Leveraging Fidelity to Achieve Substantive Interoperability

S. Y. Harmon
Zetetix
P.O. Box 2640, Agoura, CA 91376
(818) 991-0480
harmon@zetetix.com

S. M. Youngblood
Defense Modeling and Simulation Office
1901 N. Beauregard Street, Alexandria, VA 22311
(703) 824-3436
syoungblood@dmso.mil

Keywords:
interoperability, fidelity, federation, validation

ABSTRACT: *This paper introduces a mathematically rigorous approach for determining when two distributed simulations can achieve sufficient substantive interoperability to validly serve a specific purpose. This approach distinguishes between two situations that can lead to interoperability problems, functional dependencies and manifold representations. Functional dependencies occur when the result produced by one simulation requires input from another simulation. A common example of a functional dependency is a line-of-sight algorithm that depends upon a terrain model to determine if one entity can see another. Manifold representations occur when two or more interdependent simulations represent the same or related properties of the same entities. Dead reckoning is a common example of manifold representations. The authors have derived from the SISO fidelity framework, proposed by Gross et al., a set of criteria that distributed simulations must meet to interoperate substantively in one or both of these situations. This derivation has also yielded precise definitions of several classes of simulation interoperability anomalies that can occur. Operating distributed simulations outside these criteria introduce the probability that they will manifest one or more of these anomalies. The information needed to test these criteria can come from several different sources including the participating simulations' SOMs. With information in addition to that provided by the SOMs, this approach can locate the specific conditions under which interoperability anomalies are likely to occur and the particular simulation components from which those anomalies arise. This paper describes this additional information and its impact upon the completeness of validation results. This approach takes important steps towards further refining the concept of substantive interoperability first suggested by Dahmann et al.*

1. Introduction

Dahmann et al. [1] introduced the distinction between technical and substantive interoperability as a means of identifying the limits of simulation interoperability supported by the current HLA infrastructure. In their discussion they suggested the following definitions for these terms:

technical interoperability - the capability of federates (e.g., simulations) to physically connect and exchange data through those connections.

substantive interoperability - the capability of federates, when connected, to provide adequate, accurate and consistent simulated representations that adhere to the principles of "fair fight" and address the mission objectives.

They note that the HLA supports the technical interoperability between distributed simulations. Substantive interoperability, on the other hand, remains a key challenge to achieving complete distributed simulation interoperability. The discussion herein builds upon the work described by Dahmann et al. by recommending rigorously defined criteria for identifying substantive interoperability problems.

This discussion uses two additional terms, simuland and referent, in describing these criteria. Definitions for these terms are provided below to assist the reader in understanding their use herein:

simuland - the phenomena or system being simulated by a simulation.

referent - a codified body of knowledge about the thing being simulated.

This paper distinguishes between the interoperability problems arising from two distinct situations, functional dependencies and manifold representations, and suggests several criteria that must be satisfied to avoid each of these problem areas. This paper goes further by identifying those parts of the simulation object model (SOM) that can contribute to assessing the suggested criteria for a federation and suggests extensions to the current object model template (OMT) specification that would enable more automated analysis for substantive interoperability.

2. Substantive Interoperability Criteria

The SISO fidelity conceptual framework [2] describes the components of simulation fidelity. This paper builds from that framework by using these components as the elements of criteria that identify potential substantive interoperability problems. Development of these criteria requires precisely defining the meaning of substantive interoperability by characterizing particular observable anomalies that arise from interoperability problems. With these anomalies defined, this paper describes the specific criteria for both functional dependencies and manifold representations. Finally, these criteria are compared against the information provided in the SOM to identify current opportunities for testing federations for substantive interoperability problems.

Forming a federation explicitly links the federates technically through the HLA infrastructure but also implicitly links the entities represented by these federates logically through the interactions that infrastructure supports. To produce meaningful simulation results, the entities represented across the federation must work together in a manner consistent with the needs of the federation application. In effect, creating an HLA federation creates an end-to-end model by reusing selected representations from the participating federates. The information contained by the federation object model (FOM) defines the exchanges between the federates but the entity representations and their combined behaviors define the new simulation application supported by the federation Resolving technical interoperability issues insures that the federation will execute but does not guarantee that that execution will adequately accomplish the federation's mission [1].

