difference between the real-world system reference knowledge and replicated real-world system knowledge. The taxonomy structure and its constituent fidelity evaluator functions have directly been derived from the unified fidelity framework referent and simulation system knowledge-base architectures developed in chapters 5 and 6. From the top-level the taxonomy structures fidelity evaluator functions into quantitative methods and qualitative methods as can be seen in *Figure 7-8* at the next page.

Quantitative fidelity evaluator functions are further subdivided in the taxonomy in structural and behavioral oriented methods and metric. Structural fidelity evaluator functions compare the specified structural knowledge section pairs found in the fidelity referent knowledge base and the simulation system. These methods and metrics objectively characterize and quantify the simulation fidelity in terms of resolution or the level of detail used in a simulated representation of the real-world. In this chapter resolution metric and methods were presented that can be applied to assess the real-world system structural hierarchy and the system property-level degree of detail.
Behavioral oriented fidelity evaluator functions focus on the pair-wise comparison between the specified behavioral knowledge found in the corresponding fidelity referent knowledge base and the simulation system specification knowledge base sections. These metrics and methods objectively characterize and quantify the simulation fidelity in terms of accuracy of the behavioral replication of the complete real-world system and/or its subsystems. The behavioral fidelity evaluator functions are further catalogued by the taxonomy in the following four categories: real-world system non-causal behavior,

behavior samples, ordinary causal and logical relationships, and interaction causality evaluator functions.

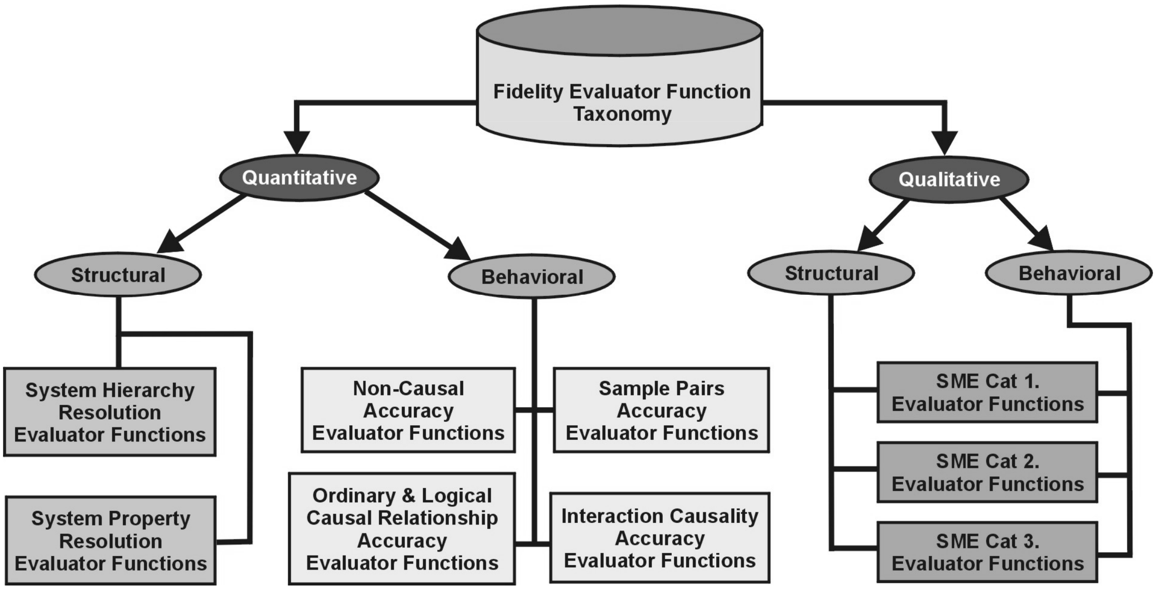


Figure 7-8 Unified Fidelity Framework Evaluator Function Taxonomy

Although quantitative evaluator functions are preferred the use of qualitative subject matter expert based methods are often inevitable. This chapter has briefly touched upon subject matter expert based fidelity metrics and measurement methods for both structural and behavioral knowledge pairs. These qualitative evaluator functions are structured in the taxonomy according the three categories in which both real-world referent and simulation system knowledge is elicited and interrelated by a SME. A generic eight-step process to perform SME based fidelity evaluations has been presented. This process facilitates a more formal and systematic approach for SME based fidelity evaluations within the unified fidelity framework to increase the reliability and repeatability of such fidelity evaluations.

As said in the introduction of this chapter fidelity evaluator functions come in a wide-range of different flavors. The taxonomy of fidelity evaluator functions presented in this chapter must therefore seen as a start point for developing more application and problem specific fidelity evaluator function taxonomies when necessary instead of being considered to be a fully exhaustive or an authoritative taxonomy. However, the here presented unified fidelity framework taxonomy of fidelity evaluator functions should provide enough information and handles as a basis for the development of such specific fidelity evaluator function taxonomies.

What and how many fidelity evaluator functions are required to sufficiently quantify and qualify the level of fidelity depends on the simulation purpose at hand. Therefore in chapter 8 it will be demonstrated how fidelity evaluator functions could be prescribed and documented as part of a fidelity requirements specification. Furthermore, although not explicitly addressed in this chapter for brevity reasons, remember that the fidelity evaluator function results must be carefully specified and stored in a database that directly links to these fidelity requirements. This will not only facilitate the traceability of the fidelity evaluation results but also the efficiency of the whole simulation fidelity assessment process in general (Chapter 9), particularly if such database is incorporated within an integrated and automated fidelity assessment tool-suit. Most of the in this chapter presented evaluator functions have been utilized in the two aerospace simulation case studies (Appendix C).

8 Unified Fidelity Framework: Application Concepts and Techniques

8.1 Introduction

In the previous chapters both the informal definition and formalism for simulation have been developed and presented (Chapter 4). These served in chapters 5 and 6 as the basis for the development for the pragmatic approach to simulation fidelity in terms of a fidelity referent and simulation system knowledge-base. The previous chapter provided a basic set of metrics and measurement methods to practically qualify and quantify the simulation fidelity using the contents of these two knowledge-base systems. This chapter focuses on how all these developed simulation fidelity formalisms, knowledge-bases, metrics and measurement methods should be utilized in the modeling and simulation enterprise (Chapter 3) to serve as a tool for improving the quality of both simulation system and its associated development and validation process. Furthermore, it develops additional concepts, formalisms and other building blocks necessary to realize this objective. In chapter 0 it is demonstrated how these elements fit into a consistent fidelity management process model for the modeling and simulation enterprise.

How much fidelity is required to meet a certain application purpose or object is the prime concern, ever-returning question, and first step in any model and simulation system development process (Chapter 0). Therefore, this chapter starts in section 8.2 with a thorough discussion on this issue. It introduces the unified fidelity framework definition, concepts and formalisms for fidelity requirements specification. Closely related to the required level of fidelity, is the assessment whether the level of fidelity of the resulting simulation does indeed fulfill this requirement. In other words answering the question whether the simulation provides a valid result. Section 8.3 presents how the simulation validation process can be improved by redefining this process in terms of simulation fidelity using fidelity requirements and the other unified fidelity framework concepts and principles as developed in this thesis. To support the simulation system development stages additional metrics and methods are presented in Section 8.4 to assist the developer in the selection, comparison and to make trade-off decisions between different federate, component and real-world system models.

8.2 Fidelity Requirements: Translation of Objectives into the Fidelity Required

Every development of a new system, including simulation system development, starts with the identification of the user needs that arise from a problem, question or deficiency and the desire for a system that addresses this problem, question or deficiency. These identified user needs and objectives form the basis for the establishment of a set of system requirements, which serve as input criteria for the actual design, development and evaluation of the system under development as well as in their selection for reuse. As discussed in Section 3.3.1 simulation systems compared to other systems have a special kind of functional requirements that specify the required real-world representational capabilities and quality necessary to achieve the user objectives. These types of simulation system requirements are referred here as fidelity requirements and are defined as follows:

'The formal specification of the simulation fidelity required in order to properly fulfilling the needs and objectives of the simulation user'

This definition is consistent with the pragmatic definition for fidelity. It is this concept of fidelity requirements that provides a transparent approach to systematically address the classical question of "how much realism is good enough" and to separate between simulation fidelity specification as an absolute measure and relative judgments such as simulation validity (Section 8.3). In the next two subsections the basic elements underlying any fidelity requirement specification are developed together with the mathematical formalisms.

8.2.1 Elements of Fidelity Requirements Specification

Any system requirements engineering process is a structured set of subsequent stages that are traversed iteratively to identify the true user needs and from there on develop a detailed product specification. Praxis shows that requirements engineering is not a front-end analysis but is interlaced with the system design activities [75] [121]. Furthermore, creation of requirements is not a trivial and requires significant experience and knowledge about the subject matter. Which yields that substantial knowledge about the problem domain and some general system architectural considerations (not detail design or implementation) are necessary to fully articulate all system requirements.

Applying the general system requirements engineering process to a simulation system in the context of simulation fidelity requirements then the resulting development framework looks as depicted in *Figure 8-1* on the next page. As discussed in Section 3.3.1, like for any system the development of a simulation system starts with the identification of the user requirements, both functional and non-functional. This yields analyzing and understanding the problem at hand in order to identify the true set of user needs that specify why the simulation system should exist [75] [82]. From these user needs a set of high level objectives and derived concrete goals are developed, which must be satisfied by the simulation system. For simulation systems these concrete goals are commonly translated into a simulation execution scenario or experiment [18]. In order to fully satisfy the user needs and objectives often multiple simulation executions are required with the simulation system during operational usage. Each of these executions may be derived from a dissimilar set of concrete goals that together completely support the user need and objectives (see *Figure 8-1*). Summarized fidelity requirements specify what the simulation system should be able to do in terms of a specification of the level of fidelity a the simulation system must display to properly satisfy the stated goals for a simulation execution. Obviously a single different concrete goal can impose other fidelity requirements on the simulation execution outcomes, which are necessary to properly accomplish this said goal. In other words the conditions for a valid set of fidelity requirements can vary with each single simulation execution. These requirements could be totally disjoint but in general each simulation execution will conform to a large set of similar fidelity requirements that are applicable to all executions performed with a simulation system.

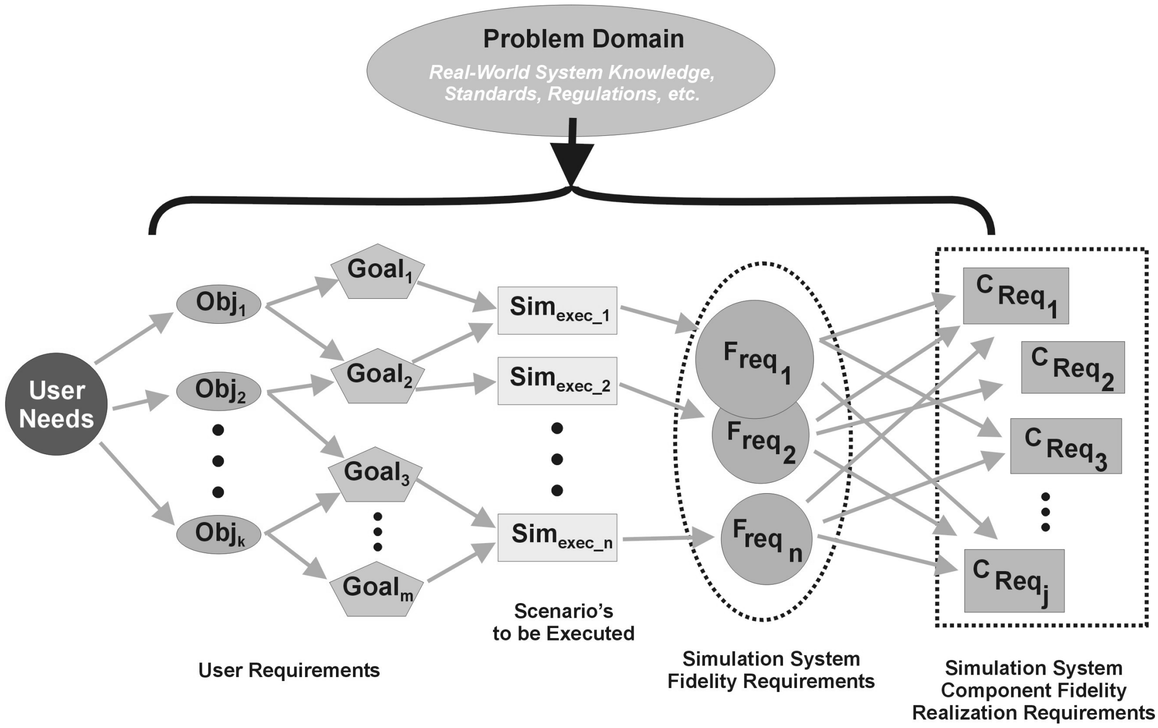


Figure 8-1 Simulation Fidelity Requirements Specification Relationships

Fidelity requirements themselves form the bases for lower level requirements for underlying constituent simulation system components. These requirements specify what is needed from each simulation system component, both hardware and software, in order to realize the required simulation fidelity during simulation execution. This type of internal system realization oriented requirements, when viewed on this level, can also be considered as a kind of fidelity requirements but now not directly related to the user objectives and needs. Therefore, this important type of requirements is referred here as fidelity realization requirements or specifications instead of fidelity requirements as defined above.

In all simulation system requirement elicitation and specification activities from left in *Figure 8-1* knowledge about the problem domain is of vital importance. The reason for this is that it is only possible to specify what is needed to create a proper representation of reality if one understands the real-world system that must be simulated. Besides gaining knowledge about this real-world also any other additional knowledge pertaining to the problem domain such as existing standards, practices and regulations must be known and taken into account.

Fidelity requirements consist of three major elements, which are discussed next. According to the definition of pragmatic simulation fidelity the fidelity of a simulation is measured against a fidelity referent R_{ref} (Chapter 4). Without such referent there is no real-world reference knowledge for measuring the actual simulation fidelity and therefore also no bases for specifying the required level of fidelity. Therefore, the specification of what kind of fidelity referent is needed is a necessary element of any fidelity requirement specification. From *Expression 4.13* it can be deduced that the specification of the required fidelity referent yields a definition of what real-world reference data knowledge is required to populate the fidelity referent knowledge-base (Chapter 5). Furthermore, it specifies how and from what sources the real-world reference knowledge is elicited, and what the quality of this real-world knowledge should be in terms of allowed error ($\delta\tilde{\Delta}_{RS}$) and uncertainty ($U_{\delta}\left(\tilde{\Delta}_{RS}^{-1}\right)$). The specification of an adequate fidelity referent is an iterative activity of subjective and intuitive nature seeking for real-world reference knowledge with enough credibility that suite or is required for the simulation application purpose at hand and it's associated risks.

The second element of a fidelity requirements specification is the definition of a set fidelity evaluator functions, which covers those real-world aspects that have to be evaluated or measured during the simulation fidelity specification of the target simulation system. Remember: due to various simulation development constraints it is practically impossible to evaluate every aspect of the simulation system against the fidelity referent. Therefore, the required fidelity evaluator function set is a subset of all possible fidelity evaluator functions such that it provides sufficient coverage of those real-world aspects of importance considering the simulation application purpose and the risks involved. Obviously, the more fidelity evaluator functions are performed the more complete the specification of the simulation fidelity will be. Furthermore, the required real-world reference knowledge contained within the fidelity referent knowledge base should be adequate for performing all these required fidelity evaluations.

Having defined both the required fidelity referent knowledge and fidelity evaluator function set it is then possible to specify the tolerated deviations of the simulated real world from this fidelity referent as quantified or qualified by the fidelity evaluator function set. These tolerances are specified in terms of an upper and lower bound that can be placed upon each of the outcomes from executing all required fidelity evaluation functions. Therefore, all tolerances together enclose an n-dimensional space inside which all measured deviations from the fidelity referent R_{ref} must reside to properly fulfill the simulation application purpose expressed by it's user and the associated risks involved.

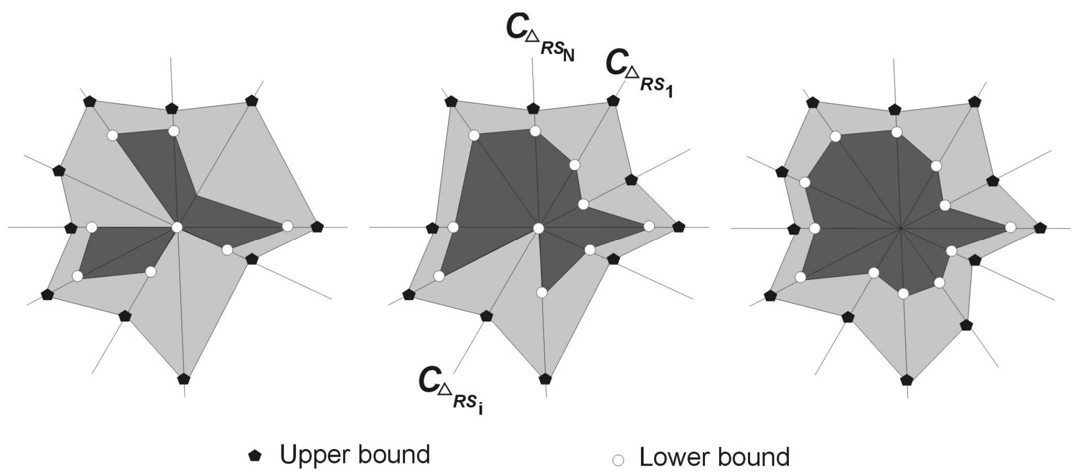


Figure 8-2 Spider graph representation of a fidelity requirement specification

To illustrate the concept of tolerances in the context of fidelity consider *Figure 8-2* at the previous page. In here three spider graphs are given, where each axes represents a one-dimensional fidelity evaluator function outcome in terms of an absolute value of the difference between for instance a certain real aircraft characteristic such as speed or altitude, and the one in the fidelity referent. A white and black dot on each axes represent the minimal and maximum allowed deviation from the referent respectively and define a range in which the measured difference must reside i.e. within the light gray area. A white dot in the center of the spider graph specifies that there is no lower bound and thus fidelity in that specific aspect direction can be infinite large. However, remember that lower bounds are necessary in some cases since a too large fidelity level can be harmful for certain simulation application purposes (Section 2.2).

8.2.2 Fidelity Requirements Formalisms

Utilizing the results of the previously deduced informal characterization of fidelity requirements, fidelity requirements or $F_{required}$ for a simulation execution are formally be defined by the following triple:

$$F_{required} = \left\langle R_{ref}^{req}, \hat{C}_{\Delta_{RS}}, T_{\hat{C}_{RS}} \right\rangle \tag{8.1}$$

Where R_{ref}^{req} is the required fidelity referent knowledge-base. Basically R_{ref}^{req} specifies what real-world reference data should be elicited and entered in the referent knowledge-base structure, how or where the reference knowledge must be elicited and any requirements about its quality in terms of acceptable uncertainties and errors (*Expression 4.13*). Furthermore, $\hat{C}_{\Delta_{RS}} \subseteq C_{\Delta_{RS}}$ is the set of required fidelity evaluation functions (*Expression 4.5*) that have to be performed and $T_{\hat{C}_{RS}}$ is the fidelity tolerances set associated with $\hat{C}_{\Delta_{RS}}$. $T_{\hat{C}_{RS}}$ is defined as follows:

$$T_{\hat{C}_{RS}} = \left\{ Tol_{c_{\Delta_1}}, \ldots\ldots, Tol_{c_{\Delta_n}} \middle| \forall c_{\Delta_i} \in \hat{C}_{\Delta_{RS}} \right\} \tag{8.2}$$

with

$$Tol_{c_{\Delta_i}} = \left(U_{bound\, c_{\Delta_i}}, L_{bound_{c_{\Delta_i}}} \right) \tag{8.3}$$

Where $U_{bound_{c_{\Delta_i}}}$ and $L_{bound_{c_{\Delta_i}}}$ are respectively the upper and lower bounds and are given by:

$$\begin{aligned} U_{bound_{c_{\Delta_i}}} &= \left\{ u_{bound_{\tilde{d}_1}}, \ldots, u_{bound_{\tilde{d}_m}} \middle| \tilde{d}_j \in \tilde{\Delta}_{RS_i} \right\} \\ L_{bound_{c_{\Delta_i}}} &= \left\{ l_{bound_{\tilde{d}_1}}, \ldots, l_{bound_{\tilde{d}_m}} \middle| \tilde{d}_j \in \tilde{\Delta}_{RS_i} \right\} \end{aligned} \tag{8.4}$$

In *expression 8.4* $u_{bound_{\tilde{d}_j}}$ represents the upper bound placed upon the value of the output variable $\tilde{d}_j$ of the j^{th} difference quantification or qualification that results from a fidelity evaluator function c_{Δ_i}. Likewise, $l_{bound_{\tilde{d}_j}}$ is the lower bound for the same outcome. In case the output variable $\tilde{d}_j$ is expressed by an alpha numerical value the next proposition should hold for its resulting value:

$$l_{bound_{\tilde{d}_j}} \leq \tilde{d}_j \leq u_{bound_{\tilde{d}_j}} \tag{8.5}$$

These types of boundaries usually apply to quantifiers for representational, functional and behavioral accuracy of the represented real-world system details (section 4.3.3). Fidelity specification in terms of resolution often results in a $\tilde{d}_j$ value that expresses the represented level of detail by means of a set of structural system elements. In those cases the output variable $\tilde{d}_j$ will always be a subset of all elements found in the structural system knowledge specification part of R_{ref} (section 3.4). This means the following proposition should hold for $\tilde{d}_j$:

$$l_{bound_{\tilde{d}_j}} \subseteq \tilde{d}_j \subseteq u_{bound_{\tilde{d}_j}} \tag{8.6}$$

As mentioned in the previous section, each simulation execution that is performed with a simulation system in order to meet the user needs may impose different fidelity requirements $F_{required}$. Therefore, the union of all fidelity requirements ($F_{required}$) for each foreseen execution with the simulation system determines the total set of fidelity capabilities required from a simulation system in order to fulfill the user needs.

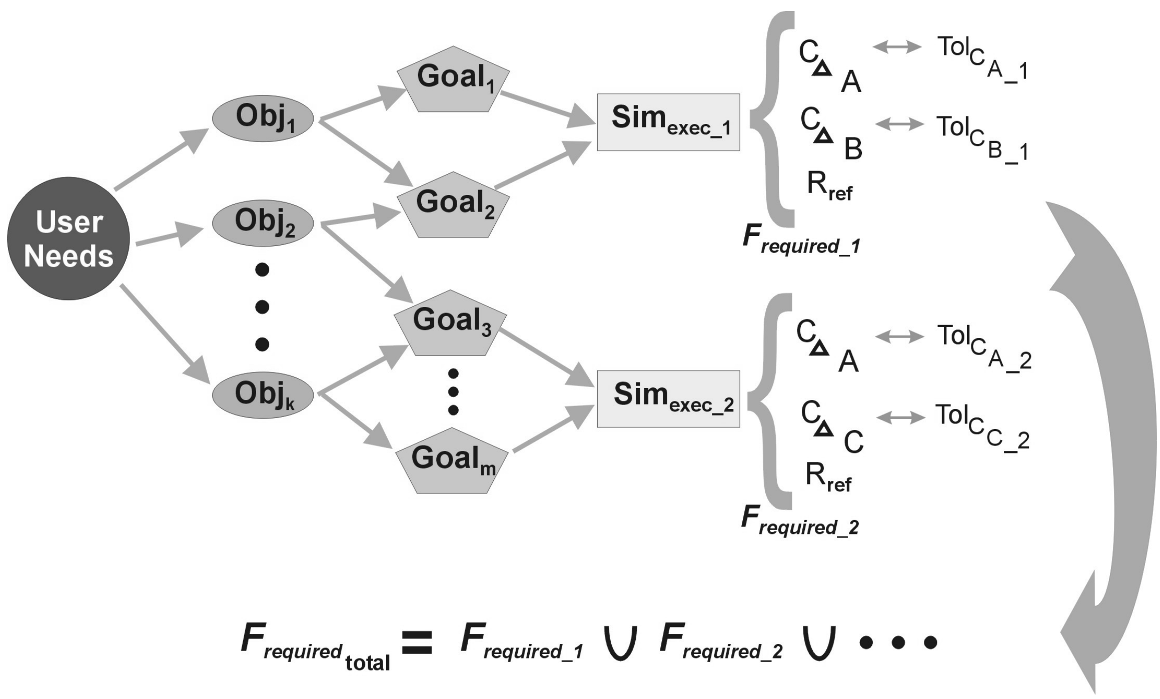


Figure 8-3 Simulation Fidelity Requirements Specification Illustration

These total simulation system requirements $F_{required_{total}}$ are the capabilities that must be established during the engineering design and development of the simulation system. $F_{required_{total}}$ is defined as follows:

$$F_{required_{total}} = \bigcup_{i=1}^{n} F_{required_i} = \left\langle R_{ref_{total}}^{req}, \hat{C}_{\Delta_{RE_{total}}}, T_{\hat{C}_{RE_{total}}} \right\rangle \qquad (8.7)$$

Where index i represents the i^{th} of the n simulation execution fidelity requirement specifications. In this expression $R_{ref_{total}}^{req}$ specifies the requirements for a single unified fidelity referent containing all real-world knowledge of such quality that it can serve all required fidelity evaluations and is defined by:

$$R_{ref_{total}}^{req} = \bigcup_{i=1}^{n} R_{ref_i}^{req} \qquad (8.8)$$

Furthermore, $\hat{C}_{\Delta_{RS_{total}}}$ and $T_{\hat{C}_{RS_{total}}}$ in *Expression 8.7* are defined as follows:

$$\hat{C}_{\Delta_{RS_{total}}} = \bigcup_{i=1}^{n} \hat{C}_{\Delta_{RS_i}} \qquad (8.9)$$

$$T_{\hat{C}_{RS_{total}}} = \left\{ \underline{Tol}_{c_{\Delta_1}}, \ldots\ldots, \underline{Tol}_{c_{\Delta_m}} \middle| \forall c_{\Delta_k} \in \hat{C}_{\Delta_{RS_{total}}} \right\} \qquad (8.10)$$

$$\underline{Tol}_{c_{\Delta_k}} = \left(\underline{U}_{bound_{c_{\Delta_i}}}, \underline{L}_{bound_{c_{\Delta_i}}} \right) \qquad (8.11)$$

Here $\underline{U}_{bound_{c_{\Delta_i}}}$ and $\underline{L}_{bound_{c_{\Delta_i}}}$ are the minimum upper and maximum lower bound set respectively, which are placed upon the evaluator function set $\hat{C}_{\Delta_{RS_{total}}}$ output values. These sets are defined as follows:

$$\underline{U}_{bound_{c_{\Delta_i}}} = \left\{ \min\left(\bigcup_{ii=1}^{n} \left(u_{bound_{\tilde{d}_1}} \right)_{ii} \right), \ldots, \min\left(\bigcup_{ii=1}^{n} \left(u_{bound_{\tilde{d}_m}} \right)_{ii} \right) \middle| \tilde{d}_j \in \tilde{\Delta}_{RS_i} \right\} \qquad (8.12)$$

$$\underline{L}_{bound_{c_{\Delta_i}}} = \left\{ \max\left(\bigcup_{ii=1}^{n} \left(l_{bound_{\tilde{d}_1}} \right)_{ii} \right), \ldots, \max\left(\bigcup_{ii=1}^{n} \left(l_{bound_{\tilde{d}_m}} \right)_{ii} \right) \middle| \tilde{d}_j \in \tilde{\Delta}_{RS_i} \right\} \qquad (8.13)$$

In here each element of $\underline{U}_{bound_{c_{\Delta_i}}}$ represents the minimum of all upper bounds placed upon the value of output variable $\tilde{d}_j$ of the j^{th} difference quantification or qualification that results from a fidelity evaluator function $c_{\Delta_i} \in \hat{C}_{\Delta_{RS_{total}}}$. Similar each element of $\underline{L}_{bound_{c_{\Delta_i}}}$ represents the maximum of all lower bounds placed upon the value of $\tilde{d}_j$.

8.2.3 Fidelity Requirements Specification Template

In this section a practical realization of the previously presented formal definition for fidelity requirements is developed in terms of the specification template as imparted within the unified simulation fidelity framework. This template plus the mathematical formalisms for fidelity requirements serve as a basis for the creation of automated fidelity tool suites to support the simulation system development and validation process. An excerpt drawn from one of the two aerospace simulation fidelity case studies is presented here to illustrate the specification of fidelity requirements using this unified fidelity framework template (Appendix C). The activities to develop and populate such a fidelity requirements specification template are discussed in Section 9.2 of this chapter as part of the fidelity management process model.

At this moment the unified fidelity framework doesn't provide any specific template for specifying the required fidelity referent. It only provides a reference to the appropriate fidelity referent. Rationale for this is the fact that development of an acceptable suitable fidelity referent for the problem at hand is an iterative process and is inextricably coupled with all requirements and conceptual model development and validation activities [103] [121]. Since these activities start with gaining real-world knowledge about the system to be simulated from a various sources this usually means that iteratively an initial body of real-world knowledge will be elicited [124]. Within the unified fidelity framework this is done by a set of subsequent steps as outlined in section 9.2. This initial body of real-world knowledge is used to populate or construct a preliminary fidelity referent knowledge-base, which is constantly updated during the first phases of the simulation system development process until its contents is accredited by subject matter experts to be acceptable or suitable for simulation fidelity measurement. In other words until the constructed fidelity referent is the one which is required given the true user needs, objectives and risks.

The unified fidelity framework requirements specification template comprises two interlinked lists, one specifying the fidelity evaluator functions and one the associated the tolerances.

The fidelity evaluator specification template contains the following elements:

- *Evaluator Function ID*: This section assigns a textual name or an identification code, by which the evaluator function can be identified, referred to and searched for.
- *Evaluation Knowledge Required*: A specification of what knowledge is required from both the fidelity referent as well as the simulation system specification to perform this fidelity evaluation.
- *Evaluation Function Description*: Specifies what the fidelity valuation comprises and how it is performed in terms of used metrics and measurement methods. The language or syntax that is suitable for this description depends on the nature of the evaluation function itself as well as the simulation system problem and application domain at hand.
- *Evaluation Output Specification*: Specifies the output vector variables, which result from performing the fidelity evaluation. This comprises name, dimension and when applicable its precision.

The fidelity tolerance template contains the following four constituent elements:

- *Evaluator Function ID*: A reference to the fidelity evaluation function for which this tolerance specification imposed on its output vector should hold.
- *Tolerance Rationale*: Description of the rationale or origin of the specified tolerance in the context of to the true user needs etc.
- *Tolerance Conditions*: Besides to which the simulation execution(s) these tolerance apply, this description should also specify any initial, termination, boundary and other applicable conditions to this fidelity evaluation.
- *Tolerance Description*: A specification of the upper and lower bounds placed upon each element of the fidelity evolution function output vector.

As an illustration of fidelity *requirements specification* consider the next excerpt from the FASE case-study (*Table 8-1*). Underlining of text represents a link to other fidelity knowledge sources where additional information is found.

Fidelity Requirements Specification		
Fidelity Referent ID	*Future Airspace Simulation Environment (FASE) Referent V0.3*	
Fidelity Evaluator Function List		
…		
Evaluation 1		
Identification	*ICAO Aircraft Coverage*	
Description	*This function specifies which aircraft types and total number of instances of the fidelity referent are represented during simulation execution in FASE*	
Required Knowledge		
Name	Description	
Airspace Entities	*Listing of all types of aircraft operating and the number of instances of each in the airspace*	
Output Specification		
Name	Function or Method	Dimension
AC Coverage Set	$R^{rw}_{aircraft} \cap M^{rw}_{aircraft}$ with $R^{rw}_{aircraft} \subseteq R^{rw}_{w} \wedge M^{rw}_{aircraft} \subseteq M_{model}$	*[-]*
AC Instance Coverage Factor	$\frac{n^{sim}_{aircraft}}{n^{ref}_{aircraft}} \cdot 100$	*[%]*
Evaluation 2		
Identification	*Normal Climb Performance Validation Test*	
Description	*This function specifies the aircraft climb performance for all engine operative by comparing the fidelity referent behavior sample with that of the simulation of an aircraft in normal cruise climb for altitude transitions above 24.000 ft.*	
Required Knowledge		
Name	Description	
AC Flight State	*A vector containing the aircraft flight state with at least the following elements: IAS, Climb-rate and Altitude*	
Output Specification		
Name	Function or Method	Dimension
Max IAS Difference	$\max\left\{ \begin{array}{l} \left\| IAS_{ref}(t_i) - IAS_{sim}(t_i) \right\| : \forall t_i \in \left[t_{h_0}, t_{h_1} \right] \\ with\ h_0 < h_1 \end{array} \right\}$	*[kts]*

Max Climb-Rate Difference	$\max\left\{\begin{array}{l}\left\|\dot{h}_{ref}(t_i)-\dot{h}_{sim}(t_i)\right\|:\forall t_i\in\left[t_{h_0},t_{h_1}\right]\\ with\ h_0<h_1\end{array}\right\}$	*[ft-pm]*

...

Fidelity Tolerances List		
...		
Tolerances Set 1		
Evaluator ID	*ICAO Aircraft Coverage*	
Rationale	*These tolerances are based on subject matter expert (ID: SME1, SME4) opinion that there is no upper bound but its lower bound must reflect the at least contain all civil transport aircraft with weight class S, M and H*	
Conditions	*These tolerances hold for every execution and condition with FASE*	
Tolerance Description		
Name	Tolerance	Dimension
AC Coverage Set	$u_{bound}: R^{rw}_{aircraft}\cap M^{rw}_{aircraft}\subseteq R^{rw}_{aircraft}$ $l_{bound}:\left\{\begin{array}{l}B744, B763, ..., \\ A321, B768\end{array}\right\}\subseteq R^{rw}_{aircraft}\cap M^{rw}_{aircraft}$	*[-]*
AC Instance Coverage Factor	$u_{bound}:100$ $l_{bound}:95$	*[%]*
Tolerances Set 2		
Evaluator ID	*Normal Climb Performance Validation Test*	
Rationale	*These tolerances are based on the FAA-AC-120-40B Document*	
Conditions	*These tolerances hold for every execution and condition with FASE*	
Tolerance Description		
Name	Tolerance	Dimension
Max IAS Difference	$u_{bound}:3$ $l_{bound}:0$	*[kts]*
Max Climb-Rate Difference	$u_{bound}:100$ $l_{bound}:0$	*[ft-pm]*

...

Table 8-1 Excerpt FASE Fidelity Requirements Specification Template

8.3 Verification & Validation: A Redefinition in Fidelity Terms

Fidelity and validity are very closely related concepts and are therefore often used in conjunction. However, as stated by fidelity theorem 6 in section 4.3 they are not each other synonyms. Fidelity is an absolute measure while validity is a relative measure with respect to the simulation application purpose. Comparison of both definitions presented in this thesis reveals that validity in the context of simulation can be characterized in terms of fidelity as follows (*Appendix A*):

'A model or simulation is considered to be valid if its level of fidelity is sufficient from the perspective the intended application purpose of the model or simulation'

A validity statement in this context is therefore something different than validity as an absolute truth confirmation in the context proving a new theory or laws for an observed phenomenon. In the next two subsections a formal definition for simulation validity in terms of fidelity will be developed along with a presentation of this definition.

8.3.1 Simulation Validity Formally Defined

Recollecting the definition and development of the fidelity requirements concept in the previous section, this yields that a model or simulation is said to be valid when the actual achieved level of fidelity meets the required level of fidelity as specified by a set of well defined and approved fidelity requirements (*Expression 8.7*). Hereby validity in the context of modeling and simulation becomes a transparent Boolean proposition in terms the fidelity adjectives *required* and *available,* which is either true or false. Using the mathematical formalisms for fidelity requirements from section 8.2.2 this can formally be written by the following logical implication:

$$F_{sufficiency} \Rightarrow V_{sim} \tag{8.14}$$

Here V_{sim} is the simulation system validity proposition, which is only true when the simulation fidelity sufficiency proposition $F_{sufficiency}$ is true. The sufficiency proposition $F_{sufficiency}$ is formally defined as:

$$\forall c_{\Delta_k} \in \hat{C}_{\Delta_{RS_{total}}}, c_{\Delta_k}\left(R_{ref}, S_{spec}\right) \in \left[\underline{L}_{bound_{c_{\Delta_k}}}, \underline{U}_{bound_{c_{\Delta_k}}}\right] \tag{8.15}$$

This proposition reads that for every fidelity evaluator function in the required fidelity evaluator set $\hat{C}_{\Delta_{RS_{total}}}$ when applied to the fidelity referent R_{ref} and the simulation system specification S_{spec}, its output vector lies within the range of the n-dimensional space encapsulated by the required minimum upper and maximum lower bound set $\underline{U}_{bound_{c_{\Delta_i}}}$ and $\underline{L}_{bound_{c_{\Delta_i}}}$ as specified by the total required simulation system fidelity $F_{required_{total}}$ (*Expression 8.7*). As an example for a fictive one-dimensional fidelity problem this means that simulation fidelity sufficiency proposition $F_{sufficiency}$ reduces to the following proposition (see *Expression 8.5*):

$$l_{bound_1} \leq c_{\Delta_1}\left(R_{ref}, S_{spec}\right) \leq u_{bound_1} \tag{8.16}$$

If this fidelity sufficiency proposition is true then by the logical implication of *Expression 8.14* the simulation is also valid.

8.3.2 Fidelity Based Verification and Validation

According to the definitions for verification and validation, the validity statement as previously formulated is the outcome of these two interlinked product evaluation processes (Section 3.3.2). The product here is the simulation system. For this reason simulation fidelity theory and practical assessment implicitly specify a basis for improving simulation verification and validation process by redefining this process in terms of fidelity [119] [126].

Fidelity-based validation comprises three major activities and is visualized in *Figure 8-4*. The first activity in this process is the development, review and evaluation of the fidelity requirements specification in order to determine whether these requirements are adequate and sufficient for properly achieving the simulation system user's needs and objectives. This involves an analysis of the risks associated in using these fidelity requirements. It tries to assess that simulation outcomes complying with these fidelity requirements have enough credibility and pose an acceptable level of risks in drawing conclusions, making acquisition decisions, providing training, etc based on these outcomes.

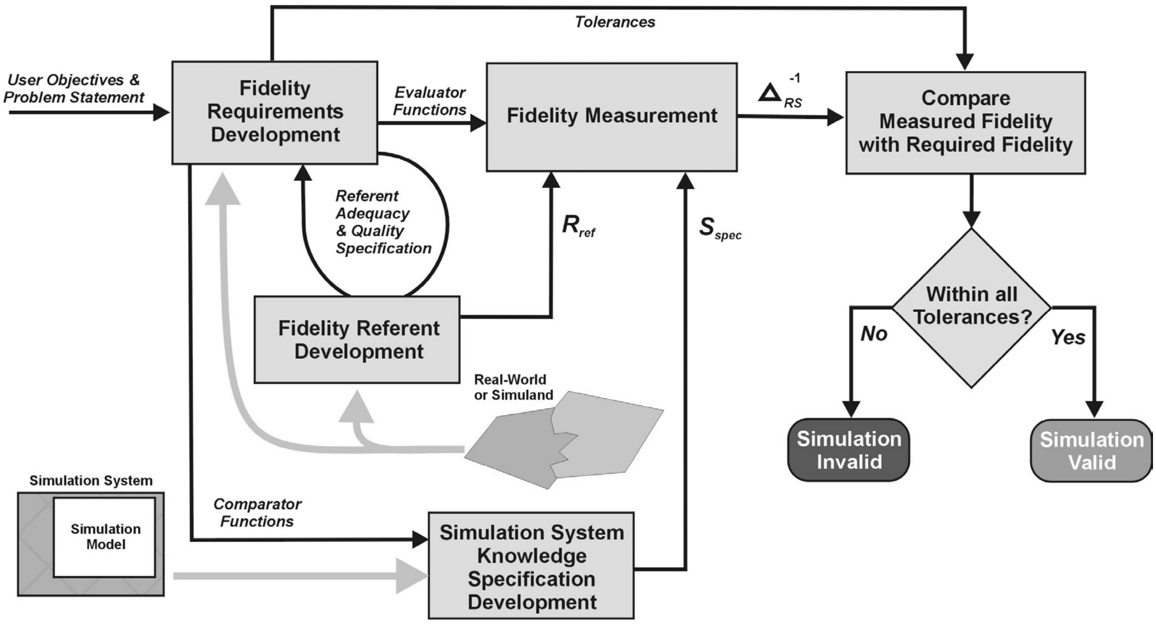


Figure 8-4 Fidelity Based Simulation Validation Process

The second activity is the fidelity measurement itself. This comprises the execution of the fidelity evaluator functions developed in activity one. The input to these functions are the fidelity referent and the simulation system knowledge specification. Fidelity measurements are conducted in two particular stage of simulation system development.

The first stage is executed at the end of the conceptual model development in order to determine what level of simulation fidelity can be attained for this given conceptual model (*Figure 3-4*). Since the simulation system isn't available at this development stage, measurements can only comprise static or execution independent fidelity evaluations. Usually, the fidelity evaluator functions applied inhere are subject matter expert evaluation, estimation and logical reasoning-based techniques about the real-world representational capabilities of the final simulation system. The conceptual model descriptions serve as the basis for the initial simulation system knowledge specification to be used for these fidelity measurements.

The second stage in which fidelity measurements are applied is during the simulation system testing (*Figure 3-4*). This involves both the measurement of the structural fidelity as well as the dynamic fidelity of the actual simulation system throughout the required simulation execution scenarios. Prior to this integrated simulation system fidelity measurement, the available fidelity capabilities of each system component are evaluated separately as soon as their development is completed (Section 8.2.1). Such an approach not only increases the credibility of the simulation system but also helps to identify any possible sources of simulation fidelity deficiencies in the system.

The third and last fidelity-based validation activity is the comparison between the measured level of fidelity and the required fidelity. This encompasses the assessment whether the measured simulation fidelity will indeed remain within all required fidelity tolerances as defined in the first validation activity. This activity can be and is performed at the end of the conceptual model development stage to assess the technical feasibility of the required level of simulation fidelity. In other words can there a valid simulation be attained with a simulation system based upon the given conceptual model. Assessing this conceptual simulation model helps to identify any possible fidelity deficiencies of the simulation system in an early stage of development. As a result prompt action can be undertaken to make required adjustments when necessary, and thus save more expensive costs when these problems have to be fixed at a later stage. Finally, this activity is also applied during simulation system testing to assess whether the required simulation fidelity has been achieved. When the total achieved simulation fidelity meets the specified fidelity tolerances the simulation system is said to be valid.

Fidelity-based verification focuses on verifying whether the conceptual simulation model and the associated fidelity realization specifications are properly translated into the operational simulation system realization. These verification activities are performed in-between the second and third validation stages during the simulation system design and its hardware and software development (*Figure 3-6*). Furthermore, the objective in here is to monitor and control the system design and development in process order to properly address any unforeseen difficulties, which might require modifications or pose limitations to the simulation system design and its implementation. The impact of these modifications and unexpected limitations on the eventual achievable simulation fidelity must be determined, checked against the fidelity requirements and when necessary action must be taken to solve any issue in properly meeting these requirements.

Once the fidelity-based verification and validation process is completed its results must be negotiated with the user to convince him about the validity of the simulation system and have him to accept the developed simulation system. This is known as accreditation (Section 3.3.2). The credibility of the verification and validation process outcomes is also determined on how concise and meticulously it is performed. Therefore it is also necessary to carefully document the verification and validation process activities themselves. Accreditation thus comprises both the verification and validation product and process evaluation in order for the user to gain enough confidence that the simulation system and its results are indeed suitable for its intended purpose. Enough confidence in this regard denotes the chance that the simulation system provides such level of fidelity during its operational usage, that the probability it will cause too harmful or hazardous effects for the application purpose is acceptably small for the user. This type of risk associated with making such a decision is known as type II error [90]; accepting the simulation system as being suitable for the intended purpose based upon the available fidelity evidence while its true simulation fidelity is insufficient.

The here discussed fidelity-based verification and validation concept will in chapter 9 be translated into a series of systematic and more concrete assessment (sub) activities as part of the fidelity management process model.

8.4 Fidelity in Simulation System Comparisons, Suitability Assessment and Trade-Off Decisions

During the development of simulation systems many evaluations and trade-off decisions must be made to fulfill the user needs, goals, functional requirements, including fidelity requirements, and the non-functional requirements, which place constraints on the development (Section 3.3). Fidelity requirements state what is necessary from the simulation system for achieving valid simulation results. Based on these requirements the simulation developer must determine whether existing simulation systems and models can be (re)used as-is or with some modifications, new ones have to be developed or combinations thereof. Such design and development decisions are made by analyzing the various available simulation fidelity specifications and comparison with the required level of fidelity for the application at hand. However, these simulation fidelity criteria are not the only criteria for making such design and development decisions. Other criteria that could have to be taken into account include all kinds of other functional requirements and constraints such as operational functionality, technical know-how, education and training, maintainability, safety, security, time-schedule, financial and organizational issues.

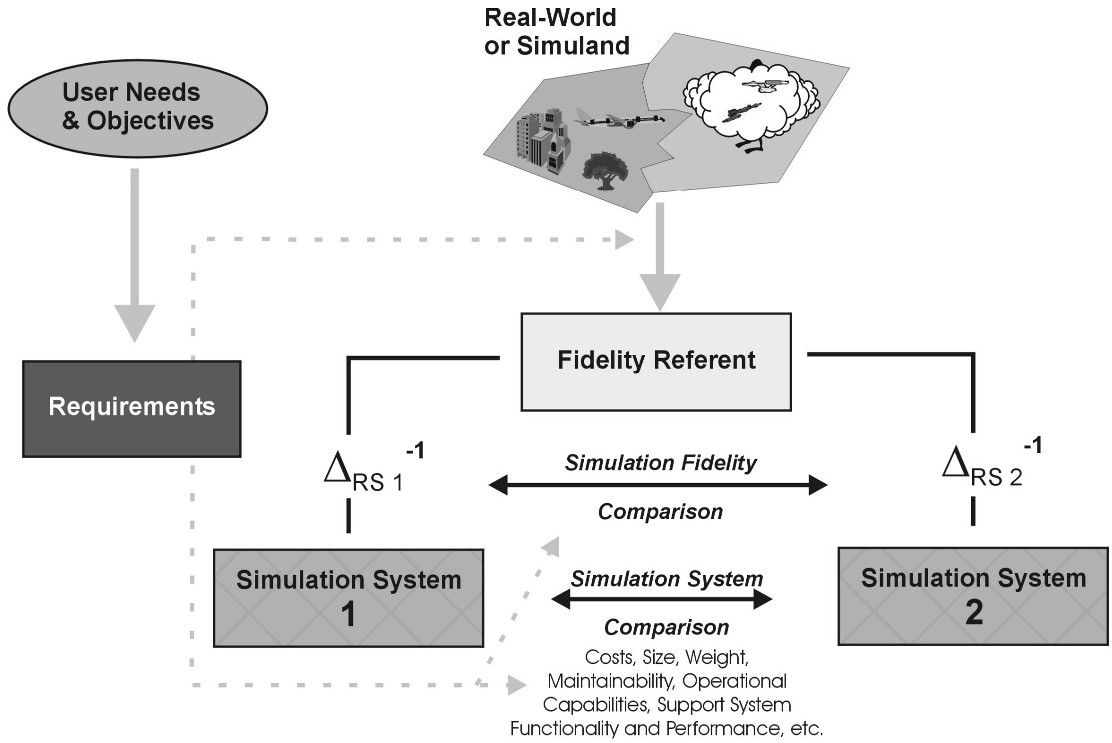


Figure 8-5 Simulation Fidelity and Simulation System Comparisons

This means that during simulation system design and development various comparisons and trade-off decisions must be made between the many fidelity criteria and the other criteria to arrive at the best possible or suitable simulation system for the user needs (*Figure 8-5*). The term best suitable simulation system is used here since in real-life it might not possible to fully meet all requirements given the circumstances and thus the most optimal design or development compromise must be selected. In relationship to fidelity requirements and validity it could be possible that due to certain resource constraints (available money, time, knowledge) not all fidelity requirements can be fully met. A perfect valid simulation system by definition cannot be attained in such case and therefore the most optimal valid simulation system must established based-on analysis of the priorities and importance of all involved criteria (Section 8.3). It should be emphasized that such analysis is not a measurement of simulation fidelity in the strict sense of the word since metrics used in here combine true fidelity measures with application depended measures. This is in accordance with the definition of fidelity and its associated theorems developed in chapter 4.

Multi-criteria analysis is a commonly used approach in systematically decision-making between alternative design and development options. This approach will be presented in the next section. The two subsequent sections will discus two examples of how the multi-criteria analysis approach can be applied in the context of the unified fidelity framework to assist in making well-founded trade-off decisions during the simulation system development and validation process.

8.4.1 The Multi-Criteria Analysis Approach

Although there exist other methods, the most widely used multi-criteria analysis method is and variations thereof are based on the Analytical Hierarchy Process (AHP) as developed by the pioneer of multi-criteria analysis, Prof. Saaty [130]. The usability and applicability of AHP for large scale and complex decision-making has been proven by many applications [27] [93] [150]. AHP is a synergy of several existing but unassociated concepts and techniques along with some new developments, which produce a process whose power is far more than the parts separately. The power of AHP is that it allows for the application of various knowledge sources, both objective and subjective, such as numerical data, experience, insight and intuition in logically and thorough mathematical way. Furthermore, it enables its users to structure complexity and systematically derive ratio scale priorities or weights as opposed to arbitrarily assigning them to make proper judgments in the whole decision process [39]. Another important aspect of AHP is that it provides compensatory decisions since alternative options that are deficient in fulfilling one or more objectives can compensate by their performance with respect to other objectives. The major AHP elements are:

1. *Construction of a Decision Hierarchy*: In this step a single complex evaluation and decision problem is decomposed into a structural hierarchy of criteria or evaluation indicators, sub- indicators, sub-sub indicators and so on (*Figure 8-6*). The root indicator of this hierarchy, not necessary a tree structure, represents qualitative statement that must be evaluated but cannot directly measured in order to make a decision. Decomposition of indicators continues until a leaf indicator is reached that can directly be measured or assessed. Only these leaf indicators are assessed for each alternative option by assigning a rating or score.
2. *Relative Critically Weighting of Indicators*: A reciprocal paired comparison of all combinations of child indicators of their parent or branch indicator. These pair wise comparisons are used to derive the relative critically influence or priority of each child indicator with respect to the outcome or score of the parent indicator. This priority is expressed by means of a relative weight, which is a scalar value between zero and one. All the child indicators weights belonging to the same parent should sum to one. These weights are computed based on SME's pair wise comparisons using the eigenvalue, mean transformation and row geometric means methods [2]

[130]. For the SME comparison a standard and well-proven pair-wise comparison scale is used [27] [39] [130]. Several SME can be used in this process to make these pair wise comparisons whose judgments can be translated in a single indicator weight through methods such as statistics and usage of SME qualification ratings (Section 5.3.3).

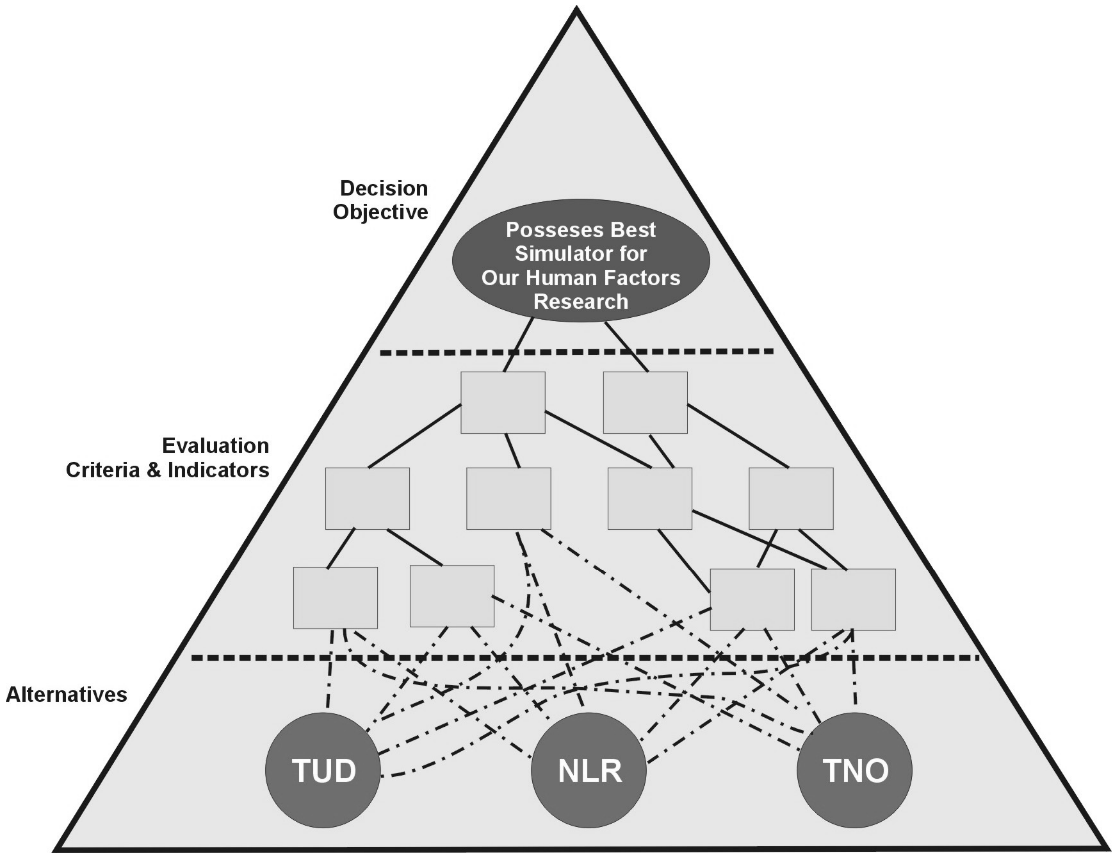


Figure 8-6 AHP Decision Hierarchy Illustration

3. *Aggregation of Scores in Branch Indicators*: The score of a branch indicator is an aggregation of the scores of its child indicators. Various methods can be used calculate this aggregated score S_{ind} but the best method is the weighted sum defined as follows:

$$S_{ind} = \sum_{i=1}^{n_{child}} w_i \cdot r_i \tag{8.17}$$

where n_{child} is the number of branch indictor children, and w_i and r_i are respectively the child indicator relative weight and rating or score.

4. *Assignment of Leaf Indicator Scores*: The set leaf indicators that are used to assign scores for each alternative can very diverse ranging from physical system properties, performance metrics to various kinds of SME ratings. To allow for proper comparisons and aggregation these scores are usually normalized and made dimensionless. Like for the weights, statistics and usage of SME qualification ratings can be used to synthesize multiple SME ratings into a single and more accurate rating for the same leaf indicator.

Once the leaf indicators scores have been assigned the whole hierarchy of scores can be calculated from bottom up to the top for each alternative option. The option with the highest root indicator score provides the best option given all criteria. Obviously, when necessary it is also possible to compare alternative options at lower levels in the hierarchy for intermediate evaluations. Such outcomes can be used to identify and analyze any bottlenecks. Another useful feature of AHP is the ability to analyze the sensitivity of a decision with respect to small changes in each alternative performance i.e. ratings as well as how sensitive the alternatives are to changes in the relative weights of each indicator. Weights with the largest sensitivity must be determined with the must be assigned with the highest accuracy.

8.4.2 Simulation Fidelity Performance Evaluation & Comparison

As formally stated in chapter 4 (*Theorem 4* and *Expression 4.23*) simulation fidelity is characterized and expressed in terms of multi-dimensional and multi-faceted array of measures that qualify and quantify how well the simulation system represent the real-world. Despite this, some sort of an overall singular metric to rate and compare simulation fidelity has always been the desire of the modeling and simulation enterprise [47] [125]. The reason for this is the fact that the dimension of this array of fidelity measures i.e. the fidelity evaluator function set $\tilde{C}_{\Delta_{RS}}$ and their total output vector set $\tilde{\Delta}_{RS}$ dimension (*Expression 4.18 and 4.19*) can be very large. Therefore, its results may be hard and time consuming to compare, analyze or interpret. One part of the solution to this problem is the availability and use of an automated tool. The other part is the use of a summarizing metric, which can be used in the simulation system life cycle to express its fidelity capabilities on various higher aggregated levels or sub-areas. Such a metric provides an efficient means for an initial judgment and comparison of the fidelity capabilities of a simulation system. Based on these judgments and comparisons a fast and focused pre-selection of alternative design solutions and reusable simulation system, components, etc. can be made. After this pre-selection the resulting candidates are then subjected to the necessary detailed fidelity analysis and comparison of each separate fidelity evaluation function outcome in combination with the fidelity effectiveness process (Section 8.4.3).

To accommodate the use of such high-level simulation fidelity capability statements or specifications within the unified fidelity framework, an application independent summarizing metric is used, called a fidelity performance metric $F^{perform}_{\tilde{\Delta}_{RS_i}}$. The basis for this metric is a rating function $f^{perform}_{\tilde{d}_j}$ operating on each j^{th} element of the output set $\tilde{\Delta}_{RS_i}$ of a fidelity evaluator function. This performance rating function specifies the fidelity performance capabilities in terms of a dimensional less and comparable scalar. A fidelity performance metric, independent of the type of its underlying fidelity evaluator function, can be constructed accordingly as long as it complies with the following four axioms:

$$(1)\quad \lim_{\tilde{d}_j \to \pm\infty} f^{perform}_{\tilde{d}_j}\left(\tilde{d}_j\right) = 0$$

$$(2)\quad \lim_{\tilde{d}_j \to 0} f^{perform}_{\tilde{d}_j}\left(\tilde{d}_j\right) = 1$$

$$
\begin{aligned}
&(3)\quad \tilde{d}_j > \tilde{d}_j^* \Rightarrow f_{\tilde{d}_j}^{perform}\left(\tilde{d}_j\right) > f_{\tilde{d}_j}^{perform}\left(\tilde{d}_j^*\right) \\
&(4)\quad \tilde{d}_j < \tilde{d}_j^* \Rightarrow f_{\tilde{d}_j}^{perform}\left(\tilde{d}_j\right) < f_{\tilde{d}_j}^{perform}\left(\tilde{d}_j^*\right)
\end{aligned}
\tag{8.18}
$$

Axiom 1 states that if the measured difference between the real world and the simulation system outcome becomes very large, the system performance in representing the real world becomes zero. Axiom 2 states that when the measured difference becomes very small the fidelity performance rating approaches to one. One indicates that there is no difference i.e. the simulation system is the real world. Except for simulation systems that include some real hardware-in-the-loop systems, this won't occur in practice. Axioms 3 and 4 state that with an increasing measured difference the fidelity performance continuously descents, which is in accordance with the normal apprehension that a larger difference between the real world and the simulation outcomes results in a lower level of fidelity (Section 4.2.1). The range of any fidelity performance rating function thus varies from one to zero. The fidelity performance metric of the i^{th} fidelity evaluator function $c_{\Delta_{RS_i}}$ output vector can now be defined by the following arithmetic mean:

$$
\begin{aligned}
&F_{\tilde{\Delta}_{RS_i}}^{perform} = \frac{1}{k}\sum_{j=1}^{k} f_{\tilde{d}_j}^{perform} \\
&with \\
&k = \left|\tilde{\Delta}_{RS_i}\right|
\end{aligned}
\tag{8.19}
$$

$F_{\tilde{\Delta}_{RS_i}}^{perform}$ provides thus an application independent and dimensionless measure for how well a portion of the simulation system on average performs in representing the real-world i.e. gives an overall score for the system's fidelity capabilities.

In practice the fidelity evaluator function set $\tilde{C}_{\Delta_{RS}}$ is the union of two or more hierarchically nested subsets. These subsets combine fidelity evaluator functions that are related to each other according some grouping criteria. Each of these areas can be further subdivided in smaller sets according a specific aspect of the real world that is evaluated by a subset of evaluator functions (See *Figure 8-9*). These nested subsets of fidelity evaluator functions represent a hierarchical tree structure, which with the use of the previously defined fidelity performance metric can be used to specify the overall or average simulation fidelity performance score for various aggregations levels and areas. This process is similar to the multi-criteria approach outlined in section 8.4.1 with this difference that the weights assigned to each child indicators for an aggregating branch score are inverse proportional to the number of branching child indicators. With this in mind the total fidelity performance score for a branch or a nested subset of fidelity evaluator function can now be defined as follows:

$$
F_{branch_j}^{perform} = \frac{1}{n_{child}}\sum_{k=1}^{n_{child}} F_{\tilde{\Delta}_{RS_k}}^{perform}
\tag{8.20}
$$

In *expression 8.20* n_{child} is the number of child indicators for the j^{th} branch or number elements in the subset of fidelity evaluator functions. Since all fidelity performance functions result in a rating between zero and one, also the total fidelity performance at any aggregation level will result in score in this range. Therefore, the higher the total fidelity performance score is, the better the simulation systems on average performs in representing the whole or a smaller portion of the real world. With the use of this metric it is possible to quantitatively specify, organize and evaluate the various overall fidelity performance capability differences of simulation systems in an application independent and objective manner. For instance when comparing three simulation models on its fidelity performance in a specific evaluation area, it is possible to make the following kind of formal raking statements about which one on the average performs better than the other:

$$F_{hq_3}^{perform} > F_{hq_1}^{perform} > F_{hq_2}^{perform} \tag{8.21}$$

However, according fidelity *Theorem 10* one should keep in mind that a in a comparison of simulation systems a higher fidelity performance score is necessary but not a sufficient condition for the statement that fidelity is also higher (Section 4.3.2). As outlined by this theorem the statement that a simulation system A in a pair-wise comparison with a simulation system B has a higher level of fidelity is only legitimate when for the same fidelity referent R_{ref} and fidelity evaluator set $\tilde{C}_{\Delta_{RS}}$ the following proposition is true:

$$\begin{aligned}
&\forall\left(d_{RS_a}^{-1}, d_{RS_b}^{-1}\right) \in C_{AB}\ ,\ d_{RS_a}^{-1} > d_{RS_b}^{-1}\\
&with\\
&C_{AB} = \left\{\left(d_{RS_a}^{-1}, d_{RS_b}^{-1}\right) \middle| d_{RS_a}^{-1} \in \overline{\Delta}_{RS_A}^{-1} \wedge d_{RS_b}^{-1} \in \overline{\Delta}_{RS_B}^{-1}\right\}
\end{aligned} \tag{8.22}$$

When *Proposition 8.22* is true then by definition of $F_{\tilde{\Delta}_{RS_i}}^{perform}$ (*Expression 8.19*) also the fidelity performance of simulation system A will be better than that of simulation system B. Furthermore, both expressions clearly illustrate the major drawback of the fidelity performance measure and any other overall fidelity rating metric, which is that information about possible pair-wise differences may be averaged out and thus not seen in an overall score. Despite its very useful technical utility in a first evaluation, comparison and selection between various simulation systems, component or real-world models, this metric is therefore no substitute for a detailed low-level fidelity analysis and specification in the final assessment of simulation suitability and validity.

8.4.3 Simulation Fidelity Effectiveness Evaluation & Trade-Off Decisions

During simulation system development one often has to compare and evaluate existing simulation systems, federates, simulation model components or real-world system models for their appropriateness in meeting all or a specific subset of fidelity requirements. It is not uncommon in practical modeling and simulation that certain parts of the requirement are not fully met by all available alternatives. In other words there isn't a perfect alternative that meets the fidelity sufficiency and validity proposition (*Expression 8.14 and 8.15*). Therefore, one has to select the alternative that best fits the simulation fidelity requirements instead. This selection usually involves the evaluation of a complex set with various kinds of fidelity measures and the relative importance of each fidelity requirement in achieving the user needs.

In the unified fidelity framework the multi-criteria approach is utilized to transform this multidimensional decision problem into a one-dimensional problem, which evaluates the alternatives by means of the so-called total simulation fidelity effectiveness ($F_{tot}^{effective}$). This overall score characterizes degree of validity of the alternatives, i.e. how effectively the available fidelity of a simulation system and/or its constituent components suit the given specific set of fidelity requirements. The basis for this total simulation fidelity effectiveness is the fidelity effectiveness rating function or metric ($f_{\tilde{d}_j}^{effective}$), which specifies the degree of how well the value of an output variable $\tilde{d}_j$ for a fidelity evaluator function c_{Δ_i} meets its associated tolerances *(Expression 8.3*). Depending on the type of fidelity evaluator function and output, a fidelity effectiveness metric can be constructed accordingly as long as it complies with the following four axioms:

$$\begin{aligned}
&(1)\ f_{\tilde{d}_j}^{effective}:\left(\tilde{d}_j, l_{bound_{\tilde{d}_j}}, u_{bound_{\tilde{d}_j}}\right)=1 \quad \text{for } \forall \tilde{d}_j \in \left[l_{bound_{\tilde{d}_j}}, u_{bound_{\tilde{d}_j}}\right] \\
&(2)\ u_{bound_{\tilde{d}_j}} < \tilde{d}_j < \tilde{d}_j^* \Rightarrow f_{\tilde{d}_j}^{effective}\left(\tilde{d}_j, l_{bound_{\tilde{d}_j}}, u_{bound_{\tilde{d}_j}}\right) \geq f_{\tilde{d}_j}^{effective}\left(\tilde{d}_j^*, l_{bound_{\tilde{d}_j}}, u_{bound_{\tilde{d}_j}}\right) \\
&(3)\ l_{bound_{\tilde{d}_j}} > \tilde{d}_j > \tilde{d}_j^* \Rightarrow f_{\tilde{d}_j}^{effective}\left(\tilde{d}_j, l_{bound_{\tilde{d}_j}}, u_{bound_{\tilde{d}_j}}\right) \geq f_{\tilde{d}_j}^{effective}\left(\tilde{d}_j^*, l_{bound_{\tilde{d}_j}}, u_{bound_{\tilde{d}_j}}\right) \\
&(4)\ \lim_{\tilde{d}_j \to \pm\infty} f_{\tilde{d}_j}^{effective}\left(\tilde{d}_j, l_{bound_{\tilde{d}_j}}, u_{bound_{\tilde{d}_j}}\right)=0
\end{aligned} \tag{8.23}$$

Axiom 1 states that whenever a fidelity evaluator function outcome is within the fidelity tolerances the alternative is effectively representing that particular aspect of the real-world system. Axioms 2 and 3 state that the fidelity effectiveness for a given fidelity aspect is reducing with the increasing distance between either the upper or lower bounds and the actual fidelity evaluator function outcome. A smaller fidelity effectiveness rating thus implies a better fulfillment of the set fidelity requirements. According to axiom 4 the fidelity effectiveness rating should always reduce to zero when the distance between either the upper or lower bounds and the actual fidelity evaluator function outcome becomes very large. A zero value indicates that the alternative isn't capable to represent this aspect of the real world. The range of any fidelity effectiveness rating function thus varies from one to zero. *Figure 8-7* at the next page shows two typical examples of $f_{\tilde{d}_j}^{effective}$ functions.