2.1 Representational Anomalies

In this paper, anomalies represent those situations where a simulation's behavior deviates to an unacceptable degree from the behavior of the actual objects that simulation represents (i.e., the simulands). In other words, representational anomalies are states and events that would not occur in the simuland under identical conditions. These anomalies result because a simulation has omitted or incorrectly represented, in its abstractions, some aspects of object coupling that exist in the physical world.

State Error Anomalies. Object state error anomalies occur when there exists a difference between the state a simulated object assumes and the state that object's referent assumes under identical conditions and that difference is beyond levels tolerable by the application [2]. Positional inaccuracies in the trajectories of moving objects due to approximations of the gravitational constant or friction coefficients are examples of state error anomalies.

Event Ordering Anomalies. Event ordering anomalies occur when a simulated object produces the same events that the simuland would under identical conditions but in a different order. A detection algorithm with a stochastic component acquiring a more distant target before acquiring a nearer target represents one example of an event ordering anomaly. The substitution of stochastic representations for their much more complex deterministic simulands often results in many event ordering anomalies.

Event Phase Anomalies. Event phase anomalies occur when a simulated object produces the same events in the same order that the simuland would under identical conditions but with a timing or phase error. This sort of anomaly is analogous to state error anomalies but along the time axis rather than along the state value axis. Inaccurate decision times due to approximations of decision latencies could lead to one example of event phase anomalies.

Registration Anomalies. Object state registration anomalies occur when the simulated states of two coupled objects differ from what the states of their coupled simulands would under the same conditions. Registration anomalies are related to state error anomalies but can occur even if the absolute errors of the two object simulations are small. Registration anomalies occur frequently when a simulation maintains multiple representations of the same object (e.g., terrain).

2.2 Functional Dependencies

Functional dependencies occur when the computation of one or more object states in one simulation depend upon the data provided by another simulation. For example, a functional dependency exists between a vehicle simulation and an environment simulation when vehicle fuel consumption depends upon terrain slope and surface conditions as well as vehicle speed. A weather simulation that generates wind speed data for an aircraft dynamics simulator is another example of a functional dependency. Figure 1 illustrates this situation.

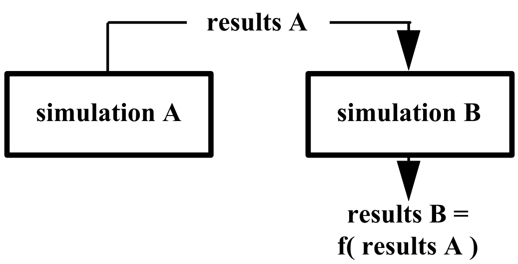


Figure 1. Functional Dependency between Two Simulations.

The occurrence of representational anomalies in simulations but not in their corresponding simulands indicates that some constraints exist in those simulands that the simulations do not enforce. This reasoning suggests that some criteria may exist through which to identify those constraints and when the interacting simulations defy those constraints. Reference [3] formally derives several of these criteria for functional dependencies. This paper only summarizes, in plain language, these interoperability criteria.

Dependency Representation. The first and most important criterion is that one or both of the simulations should represent a functional dependency where one exists and is relevant to the simulations' purpose (i.e., the f(results A) is computed). That dependency should also represent those independent and dependent variables that are relevant. For example, if simulation A is a terrain elevation model and simulation B is a vehicle model that includes a fuel consumption model then a representational anomaly may occur if either the fuel consumption model does not depend upon terrain slope or if the terrain model does not supply terrain slope to the vehicle model.

Representational Accuracy. The next criterion requires that the represented dependency produces results that are within the desired range of accuracy for the application (i.e., the f(results A) is within the desired error tolerance). This criterion captures the need for two interacting simulations to exchange information so as to preserve the accuracy of their results. This criterion implicitly encompasses the problems associated with mapping the output of one function to the input of another (e.g., unit consistency, coordinate system consistency). For example, if simulation A provides a distance measure in units of feet and simulation B expects that measure in units of meters then simulation B will likely manifest a representational anomaly when it uses the data simulation A supplies. Any interaction between simulations A and B that cause the accuracy of B's representation to deviate beyond that required falls under this criterion.