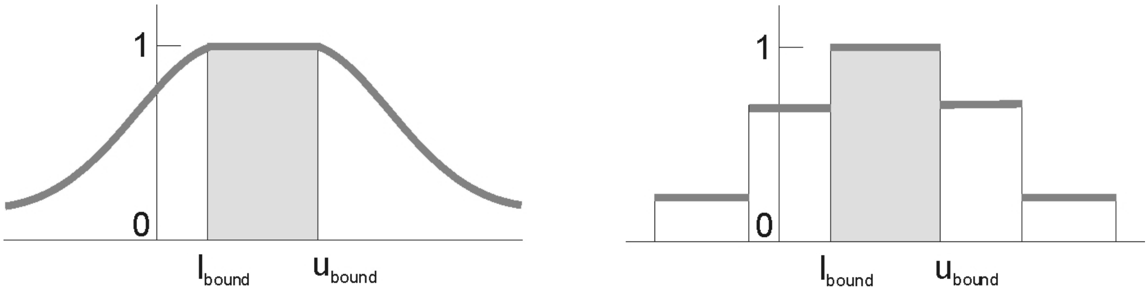


Figure 8-7 Fidelity Effectiveness Rating Function Samples

A fidelity evaluator function by definition has multiple outcomes that characterize the level of fidelity for that specific evaluation area. Therefore an aggregated overall fidelity effectiveness score can be constructed for the complete output set $\tilde{\Delta}_{RS_i}$ each fidelity evaluator function, which specifies how effective the total set of associated tolerances is met. This overall fidelity effectiveness score of the i^{th} fidelity evaluator function c_{Δ_i} output vector is defined as follows:

$$F_{\tilde{\Delta}_{RS_i}}^{effective} = \sum_{j=1}^{k} w_j \cdot f_{\tilde{d}_j}^{effective}$$
$$with \qquad (8.24)$$
$$k = \left|\tilde{\Delta}_{RS_i}\right|$$

In here w_j is defined as the j^{th} weight of the normalized relative importance weight vector for the i^{th} fidelity evaluator function output vector:

$$W_{\tilde{\Delta}_{RS_i}} = \left(w_1, w_2, ..., w_k\right) \qquad (8.25)$$

These weights are assigned by means of SME judgments using the Saaty's pair-wise comparison scale [2]. To normalize this vector each element is divided by the length of the weight vector. As a result of a normalized weight vector the range of $F_{\tilde{\Delta}_{RS_i}}^{effective}$ varies from zero to one. A value of one indicates that the alternative effectively meets all fidelity requirements for that part of the real world evaluated by c_{Δ_i}. Zero means that the alternative isn't capable to represent this whole portion of the real world.

Recalling *Figure 8-1* developed in section 8.2.1 one sees that the total simulation requirements specification to fulfill the final user needs or simulation application purpose is hierarchical structure. A structure that decomposes the user needs in a set of high-level objectives that must be accomplished, a set of more concrete goals that must be fulfilled to achieve the high-level objectives. A goal may be decomposed in (sub) goals or sets of simulation executions that must be performed to accomplish each goal. At the bottom there are the fidelity requirements that must be met for each simulation execution. The AHP approach is used to assign relative importance weights to each branching indicator in the simulation requirements hierarchy (*Expression 8.23*). Once these weight vectors have been assigned the scores can be aggregated bottom up by means of a weighted sum (*Expression 8.17*), starting at the leaves with the calculation of the overall fidelity effectiveness rating according to *Expression 8.22*. Finally at the root, the total overall fidelity effectiveness result indicates how effectively the available simulation system fidelity suits the user needs or application purpose.

This fidelity effectiveness evaluation process is illustrated in *Figure 8-8* at the next page. For brevity and clarity of figure the fidelity effectiveness rating functions are presented as a large non-structured set. As discussed in the previously in practice the required fidelity evaluator functions are usually grouped into hierarchically structured subsets. This means that the overall fidelity effectiveness also at these fidelity evaluator levels can be evaluated by assigning normalized relative importance weight vectors to each of these subsets. Doing so will help to specify and judge the relative importance of fidelity requirements and their fulfillment by the alternatives from more abstract and naturally defined simulation fidelity areas of interest.

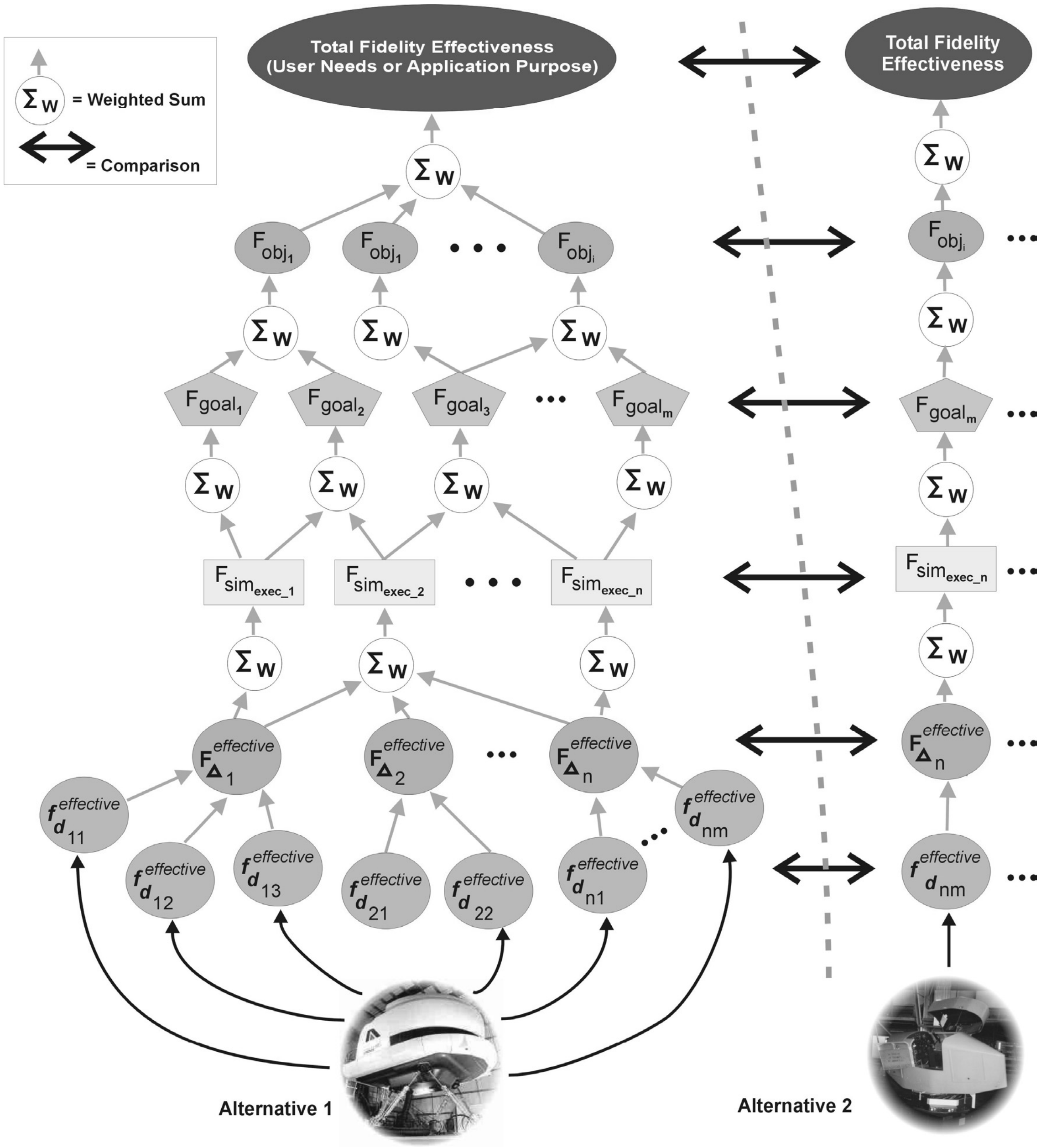


Figure 8-8 Simulation Fidelity Effectiveness Evaluation Process Sample

Since the fidelity effectiveness ratings always produce a value in the range of zero to one and the fact that normalized weights are used, the total overall fidelity effectiveness score will also have this range as well as any intermediate or local score in the simulation fidelity effectiveness hierarchy. A value zero at the root indicates that the alternative (simulation system, federate, component, real-world model, etc.) is totally unable to meet the specified fidelity requirements i.e. the alternative is completely invalid. Likewise, a value of one at the root indicates that the alternative effectively fulfills all the specified fidelity requirements for the application purpose. In other words the alternative is completely valid as formally defined by fidelity sufficiency proposition (*Expressions 8.14 and 8.15*). Any other value between zero and one thus yields that some fidelity requirements haven't been fully met i.e. the alternative is only partially valid. In this case a higher rating means a better fulfillment of the fidelity requirements as set for the simulation system application purpose. Therefore the fidelity effectiveness metric and evaluation process can also be used during validation when simulation system is not perfectly valid to quantify its total or local overall degree of validity.

The fidelity effectiveness evaluation process also provides a sensitivity analysis means to systematically assess where the most critical fidelity requirements occur and judge the implications of their violation on the overall validity of the simulation system. This kind of evaluation therefore also addresses tradeoffs in the context of any possible risks in using the simulation system and their acceptability. In this way fidelity effectiveness evaluation can be used to make the most optimal decisions regarding to where invest the most or additional resources, when necessary, during simulation system design and development.

Making selections and trade-off decisions between various alternatives involves more criteria than only the simulation fidelity effectiveness. These criteria result from the other functional or operational requirements and non-functional requirements, which the user placed on the development process of the simulation system. Examples of such other decision criteria have been mention in the introduction of this section. Therefore, in practice the fidelity effectiveness process is always an integrated part of a lager AHP system. An AHP system in which fidelity effectiveness versus other criteria trade-off analysis can be performed in order to make well founded and cost-effective simulation system design and development decisions.

8.5 Fidelity Application Concepts & Techniques Samples

In this section the fidelity application concepts and techniques discussed in the forgoing sections of this chapter will be illustrated by means of some samples. These samples have been drawn from the IAe CN235-200M case-study, which is discussed in Appendix C.

8.5.1 Fidelity-Based Validation Sample

Simulation validity has formally been defined in terms of fidelity in section 8.3.1. The fidelity-based assessment of simulation validity comprises three major steps (Section 8.3.2), which will be illustrated in the next paragraphs.

Fidelity Tolerances and Evaluator Function Hierarchy Development

Simulation validity is determined based upon the fidelity sufficiency proposition (*Expression 8.15*). This fidelity sufficiency proposition is deduced from the simulation fidelity requirements placed upon the simulation system and its underlying components. The basis for the fidelity requirements of IAe CN235-200M flight simulation model was driven by the external constraint that the whole simulator system had to comply with the FAA airplane simulator qualification advisory circular. Particular that the simulator had to approximate the in here described level D requirements. Within this advisory circular the fidelity evaluator functions are subdivided in three major high-level area according the type of evaluation: functional test, subjective testing, and flight data base testing. Each of these three areas is subdivided in smaller sets according specific aspects of the real-world that are evaluated by a subset of evaluator functions. This results in an evaluator function hierarchy as discussed in section 8.4.2 and is depicted by means of a Venn diagram in *Figure 8-9* below.

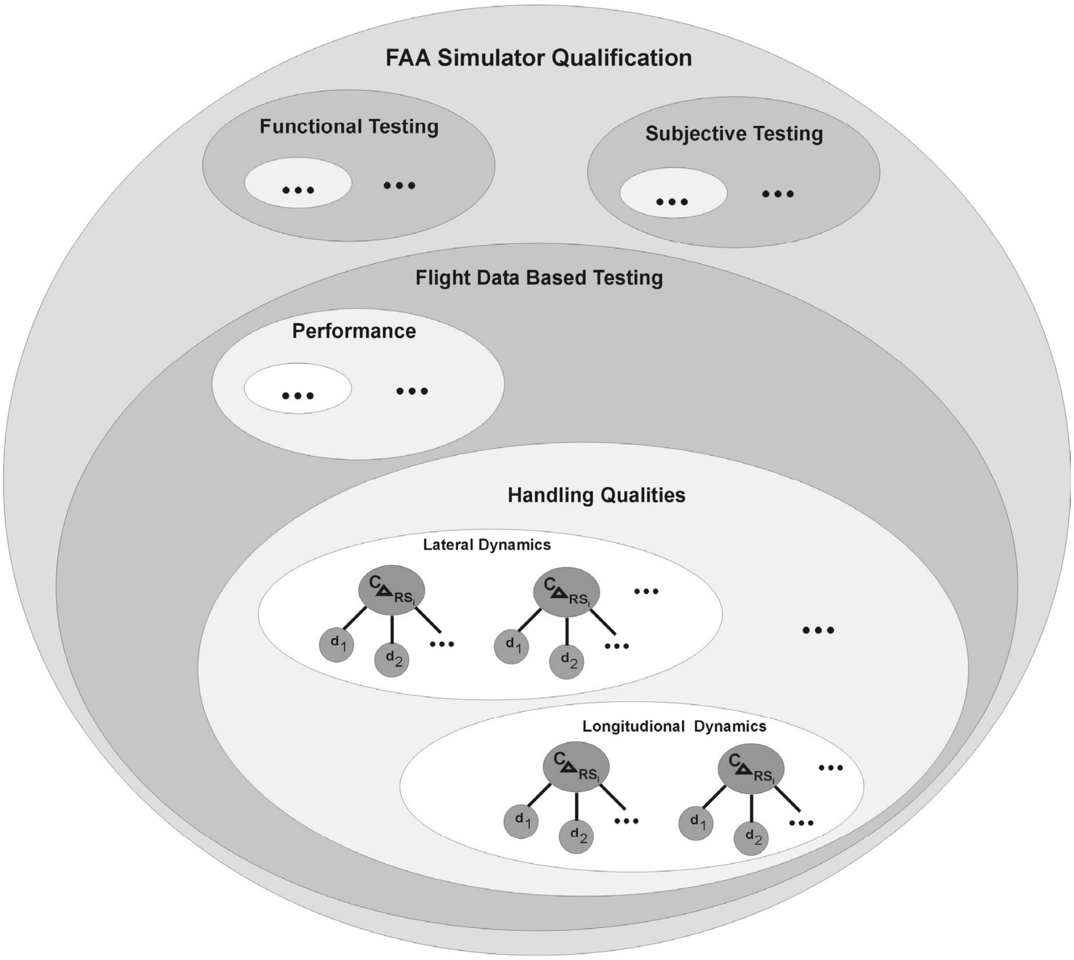


Figure 8-9 CN235 Simulator Fidelity Evaluator Function Set Venn Diagram

For brevity the focus here will be on the longitudinal dynamics subset of the handling qualities set and particular the phugoid and short period dynamics fidelity evaluations (*Table 8-2* and *8.3*). The total subset comprises eleven multi-dimensional fidelity evaluations. Within the CN235 simulator project no additional evaluator functions above these eleven have been added. However, the dimension of each evaluator function has been increased with additional metrics to provide supporting quantitative evidence for the level of fidelity.

Trajectory	Metric or Method	Tolerances	FAA
Theta	*Period Difference (Expression 7.39)*	$\left\| f_{diff}^{period} \right\| \leq 0.1 \cdot t_{ref}^{period}$	*Yes*
Theta	*Time to Half (Expression 7.36)*	$\left\| f_{diff}^{half} \right\| \leq 0.1 \cdot t_{ref}^{half}$	*Yes*
Theta	*Abs Max Error (Expression 7.25)*	$f_{diff}^{max} \leq 2^0$	*No*

Table 8-2 Phugoid Dynamics Fidelity Evalution Metrics and Tolerances

Trajectory	Metric or Method	Tolerances	FAA
Pitch Rate	*Abs Max Error (Expression 7.25)*	$f_{diff}^{max} \leq 2^0 / s$	*Yes*
N_{normal}	*Abs Max Error (Expression 7.25)*	$f_{diff}^{max} \leq 0.1g$	*Yes*
Pitch Rate	*Integrated Error (Expression 7.27)*	$f_{diff}^{abs} \leq 0.05 \cdot \int \left\| \mathrm{q}_{ref}(t) \right\| \cdot dt$	*No*
N_{normal}	*Integrated Error (Expression 7.27)*	$f_{diff}^{abs} \leq 0.25 \cdot \int \left\| \mathrm{N}_{ref}(t) \right\| \cdot dt$	*No*

Table 8-3 Short Period Dynamics Fidelity Evalution Metrics and Tolerances

Performing Fidelity Measurements

All the fidelity evaluations with the developed IAe CN235-200M flight simulation model were conducted within a dedicated automated tool. This tool provides a trim routine that trims the flight simulation model to the initial condition of the actual measured flight test trajectory. Furthermore, it can read and feed the original measured control system input into the simulation model during simulation. As a result any of the simulation model trajectory output differences with the actual flight test data are solely induced by the simulation model realization itself. The tool also provides analysis capabilities to plot both simulated and real trajectories for visual inspection, automatic fidelity metric calculations plus their result presentation and storage. The next figures and tables provide some outcomes of fidelity evaluations performed with this tool for the required phugoid and short period dynamics discussed in the previous paragraph.

From *Figure 8-10* the period and time to half values have been derived for both the reference phugoid measured in the real flight and the simulated one. Table below summarizes these results. The absolute maximum error in pitch angle is 1.2 [deg] as can be seen in *Figure 8-11*.

Metric	Flight Test	Simulation	*Abs Difference*
Period	*36.1 [s]*	*35.3 [s]*	*0.8 [s]*
Half Time	*77.1 [s[*	*79.5 [s]*	*2.4 [s]*

Table 8-4 Phugoid Period and Half-Time Fidelity Measurement Results

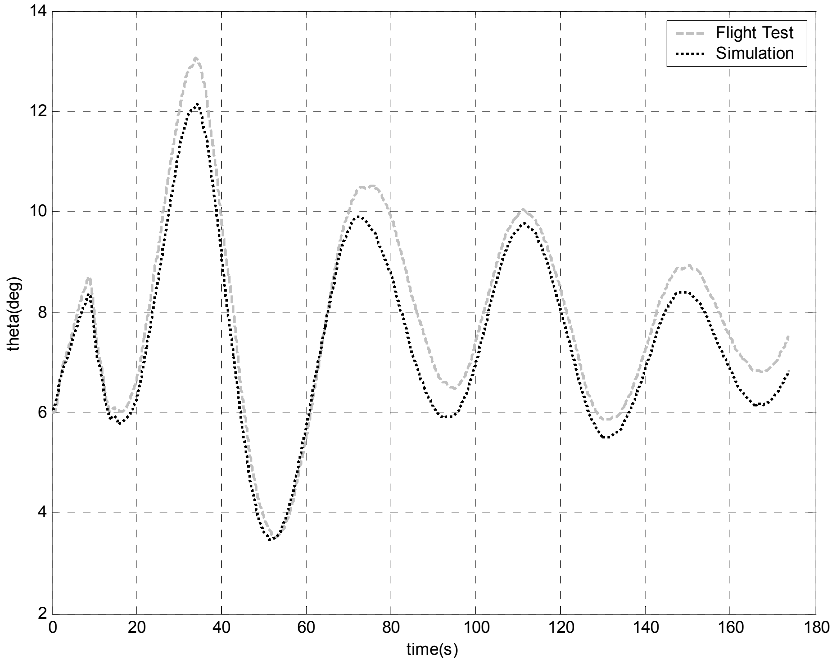


Figure 8-10 Flight Test and Simulated Phugoid

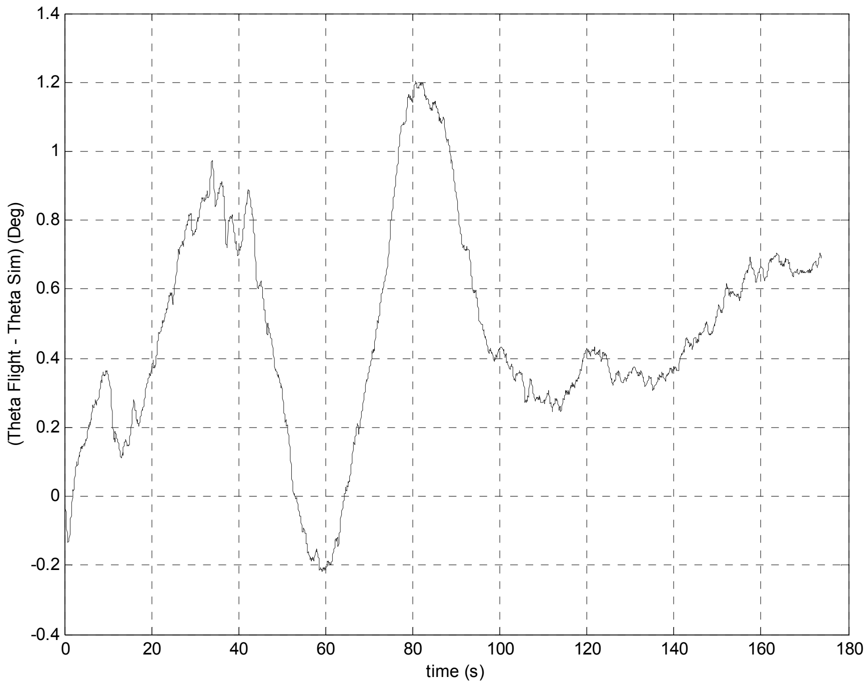


Figure 8-11 Measured Phugoid Theta Error

The next two figures present the measured differences between the normal acceleration and pitch rate of the reference short period measured in the real flight and the simulated ones.

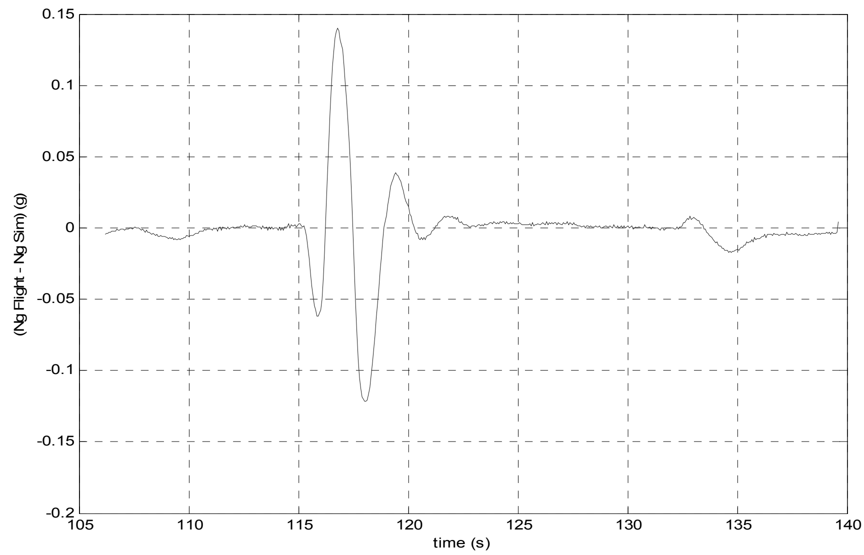


Figure 8-12 Measured Short Period Normal Acceleration Error

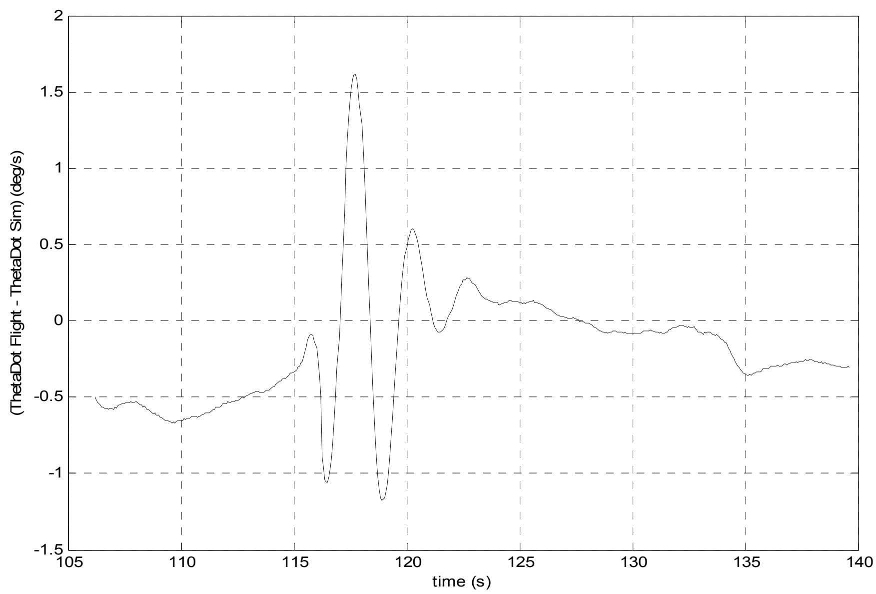


Figure 8-13 Measured Short Period Pitch Rate Error

From the previous two figures the absolute maximum error in normal acceleration is determined to be 1.38 [g] and the absolute maximum error in pitch rate 1.63 [deg/s]. The integrate absolute errors calculated from these figures are presented in the next table along with the integrate referent signal.

Variable	Integrated Difference	Integrated Referent Signal
Norm Acc.	*0.389*	*0.174*
Pitch Rate	*11.36*	*227.46*

Table 8-5 Short Period Absolute Integrated Error Results

Evaluation of the Validity Proposition

Due to its definition the evaluation of the validity proposition simply involves checking whether all specified tolerances for each fidelity evaluator function output are met by the actual fidelity evaluator function results. In case all outcomes are within their associated tolerances the validation proposition is true otherwise it is false.

From *Table 8-2* and *Table 8-4* it can be concluded that for the phugoid dynamics all fidelity tolerances have been met the CN235 flight simulation model. Therefore the validity proposition is true and can be stated that the model provides a valid representation of the CN235 phugoid dynamics. However, the validity proposition for the short period dynamics is false, since only the pitch rate tolerance have been met.

Several straight forward metrics can be applied to specify the degree of validity in case the validity proposition is false. Examples include range error and accuracy like metrics such as defined by *Expression 7.17* in section 7.3.1 but now with respect to the tolerance boundaries. However, a more sophisticated metric for this purpose is the already developed simulation fidelity effectiveness evaluation and which will be demonstrated in next section of this chapter.

8.5.2 Fidelity Performance and Effectiveness Samples

This section will build upon the samples presented in the previous section to illustrate the practical usage and application of both fidelity performance and effectiveness evaluation within the modeling and simulation enterprise.

Calculation of Fidelity Performance

To calculate the local and global fidelity performances a fidelity performance metric must be defined for each fidelity evaluator function outcome. The axioms to which such metric must comply with are given by *expression 8.18*. A generic function that has been used for this purpose in the case-studies is:

$$f_{\tilde{d}_j}^{perform}\left(\tilde{d}_j\right) = 1 - \frac{\tilde{d}_j}{\tilde{d}_j + \dfrac{1}{k_j}} \qquad (8.26)$$

Where k_j is an appropriate chosen scaling factor. This function is similar to several proportional scaled equivalents developed in Chapter 7. *Table 8-6* and *Table 8-7* gives the calculated fidelity performances for the phugoid and short period dynamics fidelity evaluation outcomes. The total fidelity performance $F_{\tilde{\Delta}_{RS_i}}^{perform}$ of the phugoid and short period dynamics evaluation function calculated by *expression 8.19* is also given in these tables.

Trajectory	Metric or Method	$f^{perform}$	$F^{perform}$
Theta	*Period Difference*	*0.96*	*0.94*
Theta	*Time to Half*	*0.93*	
Theta	*Abs Max Accuracy*	*0.91*	

Table 8-6 Phugoid Dynamics Fidelity Performance

Trajectory	Metric or Method	$f^{perform}$	$F^{perform}$
Pitch Rate	*Abs Max Accuracy*	*0.88*	*0.66*
N_{normal}	*Abs Max Accuracy*	*0.51*	
Pitch Rate	*Integrated Error*	*0.90*	
N_{normal}	*Integrated Error*	*0.35*	

Table 8-7 Short Period Dynamics Fidelity Performance

The above fidelity performance results together with the other nine evaluations in the longitudinal dynamics fidelity evaluation subset (*Figure 8-9*) results according *expression 8.20* in a local overall fidelity performance rating of 0.85 for the longitudinal dynamics replication. Similarly the lateral dynamics fidelity performance rating has been calculated and resulted in a rating of 0.79. From this result it can be postulated that the developed CN235 simulation model replicates the longitudinal dynamics better than the lateral flight dynamics. This was also confirmed by means of subjective testing by IAe's test pilots. Working all way up to the flight data-based testing level resulted in a final global fidelity performance rating of 0.84. Unfortunately, the flight data-based test results of IAe's old flight simulation model weren't disclosed to the Delft team. Otherwise it would have been possible to apply the same performance calculations to the old simulation results and make a comparison to determine how much the fidelity performance has been improved. This could have been done globally but also locally to see where the maximum and minimum improvements in fidelity have been achieved by the new flight simulation model.

Fidelity Effectiveness Rating and Weighting Functions Development

The first thing that has to be done to able to calculate the fidelity effectiveness is to develop fidelity effectiveness rating functions based upon the specified tolerances for each fidelity evaluator function outcome. For simplicity a discrete effectiveness rating function is applied to all outcomes that is defined as follows:

$$f_{\tilde{d}_j}^{effective} : \left(\tilde{d}_j, l_{bound_{\tilde{d}_j}}, u_{bound_{\tilde{d}_j}}\right) = 1 \quad \text{for } \forall \tilde{d}_j \in \left[l_{bound_{\tilde{d}_j}}, u_{bound_{\tilde{d}_j}}\right]$$

$$f_{\tilde{d}_j}^{effective} : \left(\tilde{d}_j, l_{bound_{\tilde{d}_j}}, u_{bound_{\tilde{d}_j}}\right) = 0.7 \text{ for } \forall \tilde{d}_j \in \left[l_{bound_{\tilde{d}_j}}, l_{bound_{\tilde{d}_j}} + 0.1\left(l_{bound_{\tilde{d}_j}}\right)\right] \quad (8.27)$$

$$f_{\tilde{d}_j}^{effective} : \left(\tilde{d}_j, l_{bound_{\tilde{d}_j}}, u_{bound_{\tilde{d}_j}}\right) = 0.7 \text{ for } \forall \tilde{d}_j \in \left[u_{bound_{\tilde{d}_j}}, u_{bound_{\tilde{d}_j}} + 0.1\left(u_{bound_{\tilde{d}_j}}\right)\right]$$

Outside these ranges the fidelity effectiveness functions returns zero indicating that such deviations are totally undesirable. About 10% deviation from the tolerances is expected to provide acceptable training. Furthermore, IAe's costumer doesn't have to officially certify its simulator according the FAA regulations so it is also allowed to somewhat lessen the fulfillment of the tolerances criteria from that perspective. Therefore, the fidelity effectiveness rating for a deviation with 10% from the FAA tolerances is set to 0.7 i.e. being more than sufficient.

Two subject matter experts were consulted to make the pare wise comparison of the importance of each fidelity evaluator function output and fidelity evaluator functions in the same hierarchical subsets (*Figure 8-9*). An external SME, referred as SME_1, was used to provide the team with unprejudiced relative importance weight vectors. A SME from IAe, referred as SME_2 was also invited to do the same as a reference and to identify possible important user needs. The pair-wise rating scale as suggested by Balci has been used for this purpose [2]. After the assignments the weighted averages the resulting weight vectors were normalized by dividing each vector element by its total length (*Expression 8.25*). To avoid bias as much as possible the resulting relative importance weights of both SME's were as follows:

$$(w_i)_{total} = 0.75(w_i)_{SME\_1} + 0.25(w_i)_{SME\_2} \tag{8.28}$$

As an explicit example consider the relative importance weights for the phugoid dynamics fidelity evaluation function outputs is given in the table below.

Metric or Method	Weights SME1	Weights SME 2	Total Weight
Period Difference	*0.4*	*0.5*	*0.425*
Time to Half	*0.4*	*0.3*	*0.375*
Abs Max Accuracy	*0.2*	*0.2*	*0.2*

Table 8-8 Phugoid Dynamics Relative Importance Weights Calculation

Calculation of Fidelity Effectiveness

Similar to the fidelity performance calculations the fidelity evaluation function hierarchy is used to calculate the fidelity effectiveness at the various local and global areas of interested using *expression 8.24*. Using the *Table 8-2*, *Table 8-8* and fidelity effectiveness function *8.27* the fidelity effectiveness for the phugoid dynamics becomes 1.00 indicating a valid representation of the phugoid motion. The short period fidelity effectiveness is far less and is 0.71, which indicates that the fidelity tolerances set for this fidelity evaluation are not fully attained but the level of fidelity is such that it deviations remain fairly close to the set tolerances. The total global fidelity effectiveness of this developed flight simulation model is calculated to be 0.81. So in general it can be conclude that level of fidelity is such that the developed flight simulation model is not valid with respect to the FAA level D requirements but the level of available fidelity is significant effective with respect to the intended application purpose of the flight simulation model. It is then finally up to IAe's costumer to decide on whether this acceptable. If not the fidelity effectiveness can be used, in cooperation with the costumer, to determine those fidelity deficiencies that require improvement This iterative accreditation process was still on its way at the time of writing this thesis.

Since the flight data-based test results of IAe's old flight simulation model weren't disclosed to the Delft team no comparisons could be made between the fidelity effectiveness of both flight simulation models. Otherwise it would have been possible to determine how much more effectively the level of fidelity of the new flight simulation model meets the set fidelity requirements. This could have been done globally but also locally to see where the maximum and minimum improvements in meeting the fidelity requirements have been achieved by the new flight simulation model.

8.6 Summary

This chapter developed four important simulation fidelity application concepts and associated techniques for the development and validation of simulation systems. All these four major concepts build upon and utilize the unified fidelity framework formalisms, referent and simulation system knowledge base, metrics and measurement methods developed in the previous chapters.

The first of these four application concepts is the notion of fidelity requirements. In this chapter fidelity requirements have been defined as the formal specification of the required level of fidelity from a simulation system necessary to meet the user needs and objectives. Therefore, the concept of fidelity requirements is thus the basis for the treatment of the returning question in the modeling and simulation of “how much fidelity is good enough”. Fidelity requirements as developed in this chapter consist of three building blocks: the required fidelity referent, the set of required fidelity evaluator functions that must be performed and most important the fidelity tolerance set. These fidelity tolerances specify the boundaries in which the measured level of fidelity must lie in order to completely fulfill the user needs. Based on these elements a pragmatic fidelity specification template has been developed and illustrated, which forms an integral part of the unified fidelity framework.

Fidelity requirements formed the basis for the second application concept developed in this chapter, which is the redefinition of in terms of required fidelity and measured available fidelity. Simulation validity has been formally defined here by the proposition that the measured level of fidelity should meet the required level of fidelity. If this proposition is true for a simulation system this system is said to be valid. It was then shown how this proposition could be translated into a coherent fidelity-based verification and validation process to enhance the modeling and simulation enterprise.