Range Consistency. The SISO fidelity conceptual framework [2] suggests that all of the functions that a simulation represents have finite ranges and domains that result from the abstraction process. This observation leads to the following criterion. The range of the function that produces results A must map completely into the domain of the function that produces results B. If the values of results A fall outside the domain of the function f(results A) used in computing results B then those values could produce state errors. For example, if simulation A is a weather model that generates wind speed between 0 and 150 knots and simulation B is an aircraft model that accepts wind speed information but only from 0 to 50 knots then simulation B may manifest a representational anomaly when the wind speed exceeds 50 knots.

Stochastic Consistency. If simulation A represents a process stochastically and produces results A within some deviation then the range of those results plus two times that deviation must map completely into the domain of the function that produces results B. This criterion guarantees that if the function producing results B receives input from a function with a stochastic component then that input will always be within the acceptable domain despite the effects of the random variations. For example, if simulation A computes the number of rounds fired at a target with a stochastic algorithm that has a range of 0 to 100 rounds with a standard deviation of 10 rounds and simulation B computes the damage a target suffers from this input with an algorithm that has a domain of 0 to 100 rounds then simulation A could generate a result of 105 rounds thus exceeding simulation B's acceptable domain and creating an anomaly. If the range of results A completely account for these stochastic effects and for the stochastic effects of the inputs to the function producing results A then only the range consistency criterion need be applied.

Sensitivity Consistency. The SISO fidelity conceptual framework suggests characterizing simulation functions by their sensitivity to their independent variables and the precision of their dependent variables [2]. This characterization leads to the criterion that the precision of the results produced by simulation A must be greater than or equal to the sensitivity of results B to results A. If the results A change by a smaller amount than the sensitivity of the function f(results A) then the results B will not change until the values held by results A change enough to be reflected in results B. This latency could lead to event ordering anomalies. For example, if simulation A represents terrain elevation with a 2 meter precision and simulation B computes the location of forces employing cover and concealment techniques with a sensitivity of 1 meter then simulation B will likely find many fewer opportunities to distribute forces geographically and thus manifest a representational anomaly.

Temporal Representation. The smallest time interval to which simulations A and B are sensitive must be less than or equal to the smallest time interval within which the physical processes they represent can change state significantly (where the sensitivity of the purpose to those changes defines significance). This criterion simply states that the interacting simulations must both represent time with fine enough granularity so that they capture the events of interest to the application. For example, if simulations A and B represent an air engagement then their smallest time interval should be chosen so that the missiles can travel less than a target's width otherwise these simulations will manifest the common anomaly of missiles flying through targets without damaging them.

Interval Sensitivity. The smallest time interval to which the function f(results A) can respond must be less than or equal to the smallest time interval that simulation A represents when producing results A. This criterion is analogous to the sensitivity consistency criterion except that it applies to the time domain. If the results A can represent time intervals smaller than those to which simulation B is sensitive then results B might not represent a significant event at the proper time or, worse, may not represent that event at all. For example, if a weapons model in simulation A can fire one round per second but a damage model in simulation B can only represent those rounds fired every ten seconds then simulation B may not correctly represent the damage incurred from the weapon that simulation A represents.

Error Consistency. If simulations A and B represent objects with properties measured in intersecting metric spaces then the errors of those properties must be equal where those properties provide values for dependent functions. For example, if simulation A computes the terrain feature location with a 10 meter error but simulation B assumes those features to be located within a 1 meter error then the relative positions of those features may be different in simulation B than they are in simulation A. These differences could lead to representational anomalies in one or the other simulations. This criterion rigorously captures a necessary condition for fair play. If simulation A computes its results from a base of different errors than simulation B assumes then the results B may produce unacceptably inaccurate results, at least when compared to other results produced by simulation A. The difference between the error bases of simulations A and B effectively measures the tilt of the playing field they share. Ironically, this criterion does not identify which simulation will benefit by this difference, just that the possibility for an unfair advantage exists for one or the other.

Table 1 summarizes the functional dependency criteria and the anomaly types to which they apply.

Table 1. Linkage between the Functional Dependency Criteria and the Representational Anomalies.