To support various simulation system comparisons, suitability assessment and other fidelity trade-off decisions during simulation system development, the concepts of fidelity performance and fidelity effectiveness have been introduced. The common known multi-criteria analysis method, the Analytical Hierarchy Process, serves as the basis for both these concepts. Fidelity performance provides an application independent and overall quantitative means for the initial judgment and comparison of the fidelity capabilities of simulation systems. On the other hand fidelity effectiveness gives an overall performance measure of how effectively the simulation system fidelity capabilities meets the application depended fidelity requirements. It was demonstrated that fidelity effectiveness evaluation could be used as a means for assessing simulation validity.

The multidimensional and multifaceted character of simulation fidelity implied that in practice fidelity performance and effectiveness evaluation of alternatives is a very complex and hard task to be handle by hand. Therefore, the availability and use of a general purpose or tailored multi-criteria analysis tool is indispensable for effective and efficient application of fidelity performance and effectiveness analysis. Most preferably such tools are fully compatible and interoperable with other automated fidelity tools for simulation development and validation.

In the next chapter it will be demonstrated how each of these developed application concepts can be integrated or utilized within the unified fidelity framework’s fidelity management process model. This fidelity management process model provides a series of practical guidelines to assist simulation developers in systematically managing all fidelity characterization, measurement and design decision activities that are occur in the development and validation of a simulation system.

9 Unified Fidelity Framework: Fidelity Management Process Model

9.1 Introduction

The previous chapters of this thesis presented the basic building blocks for the unified fidelity framework approach to simulation fidelity theory and practice. These chapters demonstrated that in real modeling and simulation practice fidelity assessment is a complex multi-dimensional and multi-facetted process. A process that involves a large amount of real-world and simulation system data, which has to be specified and analyzed with the use of a broad range of the measurement and evaluation methods of various kind and nature. Therefore, careful planning and management of such fidelity assessment activities in all stages simulation system development is essential for the successful assessment of simulation fidelity in today's complex and demanding simulation systems. What is needed is a systematic and well-structured process model to help guide users, developers and VV&A agents through the entire simulation system life-cycle, from requirement to the accreditation and operational usage by the consumer. This has also formally been defined in Section 4.3.2 by fidelity theorems 4, 5 and 11.

This chapter presents the unified fidelity framework fidelity management process model. The fidelity management process model integrates the fidelity concepts, formalisms and techniques developed in this thesis into a coherent and consistent fidelity assessment methodology. The term methodology is explicitly used here since the actual fidelity assessment process varies with the nature of each simulation application. Therefore, it is virtually impossible to provide a one-for-all or ready-made approach to assess simulation fidelity. A methodology provides a general basic framework with process steps for when and how to perform assessment activities within a particular discipline. However, the user of a methodology has the freedom to tailor this basic framework at some places, when necessary, to fully suite the particular needs of the application or problem at hand.

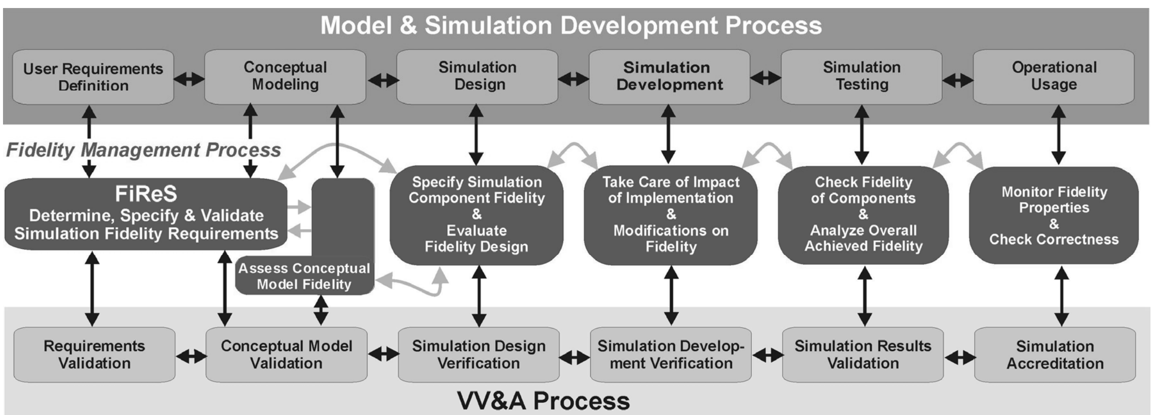


Figure 9-1 Fidelity Management Process Model Top-Level View

The fidelity management process model provides an organized, documented series of practical procedures and guidelines to assist simulation users, developers and VV&A agents in systematically applying and managing all those fidelity related requirements, characterizations, measurements and design decision activities that could occur in the development and validation of a simulation system. From a top-level perspective these process activities can be organized in six major stages. These stages can be mapped onto the generic simulation development and VV&A processes described in Section 3.3. For clarity and understandability this mapping is presented here and visualized (*Figure 9-1*) as an intermediated layer between both processes. However, in practice the fidelity management process model must be considered as an overlay whose activities thus smoothly integrate into either one of both existing processes thereby enhancing their effectiveness and efficiency in terms of development time, cost, etc. For similar reasons these six stages are presented here in a subsequent order. By no means this implies that the fidelity management activities form a water-fall process. In practice however, it is most likely they have to be traversed back and forward in an iterative fashion for correctional actions. Even some activities can and have to be performed concurrently for some periods of time depending on the specific application at hand. Next the six major stages of the fidelity management process are summarized:

- *Stage 1*: In this stage the simulation fidelity requirements are developed using the fidelity requirements specification method, abbreviated FiReS. FiReS provides an iterative set of activities for capturing the true user needs and objectives, and transforms these into a validated specification of fidelity requirements. These FiReS activities are all executed during the two first stages of the M&S development and VV&A process.

- *Stage 2*: The fidelity requirements developed in stage one are used to assess the conceptual model fidelity level and its compliance with these fidelity requirements. Secondly a feasibility study is conducted at this stage to see if the fidelity requirements are also achievable within the user constraints placed on the simulation development process.

- *Stage 3*: This stage allocates the fidelity capabilities specifications for each simulation system component in the system architectural design. This allocation is based upon the fidelity requirements that result from the FiReS stage. Furthermore, the complete simulation system design is evaluated to verify whether the resulting simulation system capabilities would fulfill the fidelity requirements. When necessary the design in terms of required system component and/or their fidelity capabilities have to be adjusted.

- *Stage 4*: The actual development and implementation of the design may reveal some unexpected difficulties of various kinds, which require modifications or impose practical limitations. Properly addressing and reporting the effects of these modifications and limitations on the final resulting simulation system fidelity are the prime activities of this fourth stage.

- *Stage 5*: In this stage the achieved level of fidelity of each component is checked against their fidelity capabilities specifications as soon as they come available. Finally, the overall achieved fidelity of the simulation execution results is checked here against the simulation fidelity requirements to determine validity of the simulation system. Based on concise reports of these and other fidelity assessment activity results, the simulation system can be accredited by the consumer for its application purpose.

- *Stage 6*: This most often overlooked stage in the model and simulation system life-cycle is the active offline and online monitoring of fidelity aspects during its operational usage. Depending on their degree of configurability a model or simulation system may be used such that it violates the operating boundaries for which the model or simulation system has been accredited. Another issue is that often during maintenance modifications are made to the original model or simulation system either unintentionally and deliberately for improvement. Obviously in both cases this induces substantial risk on the usage of the simulation execution results when the impact of these issues on the fidelity haven't been properly assessed.

During all stages of the fidelity management process any performed activity and its outcomes must be carefully documented and archived to make the process fully traceable and reproducible. This will enhance the quality of the process in case any correctional and follow-up actions are required, and will facilitate the reuse of the simulation system or its components for future application. Furthermore, it is of much help in convincing the consumer about the credibility and reliability of the simulation system.

Prior to stage one a VV&A lead team must be formed that initiates and supervises the whole fidelity management process. The role and responsibilities of each team member must be determined based upon each member's qualifications and expertise. Such knowledge should be specified in a similar fashion as described in section 5.3.3. For more objectivity and unbiased judgments it is recommended to have as many as possible independent VV&A lead team members that are at least not directly involved in the simulation system development process and at best not affiliated the division or company developing the simulation system.

In the remaining sections of this chapter each of these six top-level stages of the fidelity management process is decomposed in more detailed and concrete low-level fidelity assessment activities. The guidelines presented in here structure the way in which each activity outcomes are interrelated, used and transformed when transitioning from one (sub)stage to the next. As already noted earlier in this section the user may have to tailor the fidelity management process as appropriate. Therefore the objective of the following sections is to outline the most generic and comprehensive fidelity assessment framework as possible without pretending to be completely exhaustive in details or mandatory for all simulation systems. It is in the hands of the users to lever this framework or tool as the basic roadmap with which they can create and augment their own fidelity assessment route on a much more specific and lower detail level.

9.2 Stage 1: Simulation Fidelity Requirements Specification

The purpose of this stage is to identify and formally specify the simulation fidelity requirements, which must be met by the simulation system in order to fulfill the user needs. These requirements are the drivers for the simulation system design specification and the basis for constructing and executing the simulation validation plan. The method used for this purpose is referred here as the fidelity requirements specification process (FiReS). This FiReS method is an improved version of the original FiReS method that has been developed and presented in one of the author's earlier publications [121]. As discussed in this paper the FiReS forms an integral part of the fidelity management process model and its activities are closely related to the first two stages of the modeling and simulation enterprise (Section 3.3). *Figure 9-2* below illustrates the major FiReS activities. Although, these major activities are presented next in a sequential order, it should be noted that FiReS is an iterative and concurrent process like all other presented fidelity management process activities.

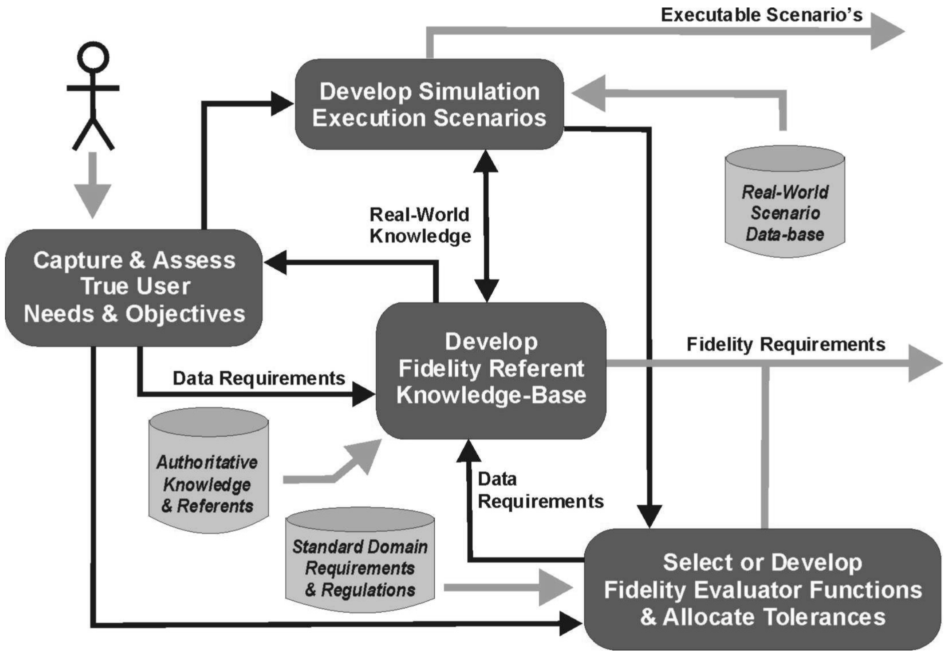


Figure 9-2 Fidelity Requirements Specification Stage Activities

9.2.1 Capture and Assess True User Needs & Objectives

The elicitation of true user needs is probably the hardest part of requirement engineering for several reasons: users have no clear idea of what they really need, users have difficulties to express needs, needs are stated in the user's language, needs may be missing or may contain non-essential wishes. Needs evolve from the user's desire to solve a problem in his current situation. A desire is what the user thinks, what is required to solve this problem. Therefore, amongst other things it also defines what he wants or expects from the real-world representation by the simulation system. A need statement is the formal expressions of the user's desires. This does not necessary imply that a user's desire is what he really needs i.e. is a true need. The only way to determine true needs is therefore posing the 'why' question to the user, which directly relates to the user's problem at hand and the simulation's role in attempting to solve this problem. A common understanding of the problem and application domain is thus essential to successfully accomplish this task.

What a simulation system is to establish, in order to satisfy the user's needs is referred here as simulation objectives. Objectives are a set of abstract functional requirements commonly agreed and understood by the simulation system developer and the user.

Usually, objectives are expressed in high-level and non-directly measurable terms. The purpose of next seven recommended tasks within this activity is to identify the true user needs and to develop the simulation objectives. These objectives provide an initial indication of the required level of simulation fidelity and are the start point for the other activities in this stage.

Task 1: Get the right people involved

Identify and select all people who could state relevant needs, objectives, fidelity and other requirements, as well as those whose contribution is required in the other activities of the fidelity management process. Next, specify all information regarding the contribution and responsibilities of each selected participant. A participant specification template similar to the one used for the fidelity referent can be used for this purpose (Section 5.3.3).

Task 2: Elicit needs and problems

Carefully prepare the elicitation process to obtain the needs and problems in well-defined and traceable manner. Next, start eliciting the needs and problems from acquiring global knowledge at a management level, using informal natural language techniques, and then gradually move to detailed knowledge about the needs and problems at the end-users or domain specialists level, using more formal techniques. Collect also any available information from the user's preliminary studies to gain essential simulation context and purpose knowledge.

Task 3: Analyze needs and problems

Analyze all needs and problems in cooperation with SMEs, to distil the core problem at hand and the true user needs. Cause-effect analysis can be used in this task as well as some heuristics [121]. Next, create a formal and prioritized list of true needs and problems to facilitate negotiation and trade off decisions. Finally, identify the problem and application domain as an entry for retrieving useful knowledge sources including standards, regulations, SME and fidelity referents.

Task 4: Negotiate the need and problem statement

Negotiate the in task 3 developed formal need and problem statement with the users in order to achieve a common understanding and agreement. Resolve any encountered issues in the negotiations by repeating tasks two and three.

Task 5: Elicit and Analyze Constraints

Elicit any additional information concerning constraints placed upon the development and usage of the simulation system, which hasn't been identified in the previous task. Analyze the identified constraints and explicitly specify their limitations or implications on the manner in which the required level of simulation fidelity has to be achieved and demonstrated. For instance a particular component or referent that has to be used or required third party data that isn't disclosed.

Task 6: Develop Hierarchy of Prioritized Simulation System Objectives

Decompose the in task four agreed need and problem statement into a hierarchy of more concrete and measurable objectives that must be accomplished to solve this statement (See *Figure 8-1* and *Figure 8-5*). These objectives, depending on the needs, can be statements of the required type and level of training that has to be provided, numerical outcomes their format and presentations such as MOPs, MOEs for design decisions making or acquisition of equipment etc. Apply the AHP of chapter 8 or other suitable method to rate the priority of each objective.

Task 7: Verify Objectives and Initiate Verification &Validation Test Plan

Check the objectives for their completeness and correctness with respect to the user needs. Next, conduct a feasibility study to determine if the objectives and associated coarse level of required fidelity are achievable within the constraints from task five. These initial fidelity requirements derive from the other FiReS activities that usually are triggered by the previous tasks. Use the prioritized list of needs and objectives to make trade-off decisions when necessary. Achieve a common agreement with the users and application SMEs on the hierarchy of objectives. Start the creation of the verification and validation test plan based on the outcomes from this activity and initial results from the other FiReS activities.

9.2.2 Develop Simulation Execution Scenarios

To achieve the set of specified simulation objectives it requires at least one and usually more simulation runs with different experimental or exercise set-ups that have to be executed (Section 3.2.3 and 8.2). Such a set-up describes a single instantiation of the represented real-world context i.e. a scenario describing the desired configuration of reality in terms of scope, initial conditions, script, time-line of events, etc. for that particular simulation execution. These simulation execution scenarios identify the major and coarse-grained structural, behavioral and functional aspects of the desired real-world from a situational and operational perspective. Therefore they are an essential element in the development of the formal fidelity requirement specification, the conceptual model and simulation configuration system design (Sections 3.2.3 and 6.4.1).

The purpose of next recommended tasks within this activity is the development of a specification of the minimal set or a range of scenarios that will be executed during the operation usage of the simulation system. These tasks are augmented by the real-world knowledge resulting from the fidelity referent development activities. The other way around the scenarios bound the part of reality that is of interest and thus provides directions and focus areas for the fidelity referent development. Similar, these simulation execution scenarios contribute to determining what fidelity evaluations have to be performed and settle associated tolerances levels.

Task 1: Develop a List of High-Level Textual Scenario Descriptions

Identify for each leaf in the objective hierarchy one or more scenarios in the form of an informal textual description. This description could state high-level real-world systems and their interactive behavior. When necessary the description is complemented with a list of sequential events. Next, compare and analyze the textual scenarios to check for completeness, redundancies and similarities.

Task 2: Determine the Desired Time Advancement Properties

Assess whether the overall simulation time advancement must be either real-time (hard or soft) or non-real-time (faster or slower). Provide any comments regarding this desired simulation time advancement approach including issues such as time-base synchronization and type (Section 3.4.3).

Task 3: Develop Top-Level Structural Composition of Real-World Systems
Refine the textual scenarios by identifying the needed key entities and other real-worlds along with their desired multiplicity for each scenario. Next determine their structural relationships and interactions over time. For clarity and negotiation a graphical representation in the form of a high-level real-world system topology map can be constructed (Section 5.4.1).

Task 4: Determine Essential Behavioral and Functional Properties
Describe for each of in task three identified top-level real-world systems the key behavioral and functional capabilities within the simulation application context. In here also any important state and parameter variables for the desired user data from the simulation execution must be addressed.

Task 5: Determine Time-Lines of External Events
Specify the various sets of pre-set and (time-) invariant external events and their occurring time(s), which may act upon the simulation while executing. When applicable any pre-conditions for an event to occur have to be expressed as well.

Task 6: Determine Initial and Termination Conditions
Determine the initial conditions that apply to each of in task three identified top-level real-world systems and their associated properties. Next, determine the conditions, based upon the state or outcomes of the simulation, for terminating the simulation execution i.e. when the simulation execution has produced the desired results to meet the associated objectives.

Task 7: Prioritize and Negotiate the Simulation Execution Scenarios
Apply the AHP of chapter 8 or other suitable method to rate the priority of each scenario and its content with respect to each parent objective. Negotiate the exaction scenarios to the users and application SMEs and achieve a common agreement regarding their proper fulfillment of the objectives.

9.2.3 Develop Fidelity Referent Knowledge Base

The purpose of this activity is to develop a suitable instantiation of the fidelity referent knowledge-base structure presented in chapter 5. Recall from that chapter that the exact real-world knowledge, which has to be entered into this referent structure, is application context dependent. Therefore, the recommended tasks for this activity specify guidelines for developing such a body of reference knowledge in a systematic and traceable manner, and not any particular domain knowledge. These fidelity referent development tasks are not only inextricably coupled and performed concurrently with the other FiReS activities but also with conceptual model development activities [126]. Although different in objective both activities have the same basis: elicitation of real-world knowledge. The next referent development tasks originate from an earlier publication on this topic [121].

Task 1: Get the right people involved
This task builds upon the results of task one from the first FiReS activity (Section 9.2.1) and assigns the fidelity referent developers and additional required validation team members. Start specifying all information for each involved participant, when

applicable, according the fidelity referent developer and validation information template structure (Section 5.3.3).

Task 2: Determine Referent Applicability and Coverage
Specify the application and problem domain, and develop an initial real-world coverage description using the fidelity referent applicability and status section. These are usually the direct result from the previous two FiReS activities. Search for existing suitable referents with similar real-world coverage description that can possibly be reused in cooperation with previously assigned domain SMEs.

Task 3: Analyze Available Referents for Reusability
Assess together with domain SMEs whether any of the in task 2 identified fidelity referents can be directly reused given the user needs and objectives. Possibly any of the previously identified problem or application domain constraints may even enforce the use of a standard authoritative referent (Section 9.2.1). When no suitable referent is found, it must be determined if any other existing referent can be adjusted (revision) or a completely new referent must be developed. Again this judgment is made in conjunction with domain SMEs.

Depending on the outcome of task three, either one of three different possible threads have to be followed (*Figure 9-3*). The most comprehensive of these three threads is the development of a new fidelity referent and its tasks are discussed next. At the end of this section the tasks of the other two are briefly presented.

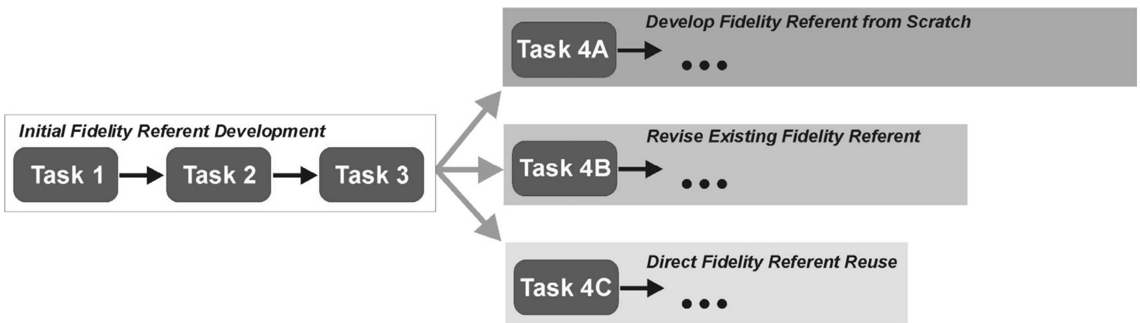


Figure 9-3 Fidelity Referent Development Threads

Task 4A: Specify the Management, Applicability and Status Information
Fill-out the identification and management information according the fidelity referent template (Section 5.3.1). Start refining the initial real-world coverage description with knowledge resulting from the next referent development tasks. Next, update the applicability and status information section by specifying the fidelity referent target and current authority level, and usage history for the current application (Section 5.3.2). Start tracking and documenting any important revisions of the referent contents in the management section.

Task 5A: Elicitation of Real-World Knowledge Sources
Form a group of application and problem domain SMEs, who will help in the elicitation, analysis and accreditation of the real-world knowledge. Next, the initial knowledge and source databases are elicited in co-operation with these SMEs. All found relevant knowledge sources are documented in the referent utilized knowledge and data source

section. These sources may also include other SMEs with additional knowledge about the real-world problem domain.

Task 6A: Specify the Real-World Structural Properties & Relationships
Compile from the previously elicited set of knowledge sources an enumeration of required real-world systems and structure them in a system topology map together with their interactions. Start point is the high-level real-world system topology that results from the simulation execution scenario development (Section 9.2.2). Continue to refine the real-world system topology map and assign the structural properties on the basis of the real-world system characteristics description template (Section 5.4.1). Explicitly state in the overall structural property section any boundary condition, assumptions or simplifications made.

Task 7A: Specify the Real-World Behavioral Data
Elicit and specify the required time-based and parametric behavioral data for each real-world (sub)system identified by the previous task. This elicitation and specification of behavioral knowledge is performed on the basis of the templates of the fidelity referent real-world behavioral data section (Section 5.4.2). Starting with high-level and qualitative behavior specification and gradually moving to behavior sample specifications. Depending on the user objectives and required fidelity evaluations (Section 9.2.4) determine whether behavioral data is available and appropriate. If not it should be decided how better data can be obtained through preparation and execution of measurements or constructed from other possible sources. Such tasks must be documented in the elicitation activities and constrains section of the referent knowledge-base.

Task 8A: Real-World Knowledge Errors, Uncertainties and Verification
Identify any possible and not yet discovered real-world knowledge error and uncertainty sources. Try to assess and specify their nature, magnitude and impact on the usability and reliability of the real-world reference knowledge using the referent error and uncertainty template structure (Section 5.3.4). Next, have a group of SMEs to formally verify that the fidelity referent is a correct and sufficient complete representation of the real-world as encompassed by the problem and application domain in relationship to the targeted authority status and current simulation objectives. For an objective and independent judgment the SME's are best selected from a pool of experts, which haven't been involved in the other referent development tasks.

Fidelity Referent Revision and Direct Reuse Tasks
In case of revising an existing fidelity referent knowledge-base scenario, basically the same tasks as discussed above are performed to make the required adjustments and accredit the revised referent. The major difference is that time and other resources spend in this process are in general far less. Another important task is that the referent revision history and status must be updated accordingly. When the referent is reused as-is the only remaining task to be done is to add the current simulation application to the referent's known usage list.

9.2.4 Select Fidelity Evaluator Functions and Allocate Tolerances

The purpose of this activity is the development of the minimal set of required fidelity evaluator functions and the associated tolerances for their outcomes. The next recommended iterative tasks for this activity build upon completing the fidelity requirements specification template as developed in Section 8.2.

Task 1: Identify or Develop Fidelity Evaluator Functions

Start eliciting applicable fidelity evaluator functions for each simulation execution scenario in relationship to its associated parent objectives (Section 9.2.2). Use the in chapter 7 developed categories and generic fidelity evaluator functions, and any standard application and problem domain evaluator functions as the basis. Develop additional fidelity evaluator functions when necessary. Complete and fill out the evaluator function details according to the fidelity requirements specification template developed in section 8.2.3.

Task 2: Analyze and Develop Minimal Fidelity Evaluator Function Set

Compare the found evaluator functions amongst each other for completeness and redundancies. Try to structure the resulting set of evaluator functions according possible application specific grouping criteria into a hierarchically ordered fidelity evaluator subsets as outlined in Section 8.4.2. Fidelity performance metrics for each of these subsets can be assigned accordingly.

Task 3: Assign Tolerances to each Fidelity Evaluator Function

Search for any standard or similar application and problem domain tolerance levels for an initial start point of this task. Use these results to start assigning tolerances to all fidelity evaluator function outputs, which for each simulation scenario objectives and their importance will produce reliable and useful simulation results. Document these tolerances according the in section 8.2.3 developed fidelity requirements specification template. Apply the fidelity effectiveness method to determine the relative importance of the evaluator function outputs and rate the effects on the simulation execution objectives (risks, reliability, etc) when tolerances are not met (Section 8.4.3).

Task 4: Verify the Minimal Fidelity Evaluator Function Set and Tolerances

Have an independent group of application and problem domain SMEs verify that the developed minimal fidelity evaluator function sufficiently covers the important real-world aspects of interest. Next, have them verify that the simulation system will indeed produce reliable and useful results if the evaluated real-world aspects reside within in the allocated tolerances. Check whether the fidelity referent contains all data necessary to execute the minimal fidelity evaluator function set.

Task 5: Complete Simulation System Verification & Validation Test Plan

The minimal set of fidelity evaluator functions and associated fidelity performance and effectiveness metrics spans a validation test matrix. Therefore, use this matrix to augment and complete the simulation system verification validation test plan by specifying when and where each fidelity evaluator function must be executed and its results be analyzed.

9.3 Stage 2: Assess Conceptual Model Fidelity

The purpose of this stage is to assess the simulation model concept fidelity level and its capabilities to meet all simulation fidelity requirements developed in the previous fidelity management process stage. This simulation model concept is the product of the conceptual modeling stage of simulation development process and servers as the blueprint for the design and development of the actual simulation model (Section 3.3.1). Also a feasibility study is conducted at this stage to see if the prospective fidelity level of the simulation model concept can practically be realized within the constraints placed on the simulation development process.

In *Figure 9-4* at the next page the major activities for this fidelity management process stage are illustrated. Although, these activities are presented in a sequential order, it should be noted that this is not mandatory. Like all other fidelity management process activities they could be traversed iteratively and concurrently when necessary.

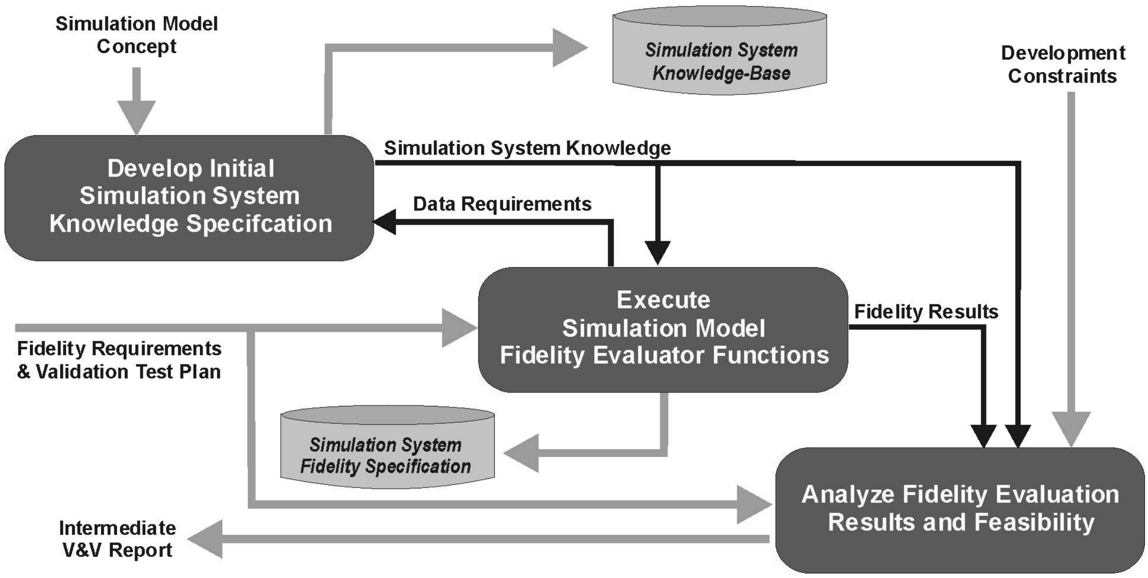


Figure 9-4 Conceptual Model Fidelity Assessment Stage Activities

9.3.1 Develop Initial Simulation System Knowledge Specification

The purpose of this activity is to elicit knowledge from the simulation model concept and specify this knowledge in the simulation system knowledge-base. This simulation model knowledge-base merges all real-world representational related knowledge, which is otherwise often scattered over the simulation conceptual model, design and development documents, into a single formal and consistent specification (Chapter 6). Therefore, the simulation system knowledge development is initiated within the conceptual model stage and is continuously updated throughout the respective stages of the simulation development process. The next three recommended and iterative tasks for this activity build upon and populate the simulation model knowledge structure as developed in Section 6.3.

Task 1: Develop Initial Meta-Model Description

Identify the model component, their types and interactions that constitute the simulation model concept. Create a meta-model topology map representation of these simulation model components and their interaction relationship. Next, specify the both the overall and detailed characteristics of simulation model concept model components using the meta-model characteristics description template (Section 6.3.1).

Task 2: Develop Initial Real-World System Realization Description
Identify the set of real-world system models and interactions, which are encapsulated by each meta-model component. Merge all these sets into a single overall real-world system model topology map. In here explicitly address and specify any manifold real-world system model representations. Finally, specify for each real-world system model all yet available information regarding their internal characteristics and realization. Utilize the real-world system model characteristics and realization templates for this specification (Section 6.3.2).

Task 3: Assess and Specify Simulation Model Uncertainty & Error Sources
Trace the simulation model concept for any evident structural, data and estimation precision error and uncertainty sources (Section 6.3.3). Assess the impact of these sources on the eventual achievable simulation fidelity and when possible try use sensitivity analysis for this purpose. Finally, document the qualitative and possibly quantitative results according the simulation model error and uncertainty specification template.

9.3.2 Execute Simulation Model Fidelity Evaluator Functions

The purpose of this activity is to qualify and to quantify the level of fidelity of the simulation model concept. Since the simulation system is not fully physically present at this stage only a static or non-runtime fidelity evaluations can be executed. This activity is performed following the next recommended tasks.

Task 1: Prepare Simulation Model Concept Fidelity Evaluation
Access the validation test plan to obtain the list of assigned fidelity evaluation for this development stage. Obtain the fidelity evaluator function descriptions and referent from the fidelity requirements specification. Check the simulation system knowledge specification for any missing and incompatibilities with the referent.

Task 2: Execute Applicable Fidelity Evaluations
Start with measuring the differences in structural properties between the referent and the simulation system knowledge specification. Next, evaluate the behavioral fidelity of the simulation model concept so far as possible. This involves comparative evaluations in the area of interaction causality, non-causal and qualitative behavioral knowledge or SME based evaluations (Chapter 7).

Task 3: Pre-Process Evaluation Results
Carefully specify and store the executed fidelity evaluator functions plus their results for further analysis and future reference into a so-called simulation system fidelity specification. Such a specification is structured according the formal definition of practical fidelity, *Expression 4.23*, developed in section 4.4.2. Explicitly address in here any encountered problems during this activity.

9.3.3 Analyze Fidelity Evaluation Results and Feasibility

The purpose of this activity is to assess if the simulation model concept has such fidelity capabilities that when implemented as an operational simulation model will result in the required level of fidelity. It is also checked whether such implementation will practically be feasible. This assessment is performed according the next recommended tasks.

Task 1: Check Fidelity Evaluation Results against Fidelity Requirements
Determine how well the fidelity evaluation results remain within their associated prefixed fidelity tolerances. When possible determine the fidelity performance and effectiveness at various levels (Sections 8.4.2 and 8.4.3). Verify whether all required simulation execution scenarios could be configured with this model (Section 9.2.2).

Task 2: Review the Simulation Model Concept and Assess Feasibility
Have group of SMEs determine that the specified theories, assumptions, logic, mathematical and causal relationships underlying simulation model concept are correct and suitable for the intended simulation objectives (Section 9.2.1). Explicitly state any sources that could result in unacceptable fidelity levels in the behavior replications during simulation execution. Next assess if the simulation model concept and fidelity requirements are practically feasible within the user constraints placed on the simulation development process.

Task 3: Produce an Initial Verification and Validation Report
Report any discovered compromising simulation model concept issues and decisions on eventual achievable fidelity during simulation execution. When necessary, report any recommendations to improve the simulation model concept. Negotiate this initial verification and validation report with both the user/sponsor and the simulation development team in order to take any necessary actions.

9.4 Stage 3: Specify and Assess Simulation Design Fidelity

The purpose of this stage is to develop fidelity capabilities specification for each component in the system simulation architectural design and to verify that total simulation system capabilities will be able to fulfill the fidelity requirements.

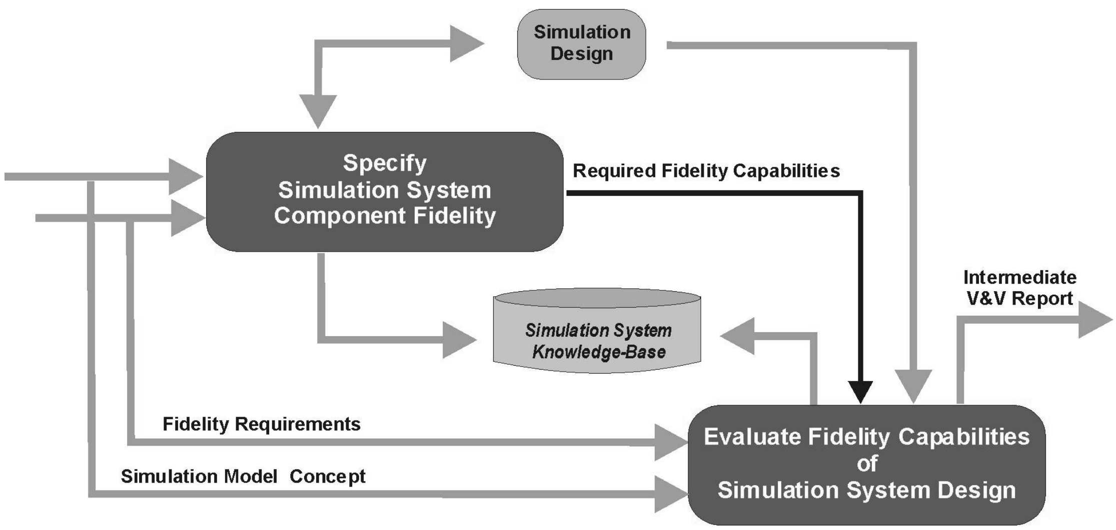


Figure 9-5 Simulation System Design Fidelity Stage Activities

When necessary the simulation system design in terms of required system components and/or their fidelity capabilities may have to be adjusted as a result of the activity outcomes of this stage. As illustrated in *Figure 9-5* the two major activities for this fidelity management process stage. Although, these activities are presented next in a sequential order, it should be noted that this is not mandatory but could also be traversed iteratively and concurrently if required.

9.4.1 Specify Simulation System Component Fidelity

The purpose of this activity is to translate the simulation model concept and fidelity requirements into detailed required simulation component fidelity capabilities specifications that can guide the simulation system development. To realize this activity purpose the next tasks are recommended.

Task 1: Develop Detailed Component Fidelity Design Requirement
Further refine the meta-model developed in the previous activity with more required low-level implementation oriented subcomponents. Develop for each model component a specification for the required physical, functional and performance capabilities of its implementing hard and software elements to be able to properly meet the fidelity requirements during simulation execution. These implementation-oriented fidelity capabilities include amongst many other things computer capabilities, network latency, visual and motions system hardware performance, programming language and code performance.

Task 2: Review Fidelity Specification of Existing Components
Based upon the results of task one and the specified constraints place upon the simulation development (Section 9.2) search for reusable existing simulation system components. Analyze the fidelity capabilities of these candidates to determine if they can be reused directly, or modifications have to be made or completely new simulation system components have to be designed.

9.4.2 Evaluate Fidelity Capabilities of Simulation System Design

The purpose of this activity is to evaluate the complete resulting simulation system architectural design for its compliance with the fidelity requirements, simulation model concept and the component fidelity capabilities specified in the previous activity. For this activity the next tasks are recommended.

Task 1: Verify Simulation Model Design & Component Fidelity Capabilities
Verify that the component fidelity capabilities are correctly reflected in the simulation system architectural design and that no components are omitted or added. Elicit, specify and update the real-world system realization description in the simulation system knowledge base. Verify that the simulation model concept has correctly been translated into the simulation system architectural design.

Task 2: Estimate and Verify Fidelity Performance and Effectiveness
Use the simulation fidelity requirement set to execute any possible fidelity evaluator functions or otherwise try to obtain best estimates for its outcomes given the current design. Next, apply simulation fidelity performance and effectiveness evaluation

methods to make best estimate for the eventual overall fidelity performance and effectiveness (Sections 8.4). Based on these evaluations identify any bottlenecks or areas of concern in the current simulation system architectural design, which limit the achievability of the required level of fidelity.

Task 3: Produce an Initial Verification and Validation Report

Report any discovered compromising simulation system design issues and decisions on eventual achievable fidelity during simulation execution. When necessary, report any recommendations to improve the simulation system design. Negotiate this initial verification and validation report with both the user/sponsor and the simulation development team in order to take any necessary actions.

9.5 Stage 4: Asses Impact of Implementation Issues on Fidelity

The purpose of this stage is to track the actual development or implementation of the simulation system design in order to detect possible differences and modifications in relationship to this design and assess their impact on the achievable level of fidelity. Usually, such differences and modifications arise from unforeseen difficulties and limitations of various kinds in the design as well as in the development process and its allocated resources, which have to be resolved in the simulation system development stage. *Figure 9-6* below illustrates the two major activities for this fidelity management process stage. Although, these activities are presented next in a sequential order they are always traversed iteratively and concurrently in practice.

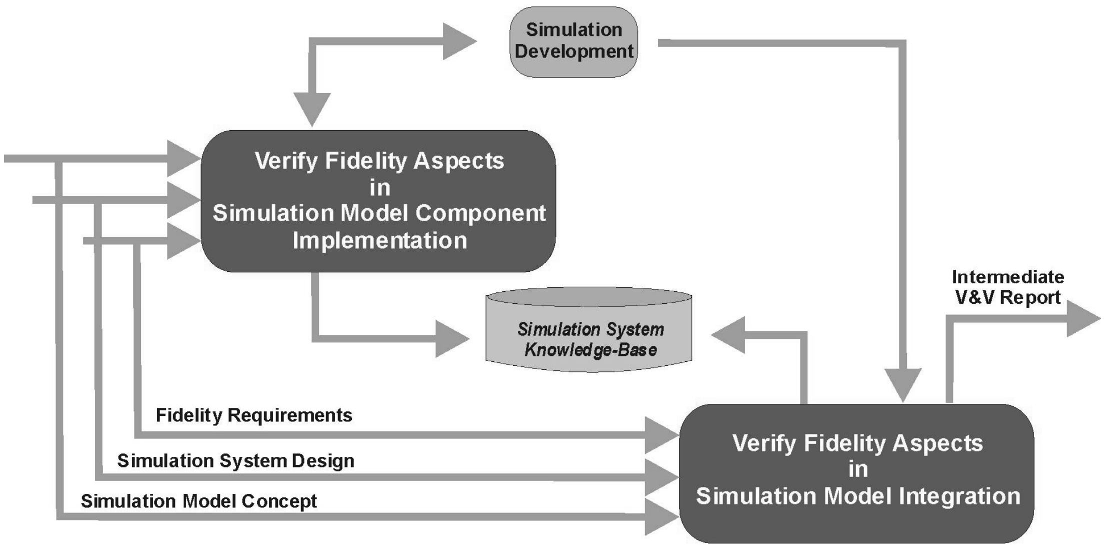


Figure 9-6 Simulation System Development Fidelity Stage Activities

9.5.1 Verify Fidelity Aspects in Simulation Model Component Development

The purpose of this activity is to verify the implementation of the each designed and individual simulation model component. For this activity the following two tasks are recommended.

Task 1: Monitor Each Simulation Model Component Implementation

Trace the complete implementation of each simulation model component for any internal modifications and extensions with respect to the original component design. In this task also any found inconsistencies in terms of simulation model concept representation and the required fidelity capabilities should be specified in the simulation system knowledge-base.

Task 2: Assess Fidelity Impacts of Differences with the Original Design
Properly address and report the effects of the in task one discovered extensions, modifications, and inconsistencies on the achievability of the final required simulation system fidelity. Use for this purpose validation SMEs, fidelity-based multi-criteria analysis or any other suitable method. Furthermore, assess the reasonableness and risks of any fidelity compromising component implementation on the simulation system usage.

9.5.2 Verify Fidelity Aspects in Simulation Model Integration

The purpose of this activity is to verify the integration of the each developed simulation model component into the complete simulation model, which underlies the simulation system. The two tasks recommended for this activity are almost similar to ones of the previous activity except these task now focus on fidelity aspects and effects of compromising implementation decisions made during the integration of the simulation model components into a completer simulation model. Therefore, these tasks are not mentioned here again. All elicited information about the simulation model implementation must be used to update the simulation system knowledge-base. Furthermore, negotiate all findings by means of a verification and validation report with both the user/sponsor and the simulation development team in order to take any necessary actions.

9.6 Stage 5: Achieved Simulation System Fidelity Checking

The purpose of this stage is to determine the level of simulation fidelity of the developed simulation system and to check how well this achieved fidelity resides within the fidelity tolerances developed in stage one. Which, when and how a fidelity evaluation method has to be used for quantifying or qualifying the level of simulation fidelity is prescribed by both the fidelity requirements set and the associated validation test plan (Section 9.2).

However, for separation of concern and efficiently resolving any sources of unacceptable deviations these fidelity evaluations are performed bottom-up starting from the lowest level simulation system component up to the total integrated simulation system level. Based on these outcomes the validity of the simulation system can be established (Section 8.3). Together with all previous fidelity management process results, the outcomes of this stage are the basis for gaining the simulation system accepted by the consumer for its application purpose.

Figure 9-7 at the next page illustrates the major activities for this fidelity management process stage. Although these activities are presented in the next sections in a sequential order this is not mandatory. In practice these activities are mostly executed concurrently and traversed iteratively.

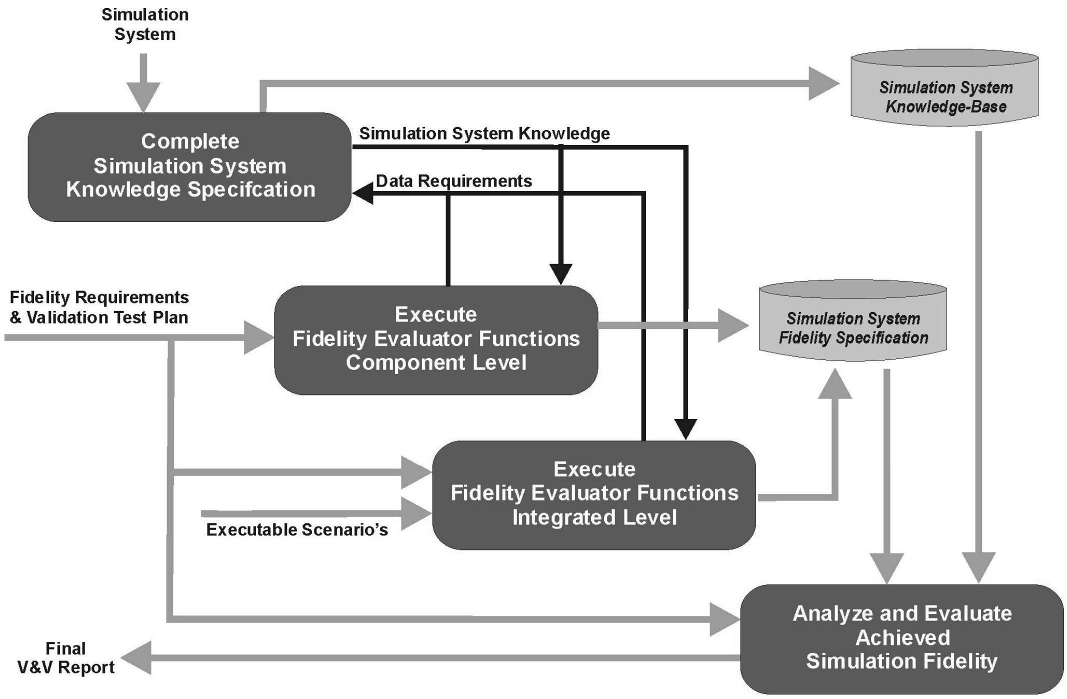


Figure 9-7 Check Achieved Simulation System Fidelity Stage Activities

9.6.1 Complete the Simulation System Knowledge Specification

The purpose of this activity is to elicit knowledge from the final simulation model and specify this knowledge in the simulation system knowledge-base. Except for being far more extensive four of the next five recommended tasks for this activity are in essence similar to the ones of activity 9.3.1 in stage two.

Task 1: Complete Existing Meta-Model Description

Verify the completeness and consistency of model components, their types and interactions in the current meta-model description with the final simulation model. Make any necessary adjustments to the meta-model topology, the overall and detailed meta-model characteristics description (Section 6.3.1).

Task 2: Complete Existing Real-World System Realization Description

Verify the completeness and consistency of real-world system models, their interactions and characteristics in the real-world system realization description with the final simulation model. Make any necessary adjustments to this real-world system realization description (Section 6.3.2).

Task 3: Assess and Specify Simulation Model Uncertainty & Error Sources

Once again trace the final simulation model concept for any evident structural, data and estimation precision error and uncertainty sources (Section 6.3.3). Since the simulation model is available in this stage for testing more precise and quantitative analysis can be conducted here. Complete the simulation model error and uncertainty specification description with new and additional findings.

Task 4: Develop Required Simulation Execution Knowledge
Identify from the simulation fidelity requirements, validation test plan and the simulation execution scenarios, which run-time fidelity evaluations have to be executed and what data is required (Section 9.6.2 and 9.6.3). Execute these desired run-time tests one after one another and log the observed replicated real-world behavior of interest. Pre-process and specify the observed behavioral data in terms of non-causal, interaction causality, qualitative and behavior sample evaluation. All in here elicited knowledge is specified according the real-world replication knowledge template developed in section 6.4.2. Furthermore, for each execution specify the simulation system configuration setting (Section 6.4.1) and the structural properties of the real-world system model instance.

Task 5: Specify Complementary Knowledge
Complete the simulation system knowledge-base complementary knowledge section with not yet specified information regarding the used simulation support systems and the knowledge elicitation processes (Section 6.5). Furthermore, update the management section for the made revisions in this activity.

9.6.2 Execute Fidelity Evaluator Functions at Component Level

The purpose of this activity is to qualify and quantify the achieved level of fidelity of each component in the final simulation model. This stand-alone evaluation of fidelity can be executed according the validation test plan as soon as a simulation model comes available from the simulation model development process. This activity has the similar recommended tasks as in activity 9.3.2 of stage two.

Task 1: Prepare Simulation Model Component Fidelity Evaluation
Access the validation test plan to create a list of applicable fidelity evaluations for each simulation model component. Obtain the fidelity referent and evaluator function descriptions from the fidelity requirements specification. Check and update the simulation system knowledge specification for any missing required knowledge or incompatibilities with the fidelity referent (Section 9.6.1).

Task 2: Execute Applicable Fidelity Evaluations
Qualify and quantify the differences in both structural and behavioral properties between the referent and the specified simulation model component knowledge according the specific type of fidelity evaluator function description (Chapter 7). This task is executed for each previously identified simulation model component.

Task 3: Pre-Process Evaluation Results
Update the formal simulation system fidelity specification with the results of the executed fidelity evaluator functions for each simulation model component. See also section 9.3.2.

9.6.3 Execute Fidelity Evaluator Functions at Integrated Level

The purpose of this activity is to qualify and quantify the total achieved level of fidelity of the complete integrated simulation model. This fidelity evaluation can be executed according the validation test plan once the simulation model integration has been completed. This activity has exactly the same recommended tasks as in the previous activity but know applying to the complete simulation model instead. Therefore, these tasks are not repeated here.

9.6.4 Analyze and Evaluate Achieved Simulation System Fidelity

The purpose of this activity is to assess how well the achieved level of fidelity of the simulation system and its underlying simulation model meets the specified fidelity requirements developed in the first stage of this fidelity management process. Furthermore, in this activity the outcomes from all previous stages are combined and utilized into a coherent and traceable body of fidelity evidence to support a sound judgment about the suitability of the simulation system for the user needs and objectives i.e. validity. To achieve these purposes the next four tasks are recommended for this activity.

Task 1: Check Achieved Fidelity against Fidelity Requirements
Once the fidelity evaluation results of each simulation component comes available start checking how well these results remain within the allocated fidelity tolerances and fidelity capability specification (Section 9.2.4 and 9.4.1). Resolve in cooperation with the developers any found local fidelity problems, which significantly affect other simulation component outcomes and thus must be immediately fixed prior to the integrated fidelity assessment. Next, repeat these subtasks but now for at the complete integrated simulation model level. Further, verify whether all required simulation execution scenarios could be configured properly with the current simulation system.

Task 2: Assess Simulation System Fidelity Performance & Effectiveness
Perform simulation fidelity performance and effectiveness analysis on the final integrated simulation model using the separate fidelity evaluation results from the previous two activities (Sections 8.4.2 and 8.4.3). In here evaluated the scores at various levels and areas of interest to identify the major causes for any possible uncovered fidelity issue.

Task 3: Evaluate the Final Simulation System and Model
Combine all outcomes from this and the previous fidelity management process into coherent and traceable body of fidelity evidence for the developed simulation system and model. Next, evaluate these outcomes in from the user needs and objectives perspective to determine that there is sufficient and convincing fidelity evidence to prove, with reasonable uncertainty and risk, that the simulation system provides a sufficient level of fidelity for the user application purpose. Employ well-qualified validation SMEs, sensitivity analysis, and error and uncertainty assessment methods to support making reliable judgments (Appendix B). This is of most importance for predictive simulation executions far outside the practicable testable range of fidelity. In case major fidelity flaws are detected, the fidelity performance and effectiveness analysis method can be used in relationship with the user constraints (cost, schedule, available resource, etc) to decide on the best compromising corrective actions.

Task 4: Produce an Final Verification and Validation Report
Complete the verification and validation report with the results from this activity. This includes not only the formal simulation fidelity specification and its evaluation conclusions but also a complete process description of the performed fidelity

management activities. The inclusion of a process description serves two purposes. First, the complete process and results become fully traceable and repeatable, which facilitates both simulation system modifications and reuse. Second, it enables process evaluation to determine the thoroughness, quality, effectiveness and efficiency of the fidelity management process. This is not only important for future application of the process but also for the consumer in gaining confidence in the fidelity management process results. Negotiate this final verification and validation report with the consumer or user for acceptance in the simulation system accreditation activities (Section 3.3.2).

9.7 Stage 6: Simulation System Operational Fidelity Monitoring

The purpose of this last stage in the fidelity management process is to actively monitor and check fidelity properties during the operational usage of the simulation system. This activity is intended to detect and address possible violations of the operational range for which the simulation system has been validated and accepted, and fidelity issues due to system modifications. *Figure 9-8* at the next page illustrates the major activities for this last stage.

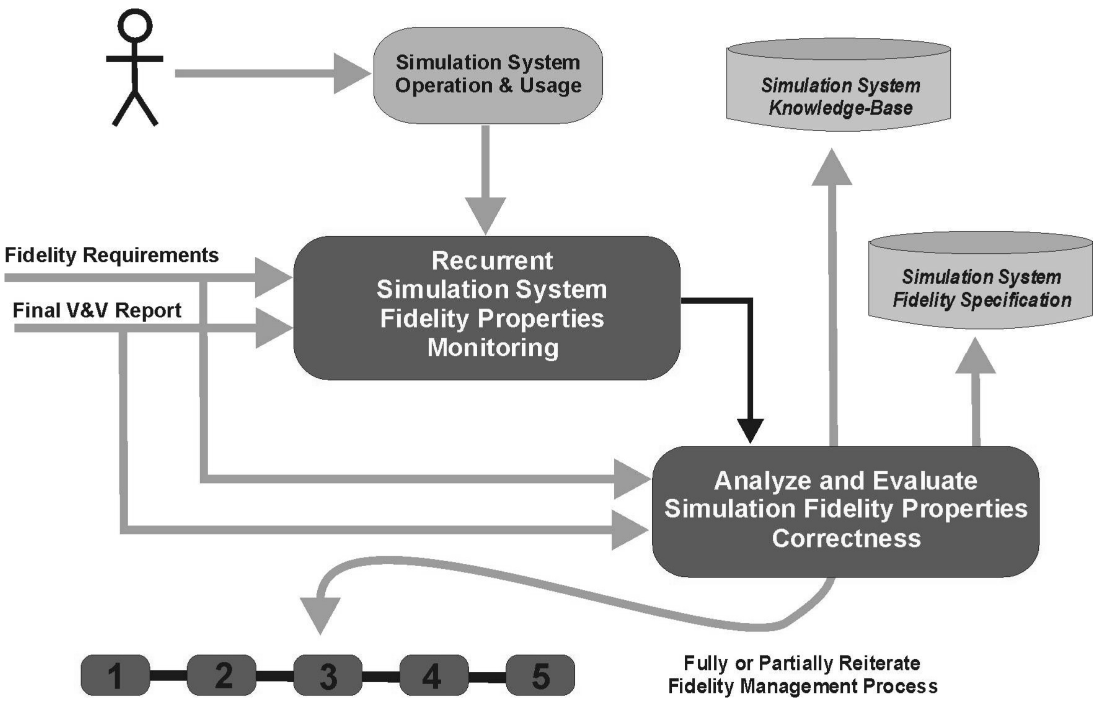


Figure 9-8 Simulation System Operational Fidelity Monitoring Stage Activities

9.7.1 Recurrent Simulation System Fidelity Properties Monitoring

The purpose of this activity is to monitor any fidelity properties that influence the level of fidelity during simulation execution. These fidelity properties are obtained from both the fidelity requirements and the verification and validation report. Careful planning, preparing and allocating resources throughout the simulation life cycle are mandatory pre-conditions for an effective and efficient execution of this activity. The following other two recommended tasks for this activity are mostly executed concurrently.

Task 1: Monitor Fidelity Properties On-line
Monitor the prefixed fidelity properties during all simulation executions conducted with the simulation system. Apply all kinds of fidelity monitoring facilities such as data-loggers, stealth-viewers, etc. for this task, which already may have been used in the previous stage.

Task 2: Monitor Fidelity Properties Off-line
Monitor for any modifications, either unintentionally or deliberately, made to the original simulation system, its underlying simulation model. Such modifications can be the results of maintenance, hard or software updates, used parameter databases, etc.

9.7.2 Analyze & Evaluate Simulation Fidelity Properties Correctness

The purpose of this activity is to analyze and evaluate the results from the previous activity to identify and resolve possible fidelity issues that may induce substantial risk on the usage and validity of the simulation execution results. Potential tasks for this activity dependent on the issues found during recurrent simulation system fidelity monitoring. These issues can be diverse and therefore it is hard to give a set of concrete tasks and follow-up actions. Such follow-up actions imply always a partial or even full reiteration of the fidelity management process. The profundity of the reiteration depends on the severity of the found fidelity issues. The tasks and actions in this activity can be divided into the next four common fidelity issue areas.

Previously Not Identified Fidelity Issues
During the first extensive usage period of the simulation system it is not quite uncommon that the user encounters fidelity problems that weren't discovered previously. Usually, these are just due to design and implementation errors on behalf of the developer, which have to be fixed and tested again to check the fidelity improvement. However, a more severe issue could be that some fidelity requirement has been misinterpreted or slipped through the fidelity requirements specification.

Wrong Usage of the Simulation System
In the operational usage of a simulation system it is possible that the system is used beyond the operating boundaries for which it has been accredited. Causes for this could be wrongly configuring the simulation system or lack of knowledge about the validity. The user can be guarded against such wrong usage by proper training and instruction of the simulation system operational capabilities, limitations and risks. Another option is to build in some safety measures to avoid or notify the users of such fidelity issues.

Effects of Simulation System Modifications and Updates
Even though it is obvious that once a simulation system is modified or updated the simulation fidelity could change significantly, it is often considered as a bit superfluous to reassess the level of fidelity after such modifications. This could result in serious risks in using the simulation results since their credibility hasn't been demonstrated. It is therefore mandatory to carefully reiterate the fidelity management process to effectively address fidelity issues that rise from such modification and updates.

Updates the Fidelity Management Knowledge-Bases
Simulation operation and execution may result in new fidelity information of the system itself, which can be used to revise the current fidelity specification and other fidelity related knowledge bases. Furthermore, new real-world reference knowledge may come

available that can be used to improve the accuracy and completeness of the current simulation system fidelity specification.

9.8 Summary

This chapter presented the fidelity management process model, which is an integral part of the unified fidelity framework. It integrates all previous developed unified fidelity concepts, formalisms and techniques into a single coherent and consistent fidelity assessment methodology for the modeling and simulation enterprise. The fidelity management process model provides a uniform and systematic approach to assist users, developers and VV&A agents in applying and managing all those fidelity issues and assessment activities that occur in the entire simulation system life-cycle, from requirements to the operational usage by the consumer. From a top-level perspective the fidelity management process model is organized in the following six major stages:

1. *Fidelity Requirements Specification*
2. *Assess Conceptual Model Fidelity*
3. *Specify and Assess Simulation Design Fidelity*
4. *Asses Impact of Implementation Issues on Fidelity*
5. *Achieved Simulation System Fidelity Checking*
6. *Simulation System Operational Fidelity Monitoring*

Together, these six stages provide a well-organized series of traceable fidelity assessment activities and tasks, which smoothly integrate into either the simulation system development or VV&A process. This not only facilitates the repeatability but also enhance the effectiveness and efficiency of the fidelity management process. Depending on the specific application at hand these six stages and their underlying activities and tasks can and have to be traversed in an iterative and concurrent fashion.

It must be clearly noted that the fidelity management process model is a methodology, which like a cookbook gives a recipe or guidelines for when and how to perform certain fidelity assessment activities. Similar to a cook the user of the fidelity management process model has the freedom to tailor and extend the in here recommended activities and tasks to fully suite the particular needs of their organization or consumer's simulation application at hand. For this reason it's most likely that the in here presented general fidelity management process will serve as a common basis for the development of more detailed and specifically application or problem domain tailored versions. This was also the for both aerospace simulation case studies. In here the general fidelity management process as outlined is this chapter but their existed differences in certain areas due to the fact that the CN235 simulator had to comply with the standard FAA flight simulator requirement document and the FASE simulator required no application of such standard.

10 Conclusions and Recommendations

10.1 Discussion of the Results and Lessons-Learned

Simulation fidelity is an intrinsic element of any simulation system, one that all simulation developers and users have to deal with one way or the other. It is commonly recognized by the modeling and simulation community that simulation fidelity is an essential vehicle in properly assessing the validity and credibility of simulation results. Furthermore, fidelity is found to be one of the main cost-drivers of any model or simulation development. Despite these observations and the enormous advancements in simulation hardware and software, this thesis showed that the ability to characterize, qualify and quantify the level of simulation fidelity is still a largely uncultivated area with many incomplete, inconsistent and widely scattered views, concepts and approaches to simulation fidelity. What is primarily lacking is the absence of a systematic and general applicable fidelity assessment methodology based on a sound unifying theory for simulation fidelity and associated practices.

This thesis tried to fill this gap by the development of the unified fidelity framework. A framework that expands and integrates existing simulation fidelity approaches into a single unified fidelity theory and practice. This unified fidelity framework has been developed from a general simulation system life cycle context, not limited by any specific application or problem domain aspects. As a result the unified fidelity framework provides a common basic methodology that underlies any simulation fidelity assessment process, which can be tailored and extended to suite the particular needs of the simulation system application at hand if this is necessary. The general benefits of working from this single framework are improvements in understandability, effectiveness, repeatability and reusability of any simulation fidelity assessment process and its outcomes. These benefits are not only confined to the user’s own problem or application domain but more importantly it is believed they can transcend beyond such boundaries. Particularly, when simulation fidelity assessment and simulation validation in general become an essential part of the education of those involved in the modeling and simulation enterprise. This will further open the door, which has been set ajar by this thesis, to move to a standard and modeling and simulation community wide adopted fidelity theory and practice.

10.1.1 Simulation Fidelity Definition

The literature study presented in this thesis showed that there exists a wide variety of often contradicting definitions and connotations for fidelity and related terms. This has caused great confusion in the past. Addressing this issue is a precondition in understanding and properly solving fidelity issues. Therefore a common fidelity related terminology has been adopted here that has been developed in cooperation with the Simulation Interoperability Standards Organization community (SISO).

It has been shown in Chapter 4 that the conceptually best definition for fidelity is of an esoteric nature, and can never be practically implemented and realized within the modeling and simulation enterprise. Based upon the arguments underlying this important conclusion the fundamental concepts for a pragmatic simulation fidelity theory have been developed. The most fundamental concept herein is the real-world reference knowledge standard paradigm. This so-called fidelity referent formalizes the natural level of indirection of fidelity measurement i.e. one can actually never measure against reality itself but against an approximated interpretation of reality. By explicitly linking the real-world knowledge error and uncertainties to its structure the fidelity referent transforms the problem of esoteric or ‘exact’ fidelity into a practical evidence-based assessment of simulation fidelity.

From this fidelity referent paradigm a mathematical definition for the practical simulation fidelity has been created. This formulation is supported by twelve fidelity theorems, which together outline the basic principles, propositions and postulates for a common unified simulation fidelity assessment methodology. Two major conclusions are formalized here. The first is the fact that fidelity is an absolute property of any model or simulation characterizing its degree of realism and doesn’t equate to the relative judgment of model or simulation validity. Second, this characterization of realism is best expressed by an enumeration of various multidimensional and multi-facetted measurement methods and metrics. This umbrella approach is also reflected in the descriptive concepts for fidelity characterization: resolution, accuracy, interaction, temporality, causality, precision and sensitivity.

10.1.2 Simulation Fidelity Implementation

Chapters 5 to 8 presented a detailed and practical implementation of the unified fidelity framework concepts defined in chapter 4. The fidelity referent paradigm has practically been implemented as a knowledge-base architecture composed of a structured set of generic and interrelated specification templates along with additional mathematical formulations. These templates cover three major areas: knowledge management, the actual real-world reference knowledge and elicitation knowledge including uncertainties and errors. The real-world reference knowledge has been subdivided in two orthogonal and complementary data sets: structural and behavioral knowledge.

A similar knowledge-base structure has been developed to specify relevant knowledge about the simulation system architecture and its replication of reality. To facilitate easy comparisons the knowledge of the real-world replication is specified in similar terms and format as the referent knowledge. The simulation system specification knowledge base also provides a necessary means for the long overdue formal knowledge specification of conceptual and simulation model capabilities. This makes the assessment of simulation fidelity and its integration within the simulation system development and validation process easier.

As said in the previous section practical fidelity measurement is a multi-dimensional and multifaceted problem in which various kinds of qualitative and quantitative measures can and have to be utilized simultaneously. In this thesis a basic taxonomy has been developed that unifies the most common and elementary fidelity measurement methods and metrics. Although not fully exhaustive the application of this taxonomy clearly demonstrated how fidelity metrics and measurement methods can be implemented and utilized in real simulation practice to measure the difference between the real-world system reference knowledge and replicated real-world system knowledge in the simulation system knowledge base. The taxonomy structure directly derives from the unified fidelity framework referent and simulation system knowledge-base architectures developed in chapters 5 and 6. This taxonomy is the pragmatic answer to what constitutes a simulation fidelity measure or metric. Since there exists no single unique measure or metric for fidelity it provides a structured set of possible and reusable measures or metrics instead, which can be selected and combined from this taxonomy to properly suite the application and problem domain requirements at hand.

To tackle the returning question in the modeling and simulation of “how much fidelity is good enough” a formal specification of the required level of fidelity from a simulation system necessary to meet the user needs and objectives has been defined within the unified fidelity framework. These so-called fidelity requirements consist of three elements: the required fidelity referent, the required fidelity measurement function set and most important the fidelity tolerance set. These fidelity tolerances specify the boundaries in which the measured level of fidelity must lie in order to completely fulfill the user needs.

Using the definition for fidelity requirements a transparent definition for simulation validity has been developed in terms of the next proposition: ‘the measured level of fidelity should meet the required level of fidelity’. If this proposition is true for a simulation system this system is said to be valid. It was shown how this proposition translates into a coherent fidelity-based verification and validation process that enhances the modeling and simulation enterprise.

To address the various simulation system comparisons, suitability and other fidelity trade-off decision issues, the concepts of fidelity performance and effectiveness have been introduced. These fidelity concepts provide overall measures that build upon a standard multi-criteria analysis method to compare simulation model and system component alternatives during simulation development and validation. This facilitates well-founded and cost-effective simulation design and development decision-making.

10.1.3 Simulation Fidelity Management

Fidelity assessment is a process that involves a large amount of real-world and simulation system data that have to be specified and analyzed with the use of a broad range of the measurement and evaluation methods of various kind and nature. Therefore, careful planning and management of fidelity assessment activities in all stages simulation system development is essential for the successful assessment of simulation fidelity in today’s complex and demanding simulation systems. More importantly, the commonly adopted ad-hoc approaches to fidelity assessment are less likely to provide sufficient and convincing evidence for the specifying the level of simulation fidelity with sufficient certainty to make reliable usage of the simulation results. To address this important issue the unified fidelity framework offers a process model that integrates all other developed unified fidelity definitions, concepts, formalisms and techniques into a single coherent and consistent fidelity assessment methodology for the modeling and simulation enterprise.

This so-called fidelity management process is a systematic and well-structured process model to assist users, developers and VV&A agents in applying and managing all those fidelity issues and assessment activities that occur in the entire simulation system life-cycle. Its well-organized series of traceable fidelity assessment activities and tasks smoothly integrate into both the simulation system development and the VV&A process. This not only facilitates the repeatability but also enhances the effectiveness and efficiency of the fidelity management process. The fidelity management process is a methodology that can be tailored and extended by the user to fully suit the particular needs of his or her own organization or a specific simulation application at hand. In other words the fidelity management process presented in this thesis provides a roadmap to carefully plan and systematically execute simulation fidelity assessment activities. However, the exact route to be chosen is up to the user and depends on his simulation system objectives, time and resources to achieve these.