Functional Dependency Criterion	Affected Representational Anomalies
Dependency Representation	All Four
Representational Accuracy	State Error & Registration
Range Consistency	State Error & Registration
Stochastic Consistency	State Error & Registration
Sensitivity Consistency	Event Ordering
Temporal Representation	Event Ordering & Phase
Interval Sensitivity	Event Ordering & Phase
Error Consistency	Registration

Alas, Table A does not enumerate a mathematically complete list. Other criteria may exist that functional dependencies must satisfy to guarantee their substantive interoperability. Research in this area continues and will be reported when available.

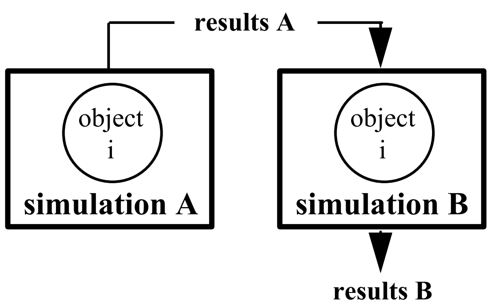
2.3 Manifold Representations

Manifold representations occur when two or more interacting simulations represent the same state or behavior of the same object. Unlike functional dependencies, manifold representations never occur in the physical world except as information. They most often serve as computational conveniences to reduce communications burdens. The dead reckoning that

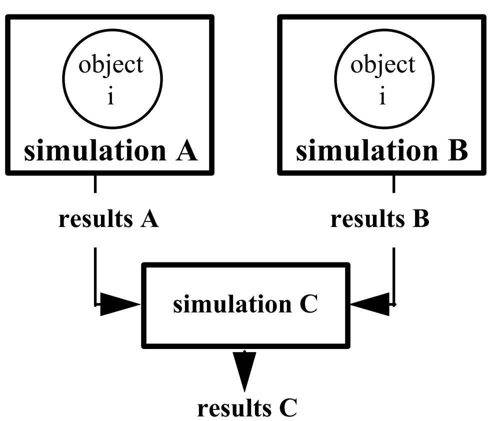
Dahmann et al cite [1] represents one example of manifold representations.

The interoperability problems arising from manifold representations can occur between simulations that interact directly, as illustrated in Figure 2A, or indirectly, as Figure 2B depicts.

Directly interacting simulations may create substantive interoperability problems when they are functionally dependent, as described above. Two simulations interact indirectly when they each produce results used by a third simulation to produce its results. The results from the contributing simulations must meet both the functional dependency criteria and the manifold representation criteria.



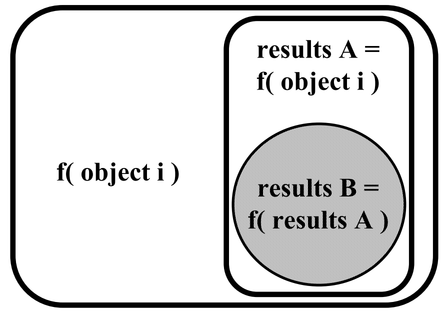
A. Directly Interacting Simulations



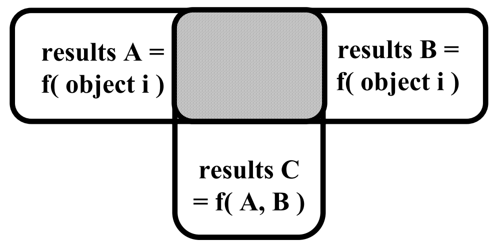
B. Indirectly Interacting Simulations

Figure 2. Manifold Representations in Directly and Indirectly Interacting Simulations.

Manifold representations only pose problems under very specific conditions as shown in Figure 3A for directly interacting simulations and in Figure 3B for indirectly interacting simulations.



A. Conditions for Directly Interacting Simulations

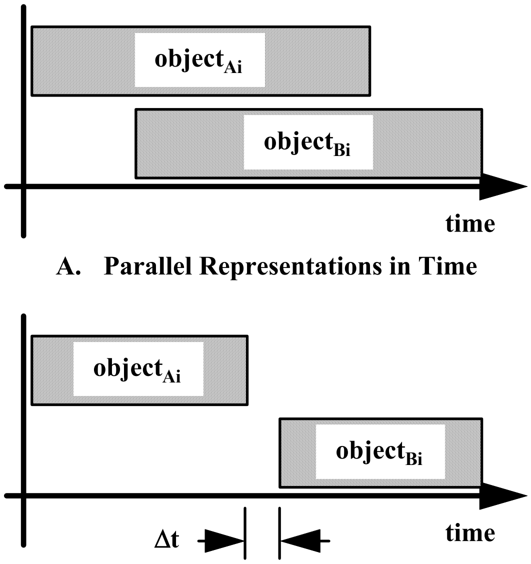


B. Conditions for Indirectly Interacting Simulations

Figure 3. Conditions for Substantive Interoperability Problems in Directly and Indirectly Interacting Simulations.