10.1.4 Lessons-Learned from two Simulation Fidelity Case-Studies

The unified fidelity framework developed in this thesis has been applied to two practical simulation system case studies for proof of concept. The first case study focused on an HLA-based distributed simulation of a civil airspace system for research in future air-traffic control and management concepts. The second case study concerned a unitary stand-alone flight simulator for pilot training. Although, limited in scope these case studies, being very different in nature, demonstrated the unified fidelity framework and underlying concepts and paradigms prove to be a promising and viable basis for a future standard fidelity theory and practice. Major benefits experienced in both case studies include a better definition of what, how and when fidelity assessment activities have to be performed, clearly defined requirements that allow for trade-off decisions and setting priorities during simulation system development, more efficient elicitation and organization of real-world and simulation data, easier and systematic identification of sources causing large and unacceptable fidelity discrepancies, and define strategies to solve this fidelity issues

The case-studies also showed that the developed unifying fidelity theory and associated practices is not yet a fully grown and steady development. Although, the developed unifying fidelity framework addresses many of the former fidelity problems the case-studies identified issues that are not or only partially addressed. This is not surprising when one considers that the research area of modeling and simulation fidelity is still in a premature and experimental state. The major contribution of this thesis to the modeling and simulation community is that it is the first known scientific publication, which brings all aspects of simulation fidelity together within a single formal fidelity theory and application framework.

Another important lesson-learned from the case studies is that the inherent multidimensional and multifaceted nature of simulation fidelity implies that in practice fidelity assessment is a very complex and hard task to be handle by hand. Therefore, the development and use of a general purpose or domain tailored automated tool-suit to assist the simulation system developers and VV&A practitioners is indispensable for a cost-effective application of formal fidelity assessment processes. Without such automated tools the adoption of more rigorous fidelity theory and practice standards, even though necessary and highly desired by the modeling and simulation community, will be highly unlikely unless they are enforced by governing bodies. Particularly, if the complexity and scale of simulation systems continuous to increase at the current rate, automated tools are mandatory for practical and economical feasible implementations of rigorous and systematic fidelity assessment methodologies.

10.2 Future Research Challenges

Rigorous assessment of fidelity is one of the most difficult and hard to grasp issues of the model and simulation enterprise. Substantial and exhaustive research endeavors in this area are still very limited. Because of this, simulation fidelity still remains a rather uncultivated area. The unified fidelity framework presented in this thesis has laid some essential fundaments for the development of a common standard fidelity theory and practice. This makes this thesis unique in its kind. However its developments are tentative and experimental. Many future research challenges have been uncovered by the development and practical application of unified fidelity framework that have to be tackled in order to move ahead towards a common simulation fidelity standard:

1. Due to the inherent multidimensional and multifaceted nature of simulation fidelity, its assessment and specification is a complex and time-consuming task. Therefore, automated tools need to be developed which assist the simulation developer and VV&A agents in the whole fidelity assessment process to increase its practical and economical feasibility. Such a fidelity assessment tool should form an integrated suite of at least the next sub-tools:

    - Simulation system and fidelity requirements specification tool
    - Real-world reference knowledge base tool
    - Simulation system knowledge base tool
    - Off/On-line fidelity measurement and analysis tool
    - Process management and fidelity reporting tool

2. To facilitate easy interoperability, exchange and reuse of fidelity referent and simulation system knowledge it is recommended to translate the knowledge specification templates developed in this thesis into equivalent XML schema. XML provides an implementation independent language for the creation and sharing of complex, structured data and documents across different computer platforms, Internet, tools and organizations.

3. Pragmatic fidelity assessment is an evidence-based process to quantify and qualify simulation fidelity up to a certain degree of certainty. This thesis briefly touched upon possible uncertainty analysis techniques and how they could fit in the fidelity assessment process. It is therefore to conduct research to more exhaustive uncertainty analysis techniques and how they can be utilized in the context of simulation fidelity assessment.

4. The unified fidelity requirements specification approach provides only a generic process and no readily available application or problem domain specific fidelity requirements. It is recommended to follow the guidelines of this thesis to develop standard sets of fidelity requirements from specific applications and problem domains, which by definition also comprises the development of authoritative fidelity referents for these same areas. The resulting fidelity requirements repositories from such research activity constitute a common reference base from which one can directly reuse requirements or decide on whether and how specific tailored requirements need to be developed. This will reduce the simulation system development time and cut costs. Especially, if such an approach is combined with research to the effects or contribution of standard simulation model components

performance (mathematical and physical hard/software) to the final level of achievable fidelity. Which is of great interest since the reuse of standard off-the-shelf simulation model components has become a widely established practice.

5. The presented taxonomy of fidelity measurement methods and metrics is a basis for the development more application and problem specific methods and metrics. To be able to address a wide range of simulation applications it is recommended to develop more exhaustive taxonomies. Particular, one should focus on the research and development of rigorous qualitative fidelity measurement methods that utilize subject matter expert opinions.

6. This thesis demonstrated how the underlying simulation system hardware and software implementation details, performance capabilities and limitations can affect the achievable fidelity. It has also been argued that metrics and measurement methods for this purpose are application implementation specific and do not directly quantify or qualify the level of fidelity. For this reason this thesis only touched upon this aspect of fidelity assessment. However, its importance during simulation system design and implementation requires additional research to complement the unified fidelity framework with application specific taxonomies of implementation oriented measures and metrics. Particular, one should focus on complex simulation hardware and software architectures as found in distributed simulation systems.

7. Conduct applied and fundamental research based on experimental usage and tailoring of the present unified fidelity framework management process from different problem and application domains. The lessons-learned from this kind of research will gradually help improve and refine the unified fidelity framework into a robust and widely acceptable fidelity theory and practice standard. The resulting repository itself will also provide substantial reusable practical knowledge and experience to reduce the development time and cut costs of new simulation systems.

8. Fidelity-based validation during the conceptual modeling stage is a difficult activity, which is primarily caused by the unavailability of an uniform and robust definition of the conceptual model. A better definition of a conceptual model is expected to help improve the fidelity assessment activities during this stage of simulation development. Therefore, more research is necessary to what a conceptual model comprises, is created and specified. The in here developed simulation system knowledge specification template could provide a basis to start off such a research effort.

9. Lack of knowledge of rigorous modeling, simulation and validation principles has hampered the development of robust and standard fidelity theory and practices in the past and within this research. The transformation of fidelity assessment, and modeling and simulation technologies in general, from an art into the desired more rigorous scientific and sound engineering practice starts with educating those already involved or newbie’s that want to get involved in the modeling and simulation business. Therefore it is recommended that educational institutes devote more time and resources to teaching modeling and simulation development and validation principles within their curriculums.

References

[1] Advisory Group for Aerospace Research & Development, *'Fidelity of Simulation for Pilot Training'*. AGARD-Report AR-159, AGARD: Neuilly Sur Seine, December 1980.

[2] Balci O., 'A Methodology for Certification of Modeling and Simulation Applications'. To be published in *ACM Transactions on Modeling and Simulation*. ACM Inc. New-York 2002.

[3] Balci O., R Nance, J. Arthur, *'Expanding our horizons in verification, validation and accreditation research and practice'*. In proceedings of the 2002 Winter Simulation Conference: San-Diego, December 2002 .

[4] Banks J., *'Handbook of Simulation: Principles, Methodology, Advances, Application and Practice'*. John Wiley and Sons: New York, 1998.

[5] Banks J., J.S. Carson, B.L. Nelson, *'Discrete Event System Simulation'*. Prentice Hall: Upper Saddle River NJ, 1999.

[6] Barnes A., 'Compromise between Accuracy and Realism in Flight Simulation''. *AIAA Modeling and Simulation Technologies Conference*. August 1991, Paper AIAA-1989-2920.

[7] Berger J.O., *'Statistical Decision Theory and Bayesian Analysis 2nd Edition'*. Springer Verlag New York Inc, New-York, 1985.

[8] Birta L.G., F.N. Ozmizrak, 'A Knowledge-Based Approach for the Validation of Simulation Models: The Foundation'. *ACM Transactions on Modeling and Computer Simulation*. Vol. 6., No. 1, January 1996, pp. 76-98.

[9] Blanchard B.S., W.J. Fabrycky, *'Systems Engineering and Analysis'* Third Edition. Prentice Hall: Upper Saddle River, 1998.

[10] Boys R., D. Darlingtion, 'Shades of Grey in Interoperability'. *98 Spring Simulation Interoperability Workshop*, March 1998, Paper 044.

[11] Brasse M., W. Huiskamp, O. Stroosma. 'A Component Architecture for Federate Development'. *99 Fall Simulation Interoperability Workshop*, September 99, Paper 025.

[12] Clark D., J. Duncan, 'A Fidelity Framework'. *97 Fall Simulation Interoperability Workshop*, September 97, Paper 037.

[13] Cockburn A., 'Basic Use-Case Template', TR.96.03a. Human and Technology, October 1998.

[14] Dahmann J. et al., 'HLA and Beyond Interoperability Challenges'. *99 Fall Simulation Interoperability Workshop*, September 99, Paper 073.

[15] Defense Modeling and Simulation Office (DMSO), '*HLA Interface Specification, version 1.3*'. Department of Defense: February1998.

[16] Defense Modeling and Simulation Office (DMSO), *'HLA Object Model Template, version 1.3*'. Department of Defense: February 1998.

[17] Defense Modeling and Simulation Office (DMSO), *'HLA Runtime Infrastructure Programmer's Guide, version 1.3*'. Department of Defense: April 1998.

[18] Defense Modeling and Simulation Office (DMSO), *'Federation Development and Execution Process (FEDEP) Checklist, version 1.5*'. Department of Defense: December 1999.

[19] Defense Modeling and Simulation Office (DMSO), *'HLA Rules, version 1.3*'. Department of Defense: April 1998.

[20] Defense Modeling and Simulation Office (DMSO), *'Verification, Validation and Accreditation (VV&A) Recommended Practice Guide*'. Department of Defense: November 1996.

[21] Defense Modeling and Simulation Office (DMSO), *'Verification, Validation and Accreditation (VV&A) Recommended Practice Guide, Build 2*'. On-line Version (http://www.msiac.dmso.mil/vva). Department of Defense: August 2001.

[22] Defense Modeling and Simulation Office (DMSO), *'VV&A RPG Glossary, Version 2.0'*. Department of Defense: 24 April 1999.

[23] Deflandre D., J.P.C. Kleijnen, '*Statistical Analysis of Random Simulation: Bootstrap Tutorual'*. Simulation News Europe, Issue 38/39, Dec 2003, pp. 29-34.

[24] Dynasim, 'Dynamic Modeling Laboratory (Dymola) Get Started Online Manual', (http://www.dynasim.se/).

[25] Dillard A.E., *'Validation of Advanced Flight Simulators for Human-Factors Operational Evaluation and Training Programs'*. Proceedings Foundations '02 V&V Workshop. Johns Hopkins University, Applied Physics Laboratory: September 2002

[26] DIS Steering Committee. *'The DIS Vision, A Map to the Future of Distributed Simulation Version 1'*. Report, DIS Steering Committee. Orlando: May 1994

[27] Drake P. 'Using the Analytical Hierarchy Process in Engineering Education'. *Int. Journal of Engineering*. Vol 14, No 3, 1998, pp. 191-196.

[28] Dubios D, H. Prade, 'Possibility theory, probability theory and multi-valued logics: a clarification'. *Annals of Mathematics and Artificial Intelligence. Vol 32 p.p. 35-66.* Kluwer Academic Publishers, 2001.

[29] Duong V.N., E. Hoffman, J.P. Nicolaon. *'Initial Results of Investigation into Autonomous Aircraft Concept (FEER-1)'*. First USA/Europe ATM R&D Seminar. Saclay, June 17-20, 1997.

[30] Easterling R.G., J. O. Berger, *'Statistical Foundations for the Validation of Computer Models'*. Proceedings Foundations '02 V&V Workshop. Johns Hopkins University, Applied Physics Laboratory: September 2002

[31] Ellison S.L.R. (editor). *'Quantifying Uncertainty in Analytical Measurement'*. EURACHEM/CITAC Guide Second Edition, 2000.

[32] Eurocontrol Division DED-2, *'STANS ATC Prototyping Platform, Overview'*. Eurocontrol Agency: Brussels, November 1998.

[33] Eurocontrol Experimental Centre, *'User Manual for the Base of Aircraft Data (BADA), Revision 3.1'*. Eurocontrol France, EEC note No. 25/98. Bretigny-sur-Orge: November 1998.

[34] European Co-Operation for Accreditation, *'Expression of Uncertainty of Measurement in Calibration'*. Ref EA4-02. December 1999.

[35] Federal Aviation Authority, *'Airplane Simulator Qualification'*. Advisory Circular AC-120-40B. September 1993.

[36] Feinberg J.M., 'New Tools for Distributed Simulation Verification and Validation'. *98 Spring Simulation Interoperability Workshop*, March 1998, Paper 099.

[37] Feinstein A.H., H.M. Cannon, 'Fidelity, Verifiability and Validity of Simulation: Constructs for Evaluation'. Wayne State University Marketing Department Working Paper 2001-006, 2001.

[38] Flach J.M., J.G. Holden, 'The Reality of Experience: Gibson's Way'. *Presence, Vol. 7, No 1*. February 1998, Pages 90-95.

[39] Forman E.H., M.A. Selly, *'Decision by Objectives: How to convince others that you are right'*. World Scientific, 2001. (ISBN: 9810241437).

[40] Forbus K.D., "Causality". Summer-School Lecture-notes Unit A2-1, Qualitative Reasoning Group, Northwestern University, Evanston IL.

[41] Foster L., 'Fidelity in Modelling and Simulation'. *97 Spring Simulation Interoperability Workshop*, March 1997, Paper 073.

[42] Foster L., K.Yelmgren, 'Accuracy in DoD High Level Architecture Federations'. *97 Fall Simulation Interoperability Workshop*, September 97, Paper 064.

[43] Gause D.C., G.M. Weinberg, *'Exploring Requirements: quality before design'*. Dorset House Publishing: New York, 1989.

[44] George G., S. Knight, E. Stark. 'Fidelity Requirements in Aviator Training Networks'. *AIAA Modeling and Simulation Technologies Conference*. August 1989, Paper AIAA-1989-3382.

[45] Gross C.G., et al. 'A case study in fidelity differential measurements'. *98 Spring Simulation Interoperability Workshop*, March 1998, Paper 185

[46] Gross D.C., R.H. Freeman, 'Measuring Fidelity Differentials in HLA Simulations. *97 Fall Simulation Interoperability Workshop*, September 97, Paper 125.

[47] Gross D.C., M. Roza, et. al., 'Report from the Fidelity Implementation Study Group'. *99 Spring Simulation Interoperability Workshop*, March 99, Paper 167.

[48] Gross D.C., 'Why Fidelity'. *99 Spring Simulation Interoperability Workshop*, March 99, Paper 168.

[49] Haddix F., 'Prescribing Fidelity and Resolution for CMMS'. *98 Spring Simulation Interoperability Workshop*, March 1998, Paper 085.

[50] Harmon S.Y., 'An Update to the FEX-Fidelity Framework'. *99 Fall Simulation Interoperability Workshop*, September 99, Invited Presentation for the FEX-ISG session.

[51] Harmon S.Y., S.M. Youngblood, 'Leveraging Fidelity to Achieve Substantive Interoperability'. *01 Spring Simulation Interoperability Workshop*, March 2001, Paper 114.

[52] Harmon S.Y., S.M. Youngblood, 'A Proposed Model for Simulation Validation Process Maturity'. *2003 Spring Simulation Interoperability Workshop*, March 2003, Paper 127.

[53] Harmon S.Y., 'Describing M&S in Words'. *SISO On-line Simulation Technology Magazine, Article 252*.

[54] Harmon S.Y., 'Just Words – About Truth'. *SISO On-line Simulation Technology Magazine, Article 288*.

[55] Harmon S.Y., 'Just Words – About M&S'. *SISO On-line Simulation Technology Magazine, Article 326*.

[56] Harmon S.Y., 'Just Words – Just Models'. *SISO On-line Simulation Technology Magazine, Article 361*.

[57] Hays R.T. '*Simulation Fidelity: A Concept Paper'*. Technical Report, US Army Research Institute for Behavioral and Social Sciences: Alexandria, Virginia, November 1980.

[58] Hays R.T., M.J. Singer, *'Simulation Fidelity in Training System Design: Bridging the Gap between Reality and Training'*. Springer-Verlag: London, 1989.

[59] Hoekstra J.M., R.N.H.W. van Gent, R.J.C. Ruigrok. 'Conceptual design of Free Flight with airborne separation assurance'. *1998 AIAA Modeling & Simulation Conference*, August 1998, Paper AIAA-98-4239.

[60] Hosman R.J.A.W., *'Aerospace Human Factors'*. Lecture Notes D-60, Delft University of Technology. Delft: February 1995.

[61] Huiskamp W., 'Interoperability between Virtual and Constructive Simulations within an HLA architecture'. *2001 European Simulation Interoperability Workshop*, June 2001, Paper 01E-SIW-031

[62] Hsu H., *'Probability, Random Variables & Random Processes'*. McGraw-Hill, Schaum' Outlines Sersies: New-York, 1997.

[63] IEEE, *'Standard for Distributed Interactive Simulation – Fidelity Description Requirements'*. IEEE-1278.5, 1995.

[64] Institute for Simulation and Training (IST) Reports, *Standard for Distributed Interactive Simulation – Fidelity Description Requirements*, IST-CR-96-04 (February 1996) and IST-TR-96-44 (November 1996).

[65] Joint Aviation Authorities, *'Joint Aviation Requirements: JAR-STD 1A Aeroplane Flight Simulators'*. JAA-HQ: April 1997.

[66] Kane M.T., 'An Argument-Based Approach to Validity'. *Psychology Bulletin Volume 112, No 3, 1992.*

[67] Kleijnen J.P.C., 'Validation of Models: Statistical Techniques and Data Availability'. *Proceedings of the 1999 Winter Simulation Conference*. Pp647-653.

[68] Kleindorfer G.B., et. al., 'Validation in Simulation: Various Positions in the Philosophy of Science'. *Management Science Volume 44, No. 8, August 1998*.

[69] Klir G.J., B Yuan, *'Fuzzy Sets and Fuzzy Logic: Theory and Applications'*. Prentice Hall, Upper Saddle River, NJ, 1995.

[70] Klir G.J., et al., *'Fuzzy Set Theory: Foundations and Applications'*. Prentice Hall, Upper Saddle River, NJ, 1997.

[71] Klir G.J., 'On fuzzy set interpretation of possibility theory'. *Fuzzy Sets and Systems. Vol 106, no 3. pp. 263-273*. Elsevier Science, 1999.

[72] Klir G.J., R. M. Smith, 'On measuring uncertainty and uncertainty-based information: recent developments'. *Annals of Mathematics and Artificial Intelligence. Vol 32 .p. 5-31*. Kluwer Academic Publishers, 2001.

[73] Knepell P.L., D.C. Arango, *'Simulation Validation: A Confidence Assessment Methodology'*. IEEE Computer Society Press monograph. Los Alamitos CA, 1993.

[74] Kohlas J, P.A. Monney. *'A Mathematical Theory of Hints: An Approach to the Dempster-Shafer Theory of Evidence'*. Springer-Verlag: Berlin, 1995.

[75] Kotonya G., I. Sommerville,*' Requirements Engineering: Processes and Techniques'*. John Wiley and Sons: Chichester, 1998.

[76] Lacy L. et. al. 'Developing a Consensus Perspective on Conceptual Models for Simulation System'. *01 Spring Simulation Interoperability Workshop*, March 2001, Paper 74.

[77] Lane N.E., E.A. Alluisi. *'Fidelity and Validity in Distributed Interactive Simulations: Questions and Answers'*. Document D-1066, Institute for Defense Analyses. Alexandria (VA), November 1992.

[78] Lipschutz S., *'Set Theory and Related Topics'*. Second Edition. McGraw-Hill, Schaum' Outlines Sersies: New-York, 1998.

[79] Lipschutz S., *'Probability'*. Second Edition. McGraw-Hill, Schaum' Outlines Sersies: New-York, 2000.

[80] Ljung L., *'System Identification: Theory for the User'*. PTR Prentice Hall: Englewood Cliffs, New Jersey, 1987.

[81] Law A.M., W.D. Kelton, *'Simulation Modeling and Analysis, 3rd edition'*. McGraw-Hill Book Company: Boston 2000.

[82] Loucopoulos P., V. Karakostas, *'System Requirements Engineering'*. McGraw-Hill Book Company (UK): London, 1995.

[83] McDonald B. 'A proposed process for defining required fidelity of simulations'. *98 Spring Simulation Interoperability Workshop*, March 1998, Paper 234.

[84] Meloche P., M.W. Blake, D. Johnson. 'Joint FAA/Army/NASA Federation'. *98 Fall Simulation Interoperability Workshop*, September 1998, Paper 122.

[85] Merriam-Webster's on-line Collegiate Dictionary. http://www.webster.com.

[86] Metz M., S.Y. Harmon. 'Using Subject Matter Experts for Results Validation'. *02 Spring Simulation Interoperability Workshop*, April 2002, Paper 095.

[87] Meyer R.J., 'You say resolution, I say fidelity, You say accuracy, I say detail,…. Let's not call the whole thing off!'. *98 Spring Simulation Interoperability Workshop*, March 1998, Paper 245.

[88] Miles E., P. Davis, D. Wing. 'Development of a Free Flight Simulation Infrastructure'. *AIAA Modeling and Simulation Technologies Conference*. August 1999, Paper AIAA-1999-4193.

[89] Montgomery D.C. *'Design and Analysis of Experiments 5th edition'*. Wiley Text Books, 2000.

[90] Montgomery D.C. et. al. *'Engineering Statistics 3rd edition'*. Wiley Text Books, 2003.

[91] Mulder J.A., J.C. van der Vaart. *'Aircraft Responses to Atmospheric Turbulence'.* Lecture notes D-47. Delft University of Technology. Delft: August 1993.

[92] Muller P.A. *'Instant UML'*. Wrox Press Ltd. Birmingham UK:1997

[93] Nouragas P., et al., 'Fidelity Assessment in Aviation Related Simulation Applications'. *97 Fall Simulation Interoperability Workshop*, September 97, Paper 116.

[94] Neefs R. *'Functional Modeling of Airspace'*. Preliminary thesis report. Delft University of Technology, Control and Simulation Division. Delft: December 2000.

[95] Oberkampf W.L., et. al. 'Estimation of Total Uncertainty in Modeling and Simulation'. Report SAND2000-0840, Sandia National Laboratories. Albuquerque (NM): April 2000.

[96] Oberkampf W.L. 'Error and Uncertainty in Modeling and Simulation'. Invited Presentation for Risk-Based Design, ICASE/NASA Langley Research Centre: February 2001.

[97] Oberkampf W.L., et. al. '*Mathematical Representation of Uncertainty'*. Proceedings AIAA Non-Deterministic Approaches Forum. Paper AIAA-2001-1645. Seattle, WA: April 2001.

[98] Oberkampf W.L. '*Investigation of Evidence Theory for Engineering Applications'*. Proceedings AIAA Non-Deterministic Approaches Forum. Paper AIAA-2002-1569. Denver, CO: April 2002.

[99] Olsder G.J., *'Mathematical Systems Theory'*. Delft University Press: Delft, 1994.

[100] Oppenheim A.V., A.S. Willsky, *'Signals and Systems'*. Prentice-Hall: Englewood Cliffs NJ, 1983.

[101] Pace D.K. 'Conceptual Model Descriptions'. *99 Spring Simulation Interoperability Workshop*, March 99, Paper 025.

[102] Pace D.K., 'Development and Documentation of a Simulation Conceptual Model'. *99 Fall Simulation Interoperability Workshop*, September 99, Paper 017.

[103] Pace D.K., 'Simulation Conceptual Model Development'. *2000 Spring Simulation Interoperability Workshop*, March 2000, Paper 033.

[104] Pace D.K., 'Dimensions and Attributes of Simulation Fidelity, draft version'. *98 Fall Simulation Interoperability Workshop*, September 1998, Paper 017.

[105] Pace D.K., 'Fidelity Considerations for RDE Distributed Simulation'. *97 Fall Simulation Interoperability Workshop*, September 97, Paper 004.

[106] Pace D.K., 'Synopsis of Fidelity Ideas and Issues'. *98 Spring Simulation Interoperability Workshop*, March 1998, Paper 071.

[107] Pace D.K., 'Simulation Conceptual Model Development'. *2000 Spring Simulation Interoperability Workshop*, March 2000, Paper 033.

[108] Pace D.K., 'Issues Related to Quantifying Simulation Validation'. *2001 Fall Simulation Interoperability Workshop*, September 2001, Paper 050.

[109] Pace D.K., J. Sheehan, *'Subject Matter Expert / Peer Use in M&S V&V'*. Proceedings Foundations '02 V&V Workshop. Johns Hopkins University, Applied Physics Laboratory: September 2002

[110] Pace D.K., 'The Roles of Metrics in Simulation Verification and Validation'. *2003 Spring Simulation Interoperability Workshop*, March 2003.

[111] Paassen van M.M., O. Stroosma, J. Delatour. *'DUECA Data-Driven Activation in Distributed Real-Time Simulation'*. Proceedings AIAA Modeling and Simulation Technologies Conference. Paper AIAA-2000-4503. Denver Co: August 2000.

[112] Pallas-Areny R., J.G. Webster. *'Sensors and Signal Conditioning*'. Wiley & Sons. New-York: 1991.

[113] Pearman G.M., 'A method to asses fidelity of a DIS-compliant or HLA-compliant Legacy model'. *98 Spring Simulation Interoperability Workshop*, March 1998, Paper 201.

[114] Pressman R.S., *'Software Engineering: A Practitioner's Approach'*. 4th Edition. McGraw-Hill, New-York 1997.

[115] Renaud J.E. et. al., 'Uncertainty Quantification Using Evidence Theory in Multidisciplinary Design Optimization'. *Internet Presentation Department of Aerospace and Mechanical Engineering*. University of Notre-Dame, Notre Dame ID, USA.

[116] Robinson S., 'Simulation Model Verification and Validation: Increasing the Users' Confidence'. *Proceedings of the 1997 Winter Simulation Conference*. Pp53-59.

[117] Royal Aeronautical Society, *'Airplane Flight Simulator Evaluation Handbook: International Standards for the Qualification of Airplane Flight Simulators'*. Second Edition. The Royal Aeronautical Society: London, February 1995.

[118] Roza M. and P. van Gool. 'Simulation Free Flight in HLA Federations'. *2000 AIAA Modeling & Simulation Conference*, Denver: August 2000, Paper AIAA-2000-4399.

[119] Roza M. J. Voogd, P van Gool, 'Fidelity Considerations for Civil Aviation Distributed Simulations', *AIAA Modeling and Simulation Technologies Conference*, August 2000, Paper AIAA-2000-4397.

[120] Roza M., 'Experiences in (Ab)Using the FDM-ISG Fidelity Framework'. *99 Fall Simulation Interoperability Workshop*, September 99, Non-Paper Presentation for the FEX-ISG.

[121] Roza M., J. Voogd, H. Jense and P. van Gool, 'Fidelity Requirements Specification: A Process Oriented View'. *99 Fall Simulation Interoperability Workshop*, September 99, Paper 032.

[122] Roza M., P. van Gool and H. Jense, 'A Fidelity Management Process Overlay onto the FEDEP Model'. *98 Fall Simulation Interoperability Workshop*, September 98, Paper 083.

[123] Roza M., A. Dieterman, 'FASE: Future Airspace Simulation Environment'. In *Leonardo Times, Aerospace Magazine Vol 1*. Delft University of Technology, Faculty of Aerospace Engineering, March 2002.

[124] Roza Z.C. J. Voogd. H. Jense, 'Defining, Specifying and Developing Fidelity Referents'. *2001 European Simulation Interoperability Workshop*, June 2001, Paper 008.

[125] Roza Z.C., *'Simulation Fidelity: A preliminary study to fidelity assessment methodologies for HLA based distributed simulations'*. TNO-Report FEL-98-I311. TNO Physics and Electronics Laboratory: The Hague, December 1998.

[126] Roza Z.C., D.C. Gross and S.Y. Harmon. 'Report Out of the Fidelity Experimentation ISG'. *2000 Spring Simulation Interoperability Workshop*, March 2000, Paper 151.

[127] Roza Z.C., *'Eclipse 500 Jet Demonstration Flight Simulator Hard & Software Development Document'*. Technical Report to Siemens N.V, Delft University of Technology, Fac. Aerospace Engineering, 2002.

[128] Roza Z.C. *'The Design of a Generic Object-Oriented Framework for the Simulation of Dynamic Entities with applications'*. Memorandum 831, Delft University of Technology, Fac. Aerospace Engineering, 1997.

[129] Sargent R.G., 'Verification and Validation of Simulation Models'. *Proceedings of the 1998 Winter Simulation Conference*. Pp121-130.

[130] Saaty T., 'The Seven Pillars of the Analytic Hierarchy Process'. *ISAHP Proceedings*. Kobe 1999.

[131] Schild R., *'Rule Optimization for Airborne Aircraft Separation'*. Ph.D. Dissertation, Technische Universitat Wien, Technische Naturwissenschaftlichen Fakultat. Wien: 26 November 1998.

[132] Schow G., et al., 'Mathematical Foundations for a Fidelity Differential Calculus'. *97 Spring Simulation Interoperability Workshop*, March 1997, Paper 022.

[133] Schow G., R. Freeman, et al. 'Derivation and Publication of Simulation Fidelity Metrics'. *97 Fall Simulation Interoperability Workshop*, September 97, Paper 059.

[134] Shafer G., *'A Mathematical Theory of Evidence'*. Princeton University Press. Princeton 1976.

[135] Shannon R.E., *'Systems Simulation: The Art and Science'*. Prentice-Hall Inc. Englewood Cliffs, N.J: 1975

[136] Sheasby S.M., 'Merging Disparate SOMs into a Reference FOM – A Case Study'. *2000 Spring Simulation Interoperability Workshop*, March 2000, Paper 133.

[137] Spiegel M.R., *'Statistics'*. Third Edition. McGraw-Hill, Schaum' Outlines Sersies: New-York, 1999.

[138] Stadley V. et. al., 'An Investigation of Fidelity Metrics by the Validation of a Safeguards Monitoring System Simulation'. *2000 Spring Simulation Interoperability Workshop*, March 2000, Paper 12.

[139] Stalcup B., 'High Fidelity Distributed Simulation Analysis'. *98 Spring Simulation Interoperability Workshop*, March 1998, Paper 105.

[140] Steurs M, et. al. 'A Cybernetic Approach to Asses Flight Simulator Fidelity'. *2004 AIAA Modeling & Simulation Conference*, Rhode-Island: August 2004, Paper AIAA-2004-5442.

[141] Stevenson, Steve, "A Critical Look at Design, Verification, and Validation of Large Scale Simulations". *IEEE Computing in Science and Engineering Journal*, 1(3): 53-63, May/June 1999.

[142] Stevenson, Steve, "An Evidence-Based Approached to Fidelity," *Proceedings of the 2001 Summer Computer Simulation Conference, 15-19 July 2001, Orlando (FL),* Society for Computer Simulation, 2001.

[143] Swierstra S. 'STANS Simulation Facility of a Total Air Navigation System'. *Proceedings 13th European Simulation Multi-Conference*, Warsaw June 1999.

[144] Turrell C., et. al., 'HLA and Beyond: Interoperability Challenges'. *99 Fall Simulation Interoperability Workshop*, September 99, Paper 073.

[145] Turring A.C. 'Computing Machinery and Intelligence'. Mind Journal Issue 49, 1950, pp 433-466.

[146] Trucano T.G., et. al., 'Description of the Sandia Validation Metrics Project'. Report SAND2001-1339, Sandia National Laboratories. Albuquerque (NM): July 2001.

[147] Vaughn S., et. al., 'An Investigation of Fidelity Metrics by the Validation of a Safeguards Monitoring System Simulation'. 2000 *Fall Simulation Interoperability Workshop*, September 2000, Paper 012.

[148] White S.A., 'Use of Mixed Time and Fidelity Legacy Simulations in an Engineering Federation'. *99 Fall Simulation Interoperability Workshop*, September 99, Paper 020.

[149] Wieringa R.J.,' *Requirements engineering Frameworks for Understanding'*. John Wiley and Sons: New York, 1996.

[150] Wijnmalen D.J.D., *'TOPSYStem User Manual'*. TNO-Report FEL-97-H136. TNO Physics and Electronics Laboratory: The Hague, 1997.

[151] Wilkinson C., G. Vandervliet, M. Roscoe, 'Modeling and Simulation of the Ship-Helicopter Environment', *AIAA Modeling and Simulation Technologies Conference*, August 2000, Paper AIAA-2000-4583.

[152] Wilkinson C., G. Vandervliet, M. Roscoe, 'Determining Fidelity Standards for Shipboard Launch and Recovery Task', *AIAA Modeling and Simulation Technologies Conference*, August 2001, Paper AIAA-2001-4062.

[153] Woodward P. E. Bennett. 'Measuring Fidelity Differential in Simulator Networks'. , *AIAA Modeling and Simulation Technologies Conference*, August 1993.

[154] Zeyada Y., R.A. Hess. 'Computer Aided Assessment of Flight Simulator Fidelity'. *AIAA Modeling and Simulation Technologies Conference*, August 2002, Paper AIAA-2002-4693.

[155] Zeigler B., et al., *'Theory of Modeling and Simulation, 1st Edition'*. John Wiley & Sons Inc., New-York: 1976.

[156] Zeigler B., et al., *'Theory of Modeling and Simulation, 2nd Edition'*. Academic Press, Inc.: January 2000.

Abbreviations, Notations and Symbols

Symbol	Description
ω	Segment
ω	System input segment
$\omega_{\langle t_1,t_1 \rangle}$	Segment with initial and end times
Ω	Set of segments
Ω	Set of system input segments
$\varnothing$	Nonevent
$\varnothing$	Empty set
ρ	System output segment
Γ	Set of system output segments
Δ	System state transition function or mapping
Δ_{RS_i}	Fidelity qualification and/or quantification result set
$\Delta_{RS}{}^{-1}$	Inverse scaled fidelity quantifications or qualification result set
Λ	System output function or mapping
τ	Small time shift on a time base
(ω,ρ)	System input-output pair
$\rho(t)$	System output segment value at time t
$\omega(t_i)$	System input segment value at t_i
Δ_{Det}	Deterministic system state transition function or mapping
Λ_{Det}	Deterministic system output function or mapping
ω_i	Segment with index i
ν_i	System input variable value of input u_i
Δ_{Sth}	Stochastic system state transition function or mapping
Λ_{Sth}	Stochastic system output function or mapping
δR_w^{ref}	Real-world knowledge elicitation error set
δS_{appx}	Simulation system knowledge elicitation error set
$\tilde{\Delta}_{RS}{}^{-1}$	Reduced inverse scaled fidelity quantifications or qualification result set
$\delta\tilde{\Delta}_{RS_i}$	Total fidelity quantifications or qualification error set
B_{ext}	System external behavior sample

B_{in}	Internal behavior sample
$\tilde{B}_{in}$	Complex internal behavior sample
$\tilde{B}_{ext}$	Complex external behavior sample
$B_{length_{\max}}$	The maximum branch length
$B_{length_{ave}}$	The average branch length
B_{seq}	Interaction chain branching variation
C_{end}	Interaction chain end-condition
C_{model}	Simulation model configurable variable and parameter set.
Con_b	Conditional propositions branching variation initiation
C_{pre}	Interaction chain pre-condition
$\tilde{C}_{\Delta_{RS}}$	Reduced fidelity evaluator function set
$\hat{C}_{\Delta_{RS}}$	Required fidelity evaluator function set
$\hat{C}_{\Delta_{RS_{total}}}$	Total required fidelity evaluator function set
$c_{\Delta_{ac\_i}}$	An accuracy type fidelity evaluator function
$c_{\Delta_{res\_i}}$	A resolution type fidelity evaluator function
d_j	Fidelity qualification and/or quantification result
D	Set of system components
D_{seq}	Subset of real-world systems involved in a interaction chain
$dom(\omega)$	Segment domain
E_{exec}	Non-invertible simulation execution mapping
E_d	Set of system components being influenced by component d
E_m	Simulation model sub-model input-output mapping set
E_{rw}	Real-world knowledge elicitation function
E_{sim}	Simulation system knowledge elicitation function
E_{trig}	Interaction chain triggering event
$f(t_i)$	Time trajectory or signal
f_{cap_i}	System functional capability or a single element of F_{cap}
$\tilde{f}_{cap_i}$	Overall functional capability element of set F_{cap_c}
F_{cap_c}	System functional capability set
$F_{cap_{model}}$	Simulation model functional capabilities set
$F^{effective}_{\tilde{\Delta}_{RS_i}}$	Fidelity efficiency metric
$f^{effective}_{\tilde{d}_j}$	Fidelity efficiency rating function
f_{diff}	Trajectory difference function
f^{select}_{diff}	A characteristics trajectory quantity generator function

f_{dist}	Range distance measurement function
$F_{\tilde{\Delta}_{RS_i}}^{perform}$	Fidelity performance metric
$f_{\tilde{d}_j}^{perform}$	Fidelity performance rating function
$F_{esoteric}$	Esoteric fidelity
FI_b	Interaction chain branching variation set
$F_{required}$	Fidelity requirements
$F_{required_{total}}$	Total required simulation system fidelity capabilities
$F_{practical}$	Practical fidelity
$F_{sufficiency}$	Simulation fidelity sufficiency proposition
I	Independent system input variable or coordinates set
IB	Conditional proposition relationship set
IC^{rw}	Interaction causality knowledge set
IC_{chain_i}	Interaction mechanism or chain description
I_d	Component output to input mapping or coupling of component d
I_m	Simulation model sub-model interaction set
I_{prim}	Primary interaction chain
k_i^{-1}	Inverse proportional fidelity scaling mapping
$l(\omega)$	Segment length
$L_{bound_{c_{\Delta_i}}}$	Fidelity tolerance lower bound set
$l_{bound_{\tilde{d}_j}}$	Fidelity tolerance lower bound set element
$\underline{L}_{bound_{c_{\Delta_i}}}$	Maximum fidelity tolerance lower bound set
L_R^{elicit}	Limitations on the reality knowledge elicitation process
L_S^{elicit}	Simulation system knowledge elicitation process limitation set
L_S^{elicit}	Simulation system knowledge elicitation process limitation set
M^{fed}	Federation simulation model
$M_{sub_i}^{fed}$	Federate simulation model
$M_{sub_j}^{comp}$	Simulation model component
M_{model}^{rw}	Simulation model component real-world system model set
M_{config}^{rw}	Configurable real-world (sub)system models set
$M_{non\text{-}config}^{rw}$	Non-configurable real-world (sub)system models set
$M_{model_{total}}^{rw}$	Total simulation model real-world system model set
$M_{manifold}^{rw}$	Manifold real-world system model set
M_{sub}	Simulation model sub-models
N_{multi}^{rw}	Real world system multiplicity set

$Nof_{violations}$	Number of discrete range violations
n^{exec}_{multi}	Multiplicity of a real world system model
$n\left(M^{rw}_{model_{total}}\right)_{config}$	Total number of configurable real-world (sub)system models
$n\left(M^{rw}_{manifold}\right)$	Total number of manifold real-world (sub)system representations
$n\left(R^{ref}_{w}\right)_{system}$	The total number of (sub)systems
$n\left(R^{ref}_{w}\right)_{leafs}$	The total number of (sub)system leafs
$n\left(R^{ref}_{w}\right)_{forks}$	The total number of (sub)system forks
$n\left(R^{ref}_{w}\right)_{interaction}$	The total number of (sub)system interaction relationships
$n\left(S_{model}\right)_{M_{model}}$	The total number of simulation (sub)models
$n\left(S_{model}\right)_{M^{fed}_{sub}}$	The total number of federate (sub)models
$n\left(S_{model}\right)_{M^{comp}_{sub}}$	The total number of simulation component models
$n\left(S_{model}\right)_{M_{config}}$	The total number of configurable simulation (sub)models
$n\left(S_{model}\right)_{interaction}$	The total number of (sub)models interaction relationships
O	System output variable or coordinates set
P	Multi-variable set of system parameters
P_{config}	Real-world system model configurable parameter set
P_i	System parameter range set belonging to the output variable p_i
$P_{non\text{-}config}$	Real-world system model non-configurable parameter set
p_j	System parameter variable or coordinates set
$p_{RL_{aggregate}}\left(t_{exec}\right)_m$	Proposition on the m^{th} aggregate variable relationship
$p_{RL_{system}}\left(t_{exec}\right)_k$	Proposition on the k^{th} system variable interrelationship
$p_{RL_{ordinary}}\left(t_{exec}\right)_k$	Proposition on the k^{th} ordinary causal relationship
$p_{RL_{chiv}}\left(t_{exec}\right)_k$	Proposition on the k^{th} change in value causal relationship
Q	Multi-variable set of system state
$q(t)$	System state value at time t
q_i	Single element of an system state variable set
$Q_{complex}$	Complex system behavior quantifier state set
Q_i	System state range set belonging to the output variable q_I
R	Reality
$range_{ij}$	Multi-variable set range operator
$R_{bsystem}$	System variable interrelationship
r_i	Indicator rating
R_{im}	Imaginary reality
$RL_{aggregate}$	Aggregate variable relationship

RL_k^{cvc}	Real-world system change in value Boolean expression
RL_k^{oc}	Real-world system behavioral cause condition
RL_k^{oe}	Real-world system behavioral effect condition
R_{mat}	Material reality
R_{ref}	Fidelity referent
R_{ref}^{req}	Required fidelity referent
$R_{ref_{total}}^{req}$	Unified required fidelity referent
R_w	Real-world
R_w^{ref}	Real-world reference knowledge specification
S	Simulation system
S_{appx}	Approximated simulation system knowledge specification
S_{config}	Simulation system configuration setting knowledge
S_{exec}	Set of simulation executions
S_{ext}	External system knowledge specification
S_{in}	Internal system knowledge specification
S_{ind}	Aggregated indicator branch score
S_{model}	Simulation model knowledge
S_{model}^{exec}	Executable simulation model
$S_{model_k}^{rw}$	Real-world system model
S_{rwr}	Set real-world system replication knowledge during execution
S^{rw}	Real-world system set
$S_{ref_i}^{rw}$	Real-world system reference knowledge specification
$S_{support}$	Simulation support system knowledge
S_{IOF}	System input-output observation frame
T	Time base
$T_{\hat{C}_{RS}}$	Fidelity tolerance set
$T_{\hat{C}_{RS_{total}}}$	Total fidelity tolerance set
$T_{frame_{\min}}$	Minimum timeframe size
T_{model}	Simulation model time base
T_{speed_up}	Simulation model real time-base scaling factor
$Tol_{c_{\Delta_i}}$	Fidelity tolerance
$T_{\Re}$	Continuous time base
$T_{\Im}$	Discrete time base
$T_{t)}$	Past time interval

$T_{(t}$	Future time interval
$T_{[t_1,t_2]}$	Closed time interval
t	Present time
t_0	Minimal element of a time-base
t_1	Initial time of a time interval
t_2	Final time of a time interval
t_i	Time point within a time interval
t_n	Maximum element of a time-base
U	Multi-variable set of system input
$U_{bound_{c_{\Delta_i}}}$	Fidelity tolerance upper bound set
$u_{bound_{\tilde{a}_j}}$	Fidelity tolerance upper bound set element
$\underline{U}_{bound_{c_{\Delta_i}}}$	Minimum fidelity tolerance upper bound set
$\overline{U}_{c_i}$	Set of composite-component system inputs involved in $\tilde{f}_{cap_i}$
U_c	Multi-variable set of composite-component system input
$U_\delta\left(\tilde{\Delta}_{RS}{}^{-1}\right)$	Total fidelity quantifications or qualification uncertainty set
$U_\delta\left(R_w^{ref}\right)$	Real-world knowledge elicitation uncertainty set
$u_\delta\left(R_w^{ref}\right)_i$	Real-world knowledge elicitation uncertainty set element
$U_\delta\left(S_{appx}\right)$	Simulation system knowledge elicitation uncertainty set
$u_\delta\left(S_{appx}\right)_i$	Simulation system knowledge elicitation uncertainty element
u_i	Single element of an independent system input variable set
$\overline{U}_i$	Set of system inputs involved in f_{cap_i}
U_i	System input range set belonging to the input variable u_i
U_{model}	Simulation model exogenous variable set
UY_d	Input of component d that results from mapping I_d
V_{sim}	Simulation system validity proposition
variables	Multi-variable set variables operator
w_i	Relative indicator weight
X	System state variable or coordinates set
Y	Multi-variable set of system output
$\overline{Y}_{c_i}$	Set of composite-component system outputs involved in $\tilde{f}_{cap_i}$
Y_c	Multi-variable set of composite-component system output
$Y_{complex}$	Complex system behavior quantifier output set
y_i	Single element of an system output variable set
$\overline{Y}_i$	Set of system outputs involved in f_{cap_i}

Y_i	System output range set belonging to the output variable y_i
Y_{model}	Simulation model endogenous variable set
YU_j	Output of component j that is used in mapping I_d of component d
Z	System parameter variable or coordinates set

Appendix A: Glossary of Fidelity Related Terminology

There exist many definitions for fidelity related terminology and almost every publication on modeling and simulation provides its own definitions. To allow for more effective communication and avoid confusion by using the same term but with different meaning, the Simulation Interoperability Standards Organization (SISO) chartered a Fidelity Implementation Study Group (FISG) whose first task was the construction of a comprehensive glossary of fidelity related terminology. The author of this thesis has been one of the principle creators of this glossary. In the creation of the glossary many existing sources have been consulted and terms were discussed via an Internet forum and at interim meetings during two SISO simulation interoperability workshops. Even though, the compiled definitions may not be perfect but are considered as a very good step in the direction towards a common accepted terminology for the modeling and simulation community. In this appendix only an excerpt of the SISO-FISG glossary is presented with only those terms directly related to this thesis. For those readers interested in the complete glossary, they are referred to the first SISO-FISG report in which the entire 28 pages long glossary has been published [47].

A

Abstraction. 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.

Accreditation. Official acceptance or certification that a model, the data for a simulation or a simulation is suitable for a specific purpose or application.

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.

Aggregation. The ability to group items, whether entities or processes, while preserving the effects of item behavior and interaction while grouped.

Aleatory uncertainty. Uncertainty due to the inherent variation of an existing entity, system or object under consideration; variability.

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

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

Axiom. A statement or proposition used in the premises of arguments and assumed as self-evidently true without proof.

B

Behavior. The way in which a system responds to stimuli over time.

Benchmark. An accepted representation or standard of a process being modeled or simulated against which the results of other models or simulations are judged.

Black box model. A model whose inputs, outputs, and functional performance are known, but whose internal implementation is unknown or irrelevant.

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.

C

Causal order. A partial ordering of messages based on the “causally happens before” relationship.

Characteristic data. Empirical, synthesized or otherwise provided parameters describing the characteristics of the system or component being simulated.

Component class. An object class that is a component, or part of, a “composite” object which represents a unified assembly of many different object classes.

Computer hardware. Devices capable of accepting and storing computer data, executing a systematic sequence of operations on computer data, or producing control outputs.

Conceptual model. 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 recognizing assumptions and limitations.

Configuration. A collection of an item’s descriptive and governing characteristics, which can be expressed: in functional terms and in physical terms.

Context. The material surrounding an item that helps define its meaning.

Continuous system. A system for which the state variables change continuously with respect to time.

Correlation. A causal, complementary, parallel, or reciprocal relationship, especially a structural, functional, or qualitative correspondence between comparable entities.

D

Data quality. The correctness, timeliness, accuracy, completeness, relevance, and accessibility that make data appropriate for use. Quality statements are required for source, accuracy, up-to-datedness/currency, logical consistency, completeness, security and classification.

Data source. A publication, organization or subject matter expert that serves as an authoritative source of data used in a model or simulation.

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

Data value. A value associated with a data element; one of the allowable values of a data element.

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.

Data. Assumed, given, measured, or otherwise determined facts or propositions in a formalized manner suitable for communication, interpretation, or processing by humans or by automatic means.

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.

Detail. A separately considered part or item.

Deterministic algorithm. A process that yields a unique and predictable outcome for a given set of inputs.

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.

Deterministic. Pertaining to a process, model, simulation or variable whose outcome, result, or value does not depend upon chance.

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.

Discrete system. A system for which the state variables change instantaneously at separated points in time.

E

Empirical. Pertaining to information that is derived from observation, experiment, or experience.

Endogenous variable. A variable whose value is determined by conditions and events within a given model; internal variable.

Enterprise model. An information model(s) that presents an integrated top-level representation of processes, information flows, and data.

Entity. A distinguishable person, place, unit, thing, event, or concept about which information is kept.

Epistemic uncertainty. Uncertainty due to incomplete and/or the lack of information about an entity, system or object.

Error model. 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.

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

Event. An individual stimulus from one object to another at a particular point of time.

Exogenous variable. A variable whose value is determined by conditions and events external to a given model; external variable.

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.

Fast time. The duration of activities within a simulation in which simulated time advances faster than actual time.

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.

Fidelity. 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.

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.

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

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.

G

Glass box model. A model whose internal implementation is known and fully visible; white box model.
Granularity. Resolution.

Ground truth. The actual facts of a situation, without errors introduced by sensors or human perception and judgment.

H

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.

Hierarchy. A ranking or ordering of abstractions.

Human-in-the-loop simulation. A simulation that requires human interaction.

I

Imaginary reality. A concept that has no exact counterpart in the material universe although parts of it may have counterparts in the material universe. Imagined reality may have a nonzero intersection with but can never be a proper subset of material reality.

Implementation. The means by which a synthetic environment, or portions of a synthetic environment, is realized.

Information system. The organized collection, processing, maintenance, transmission, and dissemination of information in accordance with defined procedures, whether automated or manual.

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.

Infrastructure. An underlying base or foundation; the basic facilities, equipment, and installations (e.g., systems and applications, communications, networks, architectures, standards and protocols, and repositories) needed for the functioning of a system.

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.

Input. A variable at the boundary of an organism or system through which information enters; the set of independent conditions, properties or states that effects a change in a system's behavior.

Interaction. The way in which object, components, systems, models or simulations affect or influence each other.

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.

J

K

Knowledge. The sum or result of what has been perceived, discovered or learned.

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

L

Latency. The observable delay between stimulus and response.

Level of detail. Resolution.

Live entity. A perceptible object that can appear in the virtual environment but is unaware and non-responsive to the actions of virtual entities.

Live simulation. A simulation involving real people operating real systems.

Local time. The mean solar time for the meridian of the observer.

M

Material reality. The material universe (or those parts of it) that is pertinent to an application domain.

Mathematical model. Any system of assumptions, definitions and equations that represents particular physical phenomena.

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.

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.

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 but measures attributes of system behavior.

Meta-model. A model of a model.

Methodology. The system of principles, practices, and procedures, applied to a specific branch of knowledge.

Metric. A measure of the extent or degree to which an item possesses and exhibits a certain quality, property, or attribute.

Mock-up. A full-sized structural, but not necessarily functional, model built accurately to scale, used chiefly for study, testing, or display.
Model. A physical, mathematical, or otherwise logical abstract representation of a system, entity, phenomenon, or process with its own assumptions, limitations and approximations.

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.

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.

Monte Carlo algorithm. A statistical procedure that determines the occurrence of probabilistic events or values of probabilistic variables for deterministic models.

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

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

N

O

Object. An atomic entity composed of state, behavior and identity.

Object-based methodology. A methodology adhering to the properties of object-orientation.

Object-oriented methodology. Object-based methodology.

Observable. Capable of being observed systematically or scientifically; a physical property that can be observed or measured directly.

Output. A variable at the boundary of an organism or system through which information exits; the products, results or the observable parts of system behavior.

Output validation. The process of determining the extent to which the output represent the significant and salient features of distributions or real world systems, events, and scenarios.

P

Parameter. A variable or constant that specifies those internal system properties, which determine the system characteristics (structure and behavior); A constant or variable that distinguishes a special instance of a general mathematical expression.

Perceived truth. That subset of ground truth acquired or distorted by sensors, human perception or judgment; the situation as perceived by an observer

Perception. An observer's awareness or appreciation of objects, processes or situations in his environment mediated through their sensory organs.

Period. The time interval between successive events in a simulation.

Physical model. A model whose physical characteristics resemble the physical characteristics of the system being modeled; a mock-up.

Platform. A generic term describing a level of representation equating to vehicles, aircraft, missiles, ships, fixed sites, etc., in the hierarchy of representation possibilities.

Precision. A measure of how meticulously or rigorously computational processes are described or performed by a model or simulation.

Probabilistic model. Stochastic model.

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.

Pseudo-code. A description of control and/or data structures in a natural language with no rigid rules of syntax.

Purpose. The objective for which a simulation or simulation exercise is intended; goal.

Q

Qualitative data. A non-numeric description of a person, place, object, event, activity, or concept.

Quantitative data. Numerical expressions that use Arabic numbers, upon which mathematical operations can be performed.

R

Reality. The quality or state of being actual or true.

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.

Real-time. In modeling and simulation, simulated time advances at the same rate as actual time.

Real-world. The set of real or hypothetical causes and effects that the model or simulation attempts to replicate. The real world defines one standard against which fidelity is measured that includes both imagined reality and material reality in order to assess of simulation fidelity when future concepts and systems are involved.

Referent. A codified body of knowledge about a thing being simulated.

Resolution. The degree of detail used to represent aspects of the real world or a specified standard or referent by a model or simulation; granularity.

S

Scenario. The description of a set of initial and termination conditions, entities that must be represented, storyboard and time line of significant events imposed on simulation systems to achieve the simulation execution objectives.

Segment. A portion of a session that is contiguous in simulation time and in wall-clock time (sidereal time).

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

Sidereal time. Time that is independent of simulation clocks, time zones, or measurement errors.

Simuland. The system being simulated by a simulation.

Simulated time. Time as represented within a simulation; virtual time.

Simulation clock. A counter used to accumulate simulated time.

Simulation environment. An entire simulation framework including software, hardware, architecture, infrastructure and interfaces where models or simulations are developed and executed.

Simulation execution. The execution of a simulation application over time.

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.

Simulation model. A digital and/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.

Simulation time. A simulation’s internal representation of time which may accumulate faster, slower, or at the same pace as sidereal time.

Simulation. 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.

Simulator. A device or physical system that implements or performs simulation.

Software model. The actual compilable and linkable software source codes that implements algorithms and data flow representing one or more mathematical models.

State transition. A change from one state to another in a system or simulation.

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 or simulation.

State. The values assumed at a given instant by the variables that define the characteristics of a system, component, or simulation.

Steady state. A situation in which a model, process, or device exhibits stable behavior independent of time.

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; probabilistic model.

Stochastic. Pertaining to a process, model, or variable whose outcome, result, or value depends on chance. See deterministic.

Structural model. A representation of the physical or logical structure of a system.

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.

Symbolic model. A model whose properties are expressed in symbols.

Symbology. A graphic representation of concepts or physical objects.

System. A collection of components organized to accomplish a specific function or set of functions.

T

Taxonomy. A classification system that provides the basis for classifying objects for identification, retrieval and research purposes.

Time management. A collection of mechanisms and services to control the advancement of time within a simulation system during an execution.

Time variable. A variable whose value represents the model or simulation time.

Time. The measurable aspect of duration. Time makes use of scales based upon the occurrence of periodic events.

Tolerance. 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.

Topology. Any relationship between connected geometric primitives that is invariant under transformation by continuous mappings.

Truth. Conformity to fact or actuality; reality.

U

Unit. A basis of measurement.

Unit conversion. A system of converting measurement from one basis to another.

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.

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.

Validity. The quality of being inferred, deduced or calculated correctly enough to suit a specific application.

Variable. A quantity or data item whose value can change.

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.

W

Wall clock time. A simulation system's measurement of true global time, where the measurement is typically output from a hardware clock.

White box model. Glass box model.

Appendix B: Overview of Error & Uncertainty Quantification and Qualification Methods

Due to the nature of fidelity measurement no method existing today or developed in the near future will be able to give exact correct or reliable result of how close a simulation represents the real-world. All fidelity measurements include some form of uncertainty and error that result from various and often combined sources. Therefore, for a fidelity measurement result to be useful it is equally important to qualitatively and/or quantitatively known the possible associated uncertainty and error. In this appendix a brief overview is given of both formerly used and potential adequate methods for qualification and quantification of error and uncertainty. This overview is by no means complete nor indented to provide an in-depth discussion for the application of these methods, since uncertainty and error methods and their applications are a research topic on their own.

B-1 Statistical Techniques

The most applied and researched methods in the context of simulation uncertainty and error methods are statistical techniques [95] [129] [146]. These techniques require both the real-world system and the simulation system to be completely observable for the properties of interest. In general statistical techniques are used to assess error and uncertainty due to system variability. There exist many publications on statistical techniques therefore only a cross-section of the major techniques will be presented here briefly. The interested reader is referred to publications such as [7] [30] [31] [34] [67] [89] [90] [137] [146] for more background and their applications to assess knowledge error and uncertainty.

Most popular and simple statistical methods are measures of dispersion that are applied to an observed sample S of n-observations from a real-world population:

- *Sample range*, defined as:

$$r = \max\left(S\right) - \min\left(S\right) \qquad \text{(B.1)}$$

- *Sample variance*, defined as:

$$s^2 = \frac{\sum_{i=1}^{n}\left(x_i - \overline{x}\right)^2}{n-1} \qquad \text{(B.2)}$$

in here is the *sample average* defined as:

$$\overline{x} = \frac{\sum_{i=1}^{n} x_i}{n} \qquad \text{(B.3)}$$

The previous discussed measures can be used to assign so-called *standard error*, or standard deviation for a sample distribution of statistic [90] [137]. Using *Expressions B-2* and *B-3* the statistics sample average and variance the standard errors are respectively defined as:

$$\sigma_{\bar{x}} = \frac{s}{\sqrt{n}}$$

$$\sigma_{s^2} = \sqrt{\frac{3s^4 - \frac{n-3}{n-1}s^4}{n}} \tag{B.4}$$

Both standard errors can be expressed as the percentage of each sample value as another indication of the error and uncertainty [89]. In cases the standard errors are hard to determine analytically or logically, for instance in case of non-normal distributions, bootstrapping can effectively be used to make a good alternative estimate [23] [67] [90]. For instance for a set of B bootstrap samples drawn from the original sample the standard error for the estimated average is determined as follows [90]:

$$\sigma_{\bar{x}} \approx \sqrt{\frac{\sum_{k=1}^{B}\left(\bar{x}_k^* - \bar{x}_m^*\right)^2}{B-1}} \tag{B.5}$$

where $\bar{x}_k^*$ is the k^{th} bootstrap sample average and $\bar{x}_m^*$ is the average of all k bootstrap sample averages.

Another statistical technique based on the mean and variance or the standard deviation of a sample distribution of statistic S is the confidence interval estimation. This technique defines an interval in which with a certain degree of confidence, expressed in a percentage, the true value of S must reside. Usually, the distribution is considered to be normal, which is approximately true for sample size larger than 30. In these cases the confidence interval for the estimate of the population mean and standard deviation are given by:

$$\bar{x} \pm z_c \frac{s}{\sqrt{n}}$$

$$s \pm z_c \frac{s}{\sqrt{2n}} \tag{B.6}$$

In here the values for z_c is determined from a normal-curve area table [137]:

Confidence	99.73%	99%	98%	...	50%
z_c	3.00	2.58	2.33	...	0.6745

Table B-0-1 Excerpt from a Normal-Curve Area Table

For small size samples with a size less than 30 the student's *t*-distribution is one of the most commonly applied technique to determine confidence intervals. In general, the confidence interval for a population means is given by:

$$\bar{x} \pm t_c \frac{s}{\sqrt{N-1}} \tag{B.7}$$

Like for a normal distribution t_c is the confidence coefficient that depends on the desired level of confidence. The confidence coefficient is determined from a student's t distribution curve area table [137]. Other similar techniques often used in determining confidence intervals for small sample sizes are based on chi-square and *F* distributions [89] [90].

Besides the above-discussed confidence interval techniques based on assumed distributions there exist other methods that are independent of such distributions and parameters. These are called *nonparametric* tests. The four mostly used test are: sign test, Wilcoxon signed rank test, Mann-Withney test and Kruskal-Wallis test [90] [137]. Furthermore, worth to mention here are analysis techniques to assess the error and uncertainty for the often-in simulation used regression models [67] [89] [129]. Such techniques include residual analysis, determination of R^2 and goodness of fit tests [89].

B-2 Probability Theory

Another traditional and often encountered technique for assessing uncertainty is probability theory. Probability theory is defined as the mathematical modeling of the phenomenon of chance or the degree of belief an event will occur. There exist two ways to obtain the probability of an event [79]:

- *A priori definition*: suppose an event can occur in *s* ways out of a total of *n* equally likely possible ways, then the probability of this event is *s/n*.
- *A posteriori definition*: suppose after n repetitions, where *n* is very large, an event occurs *s* times, then the probability of this event is *s/n*.

The basis of probability theory is formed by the three axioms using set theory. Let *S* be a sample or probability space, let ϑ be the class of all events, and let *P* be a real-valued function defined on ϑ. Then *P* is called a probability function, *P(A)* is called the probability of the event *A* when the next axioms hold:

1. $\forall A \in S \rightarrow P(A) \geq 0$
2. $P(S) = 1$ (B.8)
3. $P(A \cup B) = P(A) + P(B)$

A probability functions exist in two forms discrete and continuous. The later is also known as a probability density function and looks as follows:

$$P(a \leq X \leq b) = \int_a^b f(x)dx \tag{B.9}$$

In *Expression B.9 f(x)* satisfies the above mentioned probability axioms as follows:

$$f(x) \geq 0$$
$$\int_{-\infty}^{\infty} f(x)dx = 1 \qquad \text{(B.10)}$$

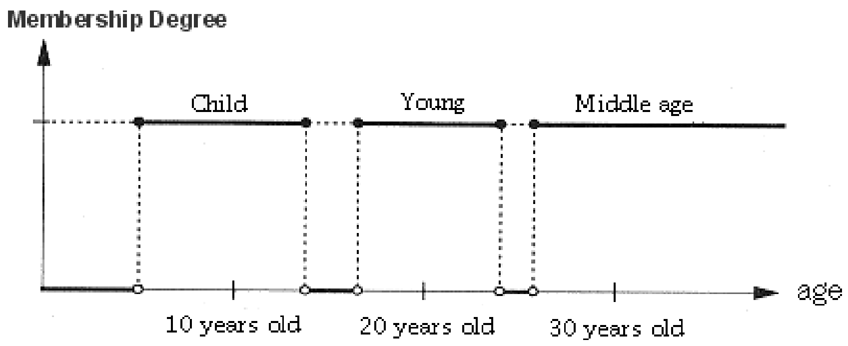
Probability theory it self does not specify how this probability function P should be obtained or looks-like, it can be assigned arbitrarily as long as it satisfies the above discussed axioms. Which is both its strength and weakness. The strength is that a wide variety of knowledge and subject matter expert sources can be used to construct a probability function in practice. A practical examples of how these aspects of probability theory can be used to model and assess uncertainty are found in [97] and [142]. However in this strength lays also its weakness since there may be only a few reliable sources available or several sources may be conflicting. The reliability of the probability of an event thus depends on the quality of the assigned quality probability function with respect to the actual limiting relative frequencies. Testing this is usually done in the form of statistics in case the actual system of events can be observed.

B-3 Fuzzy Set Theory

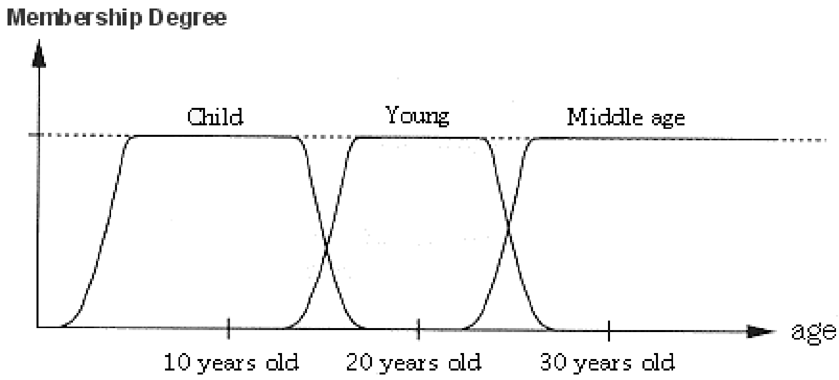
Traditional set theory defines the membership of an element as a crisp Boolean predicate [78]. It is either an element of a set or it is not. Fuzzy set theory on the otherhand the element membership of a set can gradually vary from non membership to full membership. This membership grade of an element in fuzzy sets is defined by means of a so-called *membership or possibility function*, which can be described any arbitrary continouos function or discretely by a set of paired values. Standard and normal fuzzy sets define the membership grade of an element x in the set X as a mapping onto a unit interval as follows:

$$\Lambda : X \to [0,1] \qquad \text{(B.11)}$$

The purpose and utility of fuzzy set theory is its capability to to deal with problems involving knowledge expressed in vague or fuzzy, linguistic terms. In specifying and communication of knowledge or information about an object the usage of qualitative statements is a common practice. Some of these statements are vague or uncertain because the precise datum value isn't fully known or the datum value is not measurable by an exact scale. In this sense an object may be a member of a set to some degree or a logical proposition may hold true to some extend and can only properly be described by fuzzy sets instead of traditional crisp set. As an example consider the *figure B-1* at het next page, which specifies the classification of child, young and middle age adult. Obviously wether a 16 to 19 yeasr old human is still a child or a young adult depends on the personal mental development of this human and therfore no distict boundary between both classes can be drawn.



Classification of Child, Young and Middle Age in terms of classical set theory



Classification of Child, Young and Middle Age in terms of fuzzy set theory

Figure B-1 Classical set theory versus fuzzy set theory

Uncertainty is represented in terms of fuzzy set theory by a *possibility function* that assigns to a set of alternatives (propositions, prescriptions, predictions, etc.) a number in the unit interval, which express the degree of evidence (belief, likelihood, etc.) that the true alternative is in the set [72]. A possibility function is therefore in these cases also called uncertainty function. It must be noted that the meanings of linguistic terms and the quality or correctness of the associated uncertainty functions depend on the context in which they are used. Similar to probability theory this is also its weakness in assessing the relaibility of any formal uncertainty specification based on fuzzy set theory. A good source for more information on the theory and application of fuzzy set theory in the context of uncertainty measurement can be found in [69].

B-4 Possibility Theory

There exist many interpretation of possibility theory for assessing uncertainty. Nowadays the fuzzy-set interpretation, as introduced by the founder of fuzzy set theory, is the most widely known and applicable one [28] [71]. Even though the principle ideas existed already, it was he who firstly used the terms possibility theory and possibility measure to express information in terms of fuzzy propositions supported by evidence.

In general the basis of possibility theory is characterized by two fuzzy measures, *possibility* and *necessity* [71]. Possibility is defined for a universal set X as:

$$Pos : P(X) \rightarrow [0,1] \qquad \text{(B.12)}$$

For expression B.12 the following axioms should hold: $Pos(\varnothing)=0$ and $Pos(X)=1$, and for any family $\{A_i \mid A_i \in P(X), i \in I\}$ with I an arbitrary index set:

$$Pos\left(\bigcup_{i\in I} A_i\right)=\sup_{i\in I} Pos(A_i) \quad \text{(B.13)}$$

Necessity is almost defined similarly but meets slightly different axiomatic requirements:

$$Nec : P(X)\rightarrow[0,1] \quad \text{(B.14)}$$

with the axioms:

$$Nec(\varnothing)=0 \ \wedge \ Nec(X)=1$$
$$Nec\left(\bigcap_{i\in I} A_i\right)=\inf_{i\in I} Nec(A_i) \quad \text{(B.15)}$$

Both the possibility measure and necessity measures thus respectively represent a lower and upper bound for the possibility an event x in X given a body of evidence. Possibility theory is a special branch of evidence theory (Section B-5) [69]. Compared to evidence theory it only deals with a bodies of evidence whose focal elements are nested [71] [72]. Despite possibility theory is considered by many as a natural way to express uncertainty there are only few practical applications of it described in literature [97].

B-5 Evidence Theory

Evidence theory, also called Dempster-Shafer theory named after both creators, can be considered as a more general form of the previously discussed probability theory [72] [74] [97] [134]. The major difference between both is that evidence theory provides two measures, belief and plausibility, to specify the uncertainty or likelihood of an event. In this sense evidence theory shares some similarity with possibility theory (Section B-4). A measure of believe for a given subset x of a universal set X provides an lower bound for the possibility of x and is defined by the next function:

$$Bel : P(x)\rightarrow[0,1] \quad \text{(B.16)}$$

for which should hold $Bel(\varnothing)=0$ and $Bel(X)=1$. Moreover, the believe measure meets the following in equality for each possible subset family A_i of X:

$$Bel(A_1 \cup A_2 \cup \ldots \cup A_n) \geq \sum_j Bel(A_j) - \sum_{j<k} Bel(A_j \cap A_k) + \ldots + (-1)^{n+1} Bel(A_1 \cap A_2 \cap \ldots \cap A_n) \quad \text{(B.17)}$$

The dual measure for belief is plausibility, which defines a higher bound on the possibility of $x \in X$. In other words plausibility is the largest possible probability for x given the available or known evidence and is defined as:

$$Pl(x) = 1 - Bel(x^C) \tag{B.18}$$

This expression also implies that the plausibility in the occurrence of an event plus the belief against this event must sum unity. Furthermore, the plausibility measure meets the following in equality for each possible subset family A_i of X:

$$\begin{aligned} Pl(A_1 \cap A_2 \cap ... \cap A_n) \geq \sum_j Pl(A_j) - \sum_{j<k} Pl(A_j \cup A_k) \\ + ... + (-1)^{n+1} Pl(A_1 \cup A_2 \cup ... \cup A_n) \end{aligned} \tag{B.19}$$

From *Expressions B.17* and *B.19* it can be deduced that compared to probability theory (*Expression B.8*) the evidential measures *Pl* and *Bel* for the occurrence of an event and its negation don't have to sum up to unity, or absolute certainty:

$$\begin{aligned} Bel(x) + Bel(x^C) \leq 1 \\ Pl(x) + Pl(x^C) \geq 1 \end{aligned} \tag{B.20}$$

According to *Expressions B-16* and *B-18* both believe and plausibility are defined by a function that is commonly known as *basic probability function* or *m* [72] [134]. This function expresses the degree of probability or likelihood that supports the evidential claim that true alternative (prediction, diagnosis, etc.) is the subset x of X, but not in any particular subset of x. Given this *basic probability function m* both belief and plausibility can be determined as follows:

$$\begin{aligned} Bel(x) = \sum\nolimits_{y \subseteq x} m(y) \\ Pl(x) = \sum\nolimits_{y \cap x \neq \emptyset} m(y) \end{aligned} \tag{B.21}$$

In the context of uncertainty specification using evidence theory, *Bel(x)* is the likelihood or probability that is known with certainty, given the evidence associated with the event x [97]. *Pl(x)* is considered as the maximum likelihood or probability that could potentially, given the evidence, be associated with an event x. Since the basic probability function *m* can be arbitrarily constructed as long as it complies with the in these section mentioned axioms, Evidence theory suffers from the same strength and weaknesses as probability theory (Section B-2). The major advantage of evidence theory over probability theory is the fact that it can handle the frequently encounter situation where a precise single valued or crisp probability cannot be assigned based on the available evidence. In those case there does however exist a range of probabilities, which is consistent with this given evidence. Practical examples of how evidence theory can be used to model and assess uncertainty are found in [98] [115].

Appendix C: Case-Studies Background

This appendix covers the background details of the following two aerospace simulation projects, which have been used as case-studies to test, refine and illustrate the unified fidelity framework developed in this thesis:

- *Future Airspace Simulation Environment (FASE) Project*
- *IAe/IPTN CN235-220 Aircraft Simulation Model Project*

The author of this thesis has been the principle modeling and simulation engineer in both these two simulation projects. Each simulation system and models developed in these projects will be presented in the next two sections.

C1 Future Airspace Simulation Environment (FASE) Project

The presentation below is a summary drawn from two previously published technical papers on the FASE project [118][123].

FASE Background & Objectives

Modernization of existing airspace systems requires numerous new Air-Traffic Control and Management (ATC/ATM) concepts to be developed and evaluated for their improvements in air-transportation efficiency and safety. Most of the research in this domain is conducted by means simulation systems. Although all these new ATC/ATM concepts rely on increased interactions between the airspace users, the studies executed at the present time rely mainly on the usage of stand-alone simulations. This means aspects of the air navigation system are often investigated separately on optimized independent simulation systems neglecting the crucial integrative and interaction aspect. In addition, their individual stand-alone operational character limits the efficient (re)-use of existing simulation systems. Therefore, to able to properly evaluate the improvements of new ATC/ATM concepts and systems it is necessary to integrate individually developed simulation systems within a single representative operational context. It is also desirable to have the capability of integrating such simulation systems, which may only available on specific geographical distributed locations.

To support their ATC/ATM research programs the Delft University of Technology Aerospace Control & Simulation (DUT-AC&S) division initiated the development of a new distributed simulation infrastructure called Future Airspace Simulation Environment (FASE). The underlying FASE project objective is to provide a distributed simulation environment, which allows various types of simulations to interoperate in order to create a set of representative civil airspace systems of varying scale and complexity. The types of simulations that have to interoperate in this environment range from simple air-traffic flow generating simulations, desktop flight simulators, pseudo-pilot stations, to research air-traffic control simulators and full flight simulators. Furthermore, FASE is to prepare the integration of the DUT-AC&S division available simulation facilities into a single ATC/ATM research environment in combination with simulation facilities of institutes like EUROCONTROL, SIMONA and TNO-FEL.

FASE Architectural Design

The FASE simulation system concept is based on a minimal set of real-world entities and their interactions necessary to properly simulate a wide range of future civil airspace systems. These entities and processes are grouped into three functional layers: air-entity layer, environment interlayer and ground entity layer (*Figure C-1*). The air-entity layer replicates all aircraft operating in a civil airspace. This layer is implemented by three kinds of aircraft simulators: autonomous computer generated air-traffic, pseudo-pilot controlled air-traffic, and human-in-loop flight simulators. The ground entity layer replicates all ground based air-traffic control and management aspects of civil airspace. It implements simulators like Tower and Air-Traffic-Controller working positions as well as all necessary tools for conflict resolution, strategic flight environment interlayer is the interconnection layer between ground and air entity layer. In here the real-world communication, navigation and surveillance systems are replicated together with meteorological phenomena.

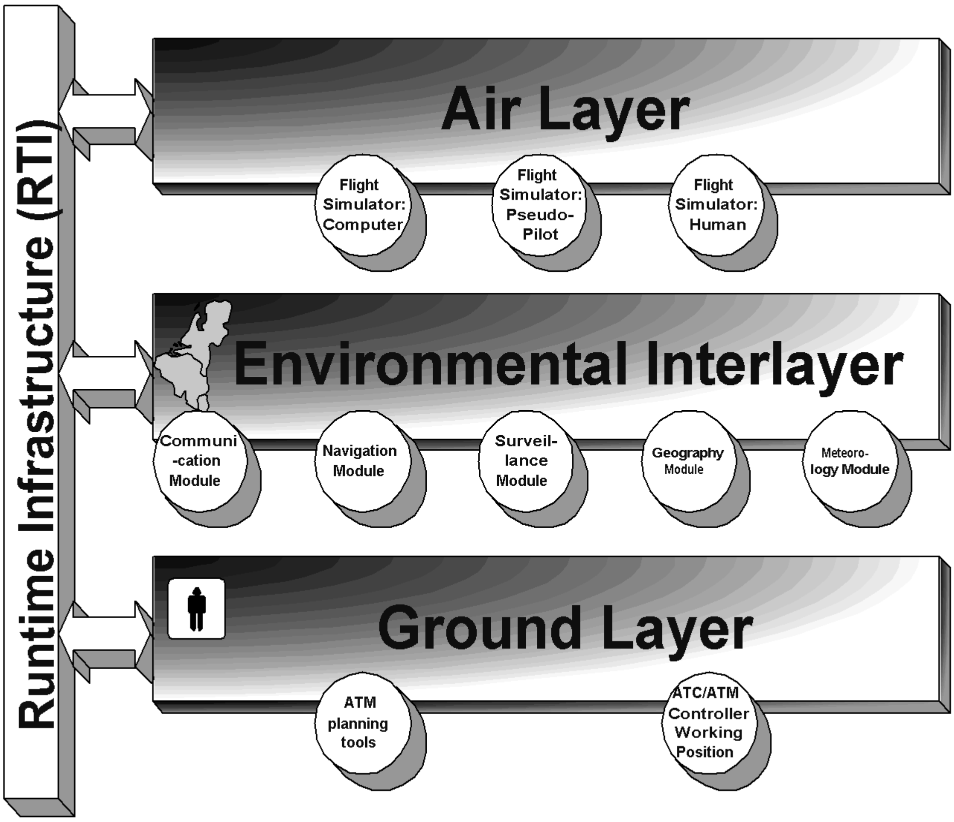


Figure C-1 FASE Functional Layer Design

The architectural design of FASE is build upon the High Level Architecture (HLA). HLA is the IEEE standard for distributed simulation, which promotes the interoperability and reuse of physically distributed simulation facilities and components. A typical HLA application is referred to as a *federation*. It consists of a number of physically distributed simulations called *federates*, which communicate with each other through a physical interconnection layer known as the *runtime infrastructure* (RTI) [17]. To be able to participate in a federation, federates must obey certain design rules, use a standardized interface protocol and be able to share all federation relevant data in a standard HLA format or *FOM* [15][16]. This FOM describes all data to be shared and intercommunicated between all federates during the execution of a federation, and is based on the information a federate can make available and requires when participating in a federation. The FASE three-layer simulation concept has been translated into a generic FASE Federation design comprising eleven generic FASE federates and associated FASE Reference FOM. These generic FASE federates can be instantiated and configured accordingly to create multiple future civil airspace system federations.

FASE Implementation

The current implementation of the FASE architectural design comprises the next operational federates:

- *Air-Traffic Server*. A simulator capable of generating large amounts of computer generated aircraft. The aircraft flight dynamics engine is an implementation of the EUROCONTROL BADA model, and provides performance, operational and procedural data for over 150 aircraft types.
- *Pseudo Pilot Station*. A simulator that enables a human or pseudo-pilot to control multiple aircraft at the same time by means of an airspace plan view, aircraft flight management and control computer interface.
- *Desktop Research Flight Simulator*. A pilot controlled B747-400 desktop flight simulator with 6-dof non-linear flight dynamics and future avionics systems like the Airborne Separation Assurance System.
- *STANS*. An existing EUROCONTROL ATM simulator that has been made FASE compliant by DUT-AC&S. The STANS platform can supply air-traffic that is recorded by the Central Floor Management Unit of EUROCONTROL, allowing real-life situational scenarios and live air-traffic data within FASE.
- *CNS Server*. A simulator providing simulation of Communication, Navigation and Surveillance systems available around the globe. It utilizes a database with more than 17,000 entries for VOR, DME, NDB, ILS beacons, primary and secondary radar systems, and GPS.
- *ATCO Working Station*. A simulator that replicates a combined tower and air-traffic controller working position. This simulator features radar displays using the output of the available radar stations, flight-plan processing and a tower outside visual.

The FASE federates are implemented on LINUX and Windows 2000 platforms and can communicate with each other through the HLA RTI over the internet or any other dedicated computer network. This allows for flexible and cross-platform distributed simulations. Depending on the type of experiment FASE can be executed in a real-time or non-real-time modus. Radiotelephony within FASE is emulated by means of freely available Voice-over-IP software, allowing chatter between all human players. A self-developed patch for Microsoft Flight Simulator allows communication with HLA environments and thus can participate in FASE.

With current FASE implementation various federation configurations have been experimented with to test its performance, robustness, and functional capabilities. The scenarios tested varied in complexity and scale but all focused on aspects of the 'Free-Flight' paradigm implementation within the en-route airspace and terminal area around airports. Experiments showed that FASE is capable of simulating representative high-density traffic air spaces using the STANS simulator and multiple instances of the Air-Traffic Server. An important feature needed for studying the overall airspace efficiency and safety of new ATC/ATM concepts. The use of two pilot-in-the-loop simulators and the ATCO working station demonstrated that FASE is also capable to accommodate research to human-factor aspects of future airspace systems from both the cockpit and air-traffic controller perspective.

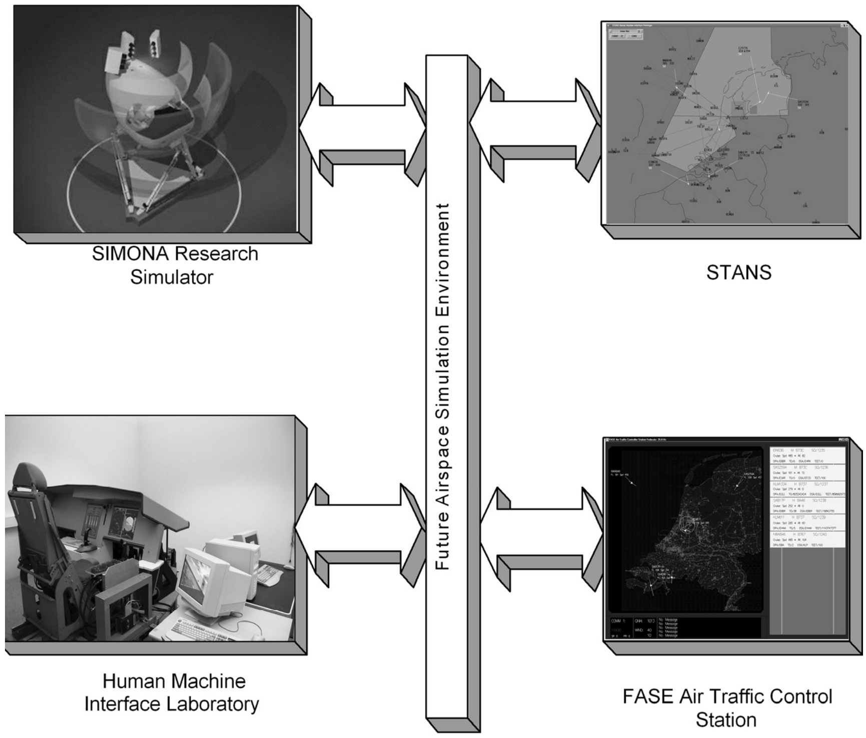


Figure C-3 A Typical FASE Federation Configuration Setting

C2 IAe/IPTN CN235-220 Aircraft Simulation Model Project

The presentation below is a summary drawn from the series of seven technical reports on the CN235-220 Aircraft Simulation Model (ASM) project. Due to the confidentiality of certain information and implementation details, only a high-level and general presentation can be given here.

CN235-220 ASM Background & Objectives

Indonesian Aerospace (IAe) is one of the indigenous aerospace companies in Asia with core competence in aircraft design, development and manufacturing of civilian and military regional commuter aircraft. IAe offers a wide range of different aircraft types. One of these aircraft is the CN-235-220, which is a multiple commuter and utility aircraft (*Figure C-4*). This multipurpose aircraft is also provided in a military version, M-version, which is designed to fulfill the requirements of all light military cargo and troop transportation operations.

To serve the needs of their costumers the Technology & Engineering Services Division of IAe has taken the initiative to design and develop a CN-235-220 full flight simulator for both civil and military training purposes. The civil version of the simulator has to comply with the FAA Airplane Simulator Qualification level D requirements. These requirements are stated in Advisory Circulation AC120-40B [35]. The military version of the simulator has also to comply with a set of additional qualification requirements concerning specific military aircraft operations, not covered by the FAA Advisory Circulation AC120-40B. These include additional requirements concerning the effect of paratroop dropping, aerial delivery system, and low altitude parachute extracting system effects on the flight dynamics.

Figure C-4 A Military and Civil Version of the CN235-220

Following the tradition of long-time cooperation with IAe and substantial experience in the field of flight simulation technologies and mathematical model building, IAe has subcontracted DUT-AC&S to assist in the development of the CN-235-220 full flight simulator. One of the working packages assigned to DUT-AC&S has been the improvement of the existing aircraft simulation model and validating this model according the Advisory Circulation AC120-40B.

CN235-220 ASM Architectural Design

The basis for the aircraft simulation model design is based on existing generic object oriented aircraft simulation model architecture developed and refined in previous aircraft simulation projects [118] [127] [128]. This design is based on a set of general applicable functional and non-functional requirements for aircraft simulation models. When applied to the CN235-220 aircraft this results in the following real-world system models that form the aircraft simulation model:

1. *CN235-220 Aerodynamic Model*: Among many other things this model reenacts the aerodynamic effects due to aircraft body movement, control surface deflections, flaps, spoilers, gear position, ramp door position, propeller wash and ground proximity.
2. *CN235-220 Structural Mass and Inertia Model*: Replicates the aircraft mass and inertia properties due to contributions such as airframe structure, fuel quantity, gear position, ice forming and different payloads.
3. *CN235-220 Propulsion System Model*: Is an aggregation of two other real-world sub models replicating the GE-CT7-9c turbo-prop engine dynamics and Hamilton Standard 14RF-21 propeller characteristics.
4. *CN235-220 Flight Control System Model*: Represents the pilot control input means and dynamics for elevator, rudder, aileron, aileron trim, elevator trim, gear, flaps and spoilers.
5. *CN235-220 Undercarriage Model:* This model includes the replication of the undercarriage dynamics, forces and moments for various runway conditions and surfaces.
6. *CN235-220 Flight Dynamics Model*: Is responsible for the replication of the aircraft rigid body dynamics and kinematics.
7. *Atmospheric Model*: Represents the atmospheric conditions around the aircraft such as air density, pressure, wind profiles and turbulence.

Based upon the actual aircraft design and development documentation, profound mathematical formalisms and algorithms have been developed for each of these real-world system models. In here system identification methods and advanced fuzzy-logic methods have been used to analyze and construct proper parametric model data available from various sources such as wind tunnel experiments, flight-tests and the manufactures engine and propeller performance decks.

CN235-220 ASM Implementation
This architectural design of the model has been translated, using the Unified Modeling Language, into an object-oriented software implementation into real-time ANSI C/C+ implementation. For the validation of the simulation model the following three tools have been used:

- *DUT-AC&S POM-Tool Suite*: A dedicated build validation tool capable of performing FAA AC120-40B compliant validation test or commonly known as Proof of Match (POM).
- *DUT-AC&S Desktop Research Flight Simulator*: A low-cost generic and modular designed real-time flight simulator running on a personal desktop computer. It uses simple game-control devices for generating pilot inputs and offers logging facilities to register any
- *SIMONA Institute Research Simulator*: A full six-degree of freedom full flight research simulator offering a high performance motion system, sound generation and a visual display system with a 180° by 40° field of view. It has a fully instrumented and generic 2-seat aircraft flight deck with hydraulically loaded controls for additional realism in pilot control force representations.

Figure C-5 The DUT SIMONA Institute Research Simulator

In the early stages of the aircraft simulation model development the DUT-AC&S desktop simulator served as the principle tool for verifying and validating the respective real-world system models. Later on in the project, when the aircraft simulation model became more mature, both the POM tool and the SIMONA research simulator were extensively used to validate the simulated aircraft behavior and to identify sources for large discrepancies with respect to the real aircraft. Finally, the SIMONA research simulator was used to do the initial FAA AC120-40B prescribed subjective pilot tests with actual CN2350-220 pilots.

Acknowledgements

Pursuing a Ph.D. is an uncertain and therefore hard to manage activity. It is an activity whose progression not only depends on personal but also on many uncontrollable external factors that either obstruct or accelerate the whole research process. Over the past years I have experienced the negative side effects such activity can have on the human body both physically and mentally. Thanks to two constant factors in my life I managed to reach this milestone in life.

The first factor, above all, are my parents with their unconditional support in all the things I try to pursue in life. One thing is for sure without them I definitely would not have accomplished this Ph.D. thesis. The second factor is practicing and teaching Martial Arts, which kept me physically and mentally in good shape. Especially, I would like to thank my training mates Serge Maurer and Bert van Breugel for their many hours on the mat, and who kicked me over and tried to blend me with the mat when this was necessary after a frustrating day of work.

Without the financial and indispensable technical support of TNO-FEL this Ph.D. thesis would never have seen the light. I own many thanks to Hans Jense for providing me this Ph.D. research opportunity in very close cooperation with his TNO-FEL simulator division. Former TNO researcher Paul van Gool, alias *Dr G*, taught me almost everything I had to known about distributed simulation technology and software engineering during the first two years of my Ph.D. research. His successor, Jeroen Voogd has provided me with so much critical and thorough reviews of all my ideas and the final manuscript, that this thesis would not be the same without them.

Obviously a good and pleasant working environment at the Delft University of Technology is a precondition for successfully completing a Ph.D. thesis. Thanks goes to my promoter, Bob Mulder, for his tireless enthusiasm and optimism during this Ph.D. project and other projects I conducted for his division. With pleasure I have shared a room with Rene van Paassen, Control and Simulation division's legendary living knowledge base. Furthermore, I would like to thank all my former university colleagues. Especially, the other *sim-squad* members Olaf Stroosma, Erwin Kipperman and Walter Berkouwer with whom I tackled many technical and equally important non-technical issues, which are an inextricably part of conducting research at the Delft University of Technology. Not to mention the great fun we had during and even more after office hours. Over the past years I mentored several students during their graduation work. I particularly enjoyed the teamwork and great philosophical discussions about almost everything in life with Stephen Fer, my co-inventor of the Future Airspace Simulation Environment. Luckily, we are still in touch with each other.

I also owe much gratitude to Kees-Jan Leliveld of Nyquist Industrial control for his patience and offering me the space to finish the last bits and bytes of this thesis, even during office hours.

Last but certainly not the least, many thanks to Tjepke Heeringa my fellow student and best friend from day one when we both started studying at the Delft University of Technology. I'm proud he accepted my invitation and traveled all the way from the Lockheed-Martin premises at Ft. Worth Texas to serve as my *paranimf* during the defense of this Ph.D. thesis.

Manfred Roza
December 2004

About the Author

Manfred Roza was born on April 21 1973, in Dussen, the Netherlands. From 1985 to 1991 he attended the Dongemond College in Raamsdonksveer, where he obtain the Atheneum certificate.

In 1991 he enrolled as a student at the Faculty of Aerospace Engineering of the Delft University of Technology. He started to study aerospace modeling and simulation technologies under the guidance of dr. ir. P. van Gool. Together with three other students he conducted his M.Sc. thesis research to the development and implementation of a generic object-oriented framework and simulation environment for the simulation of dynamic entities. The results of this work formed a base for the current simulator environment and aircraft simulation model architecture currently utilized at the Delft University of Technology. He obtained his M.Sc. degree in 1996.

After a short stay abroad he joined the DUT Control and Simulation Division as a Ph.D. student in 1997, where he extended his research to modeling and simulation technologies in the area of distributed simulation applications and validation techniques. The Netherlands Organization for Applied Scientific Research Physics and Electronics Laboratory (TNO-FEL) sponsored this research and he also worked there for a period of four years as an external researcher in that same area.

He has obtained extensive experience in HLA-based distributed simulation development and implementation in various multidisciplinary projects such as SIMULTAAN and DUT Future Airspace Simulation Environment (FASE). During the period 1998 up to 2000, he served as one of the lead-team members of the SISO Fidelity Study Groups involved in the development and application of a fidelity taxonomy and fidelity framework for HLA-based distributed simulations.

From 2001 until 2004 he has fulfilled the role as a principle modeling and simulation engineer at DUT Control and Simulation Division, where he contributed to the development of complex aircraft models and simulation systems for external costumers such as Eclipse Aviation and Indonesian Aerospace (IAe). Furthermore, during his research work at the Delft University he mentored and examined many M.Sc. students, and several times assisted in lecturing the Aerospace Simulation Technologies course.

Currently the author works for the Nyquist Industrial control company as a software engineer motion control applications and tools in the NYCe-4000 program.

Samenvatting

Fidelity vormt een intrinsiek element van ieder simulatie systeem, iets waarmee alle zijn ontwikkelaars en gebruikers op diverse manieren mee te maken krijgen. Het wordt algemeen beaamd in de modelvorming en simulatie wereld dat fidelity van simulaties een essentieel middel is in het beoordelen van de geldigheid en geloofwaardigheid van simulatieresultaten. Daarnaast is het een vaststaand feit dat fidelity een van de belangrijkste kost bepalende factoren is binnen het ontwikkelingstraject van simulatie systemen. Simulaties spelen een steeds grotere rol in onze samenleving, en is in hoog tempo het belangrijkste instrument aan het worden bij the maken van kritische beslissingen tijdens het ontwerpen, testen en evalueren van nieuw, vaak zelfs safety-critical, systemen en bij de training van de gebruiker van deze systemen. Met dit toenemende belang en afhankelijkheid van simulatiesystemen wordt het meer dan ooit belangrijk te weten hoe goed de simulatie overeen komt met de werkelijkheid om er zeker van te kunnen zijn dat de risico's bij het gebruik van simulatieresultaten zich binnen acceptabele limieten bevind.

Ondanks deze constateringen en de enorme technologische vooruitgang op het gebied van hard en software, blijven onze mogelijkheden om het niveau van realisme te karakteriseren, kwalificeren en kwantificeren een onderontwikkeld gebied. Een gebied met vele incomplete, inconsistente en ver uiteenlopende opvattingen, concepten en benadering met betrekking tot fidelity. Het belangrijkste gebrek is de afwezigheid van een systematische en algemeen toepasbare methodologie voor het beoordelen van fidelity op basis van een gedegen uniforme theorie voor fidelity met bijbehorende praktische werkwijzen.

Dit proefschrift draagt een mogelijke oplossing aan om dit gebrek op te vullen door het analyseren, modificeren en integreren van reeds bestaande fidelity benadering in een enkele uniforme theorie voor fidelity met bijbehorende praktische werkwijzen. Dit alles is gedaan vanuit een algemeen perspectief zonder zich te beperken tot enige specifieke simulatie toepassing en probleem gebieden.

Om een uniforme theorie voor fidelity te kunnen ontwikkelen is een vergelijkend onderzoek tussen reeds bestaande fidelity theorieën en werkwijzen noodzakelijk. Dit proefschrift identificeert de belangrijkste overeenkomsten, verschillen, problemen en beperkingen van een representatieve doorsnede vanuit de literatuur bekende onderzoeksresultaten op het gebied van fidelity. De hieruit voortkomende resultaten vormen een van de pijlers onder het in dit proefschrift ontwikkelde *unified fidelity framework*. Het is haast onmogelijk om zonder een contextueel modelvorming en simulatie raamwerk een uniforme theorie voor fidelity te ontwikkelen welke naadloos te integreren is binnen een simulatie ontwikkel en validatie proces. Daarom beschrijft dit proefschrift een dergelijk raamwerk welk de tweede pijler onder het *unified fidelity framework* vormt.

Het fundament van het ontwikkelde *unified fidelity framework* omvat een precieze formulering voor de term fidelity en de onderliggende concepten voor het karakteriseren en meten van fidelity. Dit alles te samen met het bijbehorende mathematische formalisme. Het belangrijkste concept in hier is het *real-world reference knowledge standard* paradigma. Dit paradigma, afgekort ook wel *fidelity referent* genoemd,

formaliseert de natuurlijke indirectheid van het meten van fidelity. Wat inhoud dat het nooit mogelijk is om fidelity direct tegen de werkelijkheid te meten maar in plaats daarvan meet men altijd tegen een benaderende interpretatie van hoe deze werkelijkheid ogenschijnlijk in elkaar steekt. Door expliciet deze onzekerheden en fouten te koppelen aan de *fidelity referent* wordt het onoplosbare probleem om fidelity exact nauwkeurig te kunnen meten getransformeerd in een praktische, op bewijsvoering gebaseerde methode, om de mate van realisme van een simulatie te kunnen beoordelen. Het andere hierbij behorende element is dat van het *simulation system knowledge specification* concept. Voor elk van deze twee concepten wordt een praktische implementatie voorgesteld in de vorm van een generieke *knowledge-base* architectuur die bestaat uit een gestructureerde kennis specificatie matrix. Deze matrix wordt ondersteund door een set van bijbehorende mathematische formalisme.

Met een formeel gedefinieerde *fidelity referent* en *simulation system knowledge specification* komt het beoordelen van simulatie fidelity praktisch gezien neer op het meten en specificeren van het inverse verschil tussen beide. Omdat fidelity een multi-dimensionaal en multi-facet karakter heeft, wordt fidelity het beste gekwalificeerd en gekwantificeerd doormiddel van een opsomming van verschillende soorten metrieke in plaats van een enkelvoudig gemeten kengetal. Dit proefschrift beschrijft een taxonomie met de meest elementaire en gangbare meetmethodes en metrieken die voor dit doeleinde gebruikt zouden kunnen worden. Deze taxonomie is een combinatie van in dit proefschrift nieuw ontwikkelde en reeds bestaande en bewezen meetmethodes en metrieken zoals in de literatuur beschreven.

Het proefschrift introduceert eveneens het zo genoemde *fidelity requirements* concept, waarmee het mogelijk is om op een formele en systematische manier het vereiste niveau van fidelity te specificeren om te kunnen voldoen aan de gebruikerseisen die voor een bepaalde simulatie toepassing gelden. Op basis van dit concept is een fidelity georiënteerd proces ontwikkeld voor het verifiëren en valideren van simulatiesystemen. Het gebruik van een multi-criteria analyse benadering wordt voorgesteld om hiermee binnen het ontwikkelproces van simulatiesystemen diverse alternatieve keuzemogelijkheden in relatie tot de fidelity performance en effectiviteit van de uiteindelijke resulterende simulatie te kunnen evalueren en vergelijken.

Het laatste belangrijke element van het ontwikkelde *unified fidelity framework* is het fidelity management procesmodel. Dit procesmodel bestaat uit een reeks van generieke stadia, activiteiten en taken die samen een gestructureerd stappenplan beschrijven om alle voorgaande elementen van het *unified fidelity framework* op een nette en systematische manier toe te passen binnen het simulatiesysteem ontwikkel en validatie proces. Het gebruik van dit fidelity management proces model is in de praktijk uitgetest aan de hand van twee casestudies uit het toepassingsgebied van de luchtvaart.

Ondanks dat beide casestudies beperkt zijn in hun omvang en diepte tonen deze studies wel aan dat het *unified fidelity framework* en zijn onderliggende concepten en paradigma's een veel belovende en levensvatbare basis vormen voor de toekomstige ontwikkeling van een noodzakelijke standaard op het gebied van simulatie fidelity theorie en met bijbehorende praktische werkwijzen. De belangrijkste ondervonden voordelen bij deze casestudies omvatten een beter definitie wat, hoe en wanneer fidelity evaluatie activiteiten moeten worden uitgevoerd en het leveren van duidelijker gedefinieerde simulatie systeemeisen die het eenvoudiger maken om keuzes en prioriteiten te stellen in het ontwikkel proces. Daarnaast, levert het *unified fidelity framework* een positieve bijdrage aan de efficiëntie waarmee kennis van het werkelijke

systeem en de bijbehorende simulatie te kunnen vergaren en te structureren. Verder, wordt het eenvoudiger om op een systematische manier oorzaken op te sporen binnen een simulatiesysteem die verantwoordelijk zijn voor onacceptabele grote fidelity discrepanties en een passende oplossing hiervoor te vinden.
Volgens de wet van Murphy komt ieder voordeel ook met een aantal nadelen. Het belangrijkste ondervonden minpunt wordt veroorzaakt door het inherent multi-dimensionale en gefaceteerde karakter van fidelity. In de praktijk leidt dit er toe dat fidelity evaluatie een complexe, tijdrovende en moeilijk met de hand uitvoerbare activiteit wordt. Iets waaraan geen andere fidelity methodologie ook aan zal kunnen ontkomen. Het is dan ook van vitaal belang dat er algemeen en/of domein specifiek toepasbare geautomatiseerde tools worden ontwikkeld om simulatieontwikkelaars en validatie personeel hierbij te ondersteunen. Zonder dergelijk tools blijft de toepassing van de formele en systematische fidelity methodieken binnen de modelvorming en simulatie wereld, ondanks hun noodzaak, economisch gezien moeilijk realiseerbaar.

Rigoureuze en accurate beoordeling van fidelity is een van de meest lastige en moeilijkst vast te grijpen kwesties binnen de modelvorming en simulatie wereld. Substantiële en grondige onderzoeksinspanningen en resultaten op dit gebied zijn erg beperkt. Hierdoor is simulatie fidelity nog steeds een nauwelijks aangeroerd en onderontwikkeld gebied. Vanuit dit perspectief gezien ligt de belangrijkste bijdrage van dit proefschrift aan de modelvorming en simulatie wereld in het feit dat het alle mogelijke aspecten van simulatie fidelity samen worden gebracht in een eenduidige formele fidelity theorie met bijbehorende praktische werkwijzen. Daarnaast legt het in dit proefschrift gepresenteerde *unified fidelity framework* een aantal essentiële fundamenten voor de ontwikkeling van een algemene standaard voor fidelity. Dit alles maakt dit proefschrift uniek in zijn soort welke de voorheen gesloten deur op een kier zet om een aantal grote stappen voorwaarts te kunnen maken om een dergelijke noodzakelijke standaard daadwerkelijk te realiseren.