Substantive interoperability problems arising from manifold representations can occur only under very limited conditions. The problems associated with manifold representations only occur in directly interacting simulations when the results of one simulation depend upon the results, produced by another simulation, that in turn depend upon the state of the manifold represented object. Similarly, these problems occur in indirectly interacting simulations only when the results of the third simulation depend upon the results from both simulations that depend upon the manifold represented object.

Manifold representations need not exist in the simulated world simultaneously as illustrated in Figure 4.



A. Parallel Representations in Time

B. Sequential Representations in Time

Figure 4. Manifold representations in Time.

Figure 4A shows the case where the execution of manifold representations overlaps in time and Figure 4B shows the case where their execution is sequential but, possibly, separated by an interval of Δt. Both of these cases can lead to substantive interoperability problems.

Finally, manifold representations need not have the same levels of abstraction and in many cases do not. For example, a wargame simulation may represent a platoon of tanks as individual platforms in one place or at one time and represent those same tanks as a single aggregated unit at a different place or time. Despite the difference in representation, this still represents a case of manifold representation. Manifold representations at different abstraction levels can pose some of the most challenging interoperability problems to detect.

State Correspondence. Analysis of the errors resulting from manifold representations shows the primary error source to be the lack of agreement between the corresponding attribute values of the different manifold representations [4]. This holds true for both directly and indirectly interacting representations. This criterion requires the difference between manifold object state representations to remain below a tolerable threshold over the entire range of valid interactions. For example, state correspondence errors in the dead reckoned vehicle positions often result in the recognizable jumps shown in visual displays of those vehicles. The sensitivity of the dependent functions (i.e., those that depend upon the information about the manifold representations) to the deviations between the representations defines this tolerable threshold and range of interactions. This criterion also requires the state changes of simultaneously existing manifold representations to correspond within the acceptable tolerances (i.e., synchronization).

Abstraction Transform. Computing the state difference between manifold representations with different levels of abstraction requires the existence of an abstraction transform function. This transform may also need to exist within the functions dependent upon information from multiple manifold representations. This criterion further requires the abstraction transform to remain continuous over the useful domain of the dependent functions. If multiple functions depend upon the same manifold representations then the required continuous domain is the union of the domains of all of those functions. Further, if functions exist that are functionally dependent and use the information from manifold representations but at their different levels of abstraction then the abstraction transform must also be invertible. Where uncertainty exists about the likelihood of these conditions (i.e., dynamic scenarios where the interacting functions can change with the prevailing situation) then the strongest criterion of the existence of a continuous and invertible abstraction transform should probably apply to be safe. This criterion captures most of the problems that occur due to aggregation and de-aggregation. For example, when a unit is de-aggregated into individual platforms for the flyover of a reconnaissance aircraft then re-aggregated, engaged with losses and again de-aggregated when the aircraft overflys again, any inconsistencies the pilot observes result from inadequate abstraction transforms.

State Continuity. The continuity criterion preserves the causal relationships between the manifold representations and the other models in a distributed simulation. This criterion requires the meaningful hand-off from one manifold representation to another if a time interval gap exists between the execution of those representations as shown in Figure 4B. If the state of a manifold representation does not change during the interval Δt then the instantiation of a new representation need only assume the state of its previous instance when it ceased to exist. This may require communications between the two simulations. If the state of a manifold representation remains dynamic over the interval Δt then either the preceding or the succeeding simulation must compute the change of state to within the level of tolerance determined by the dependent functions. If the dependent functions require information about the manifold representation's state during the interval Δt then one or the other simulations must provide that information to preserve the causal relationships involving that object. Inadequate attention to preserving state continuity can result in an entity jumping magically from one location to another as its position is controlled by the different simulations at different times.

Table 2 maps the manifold representation criteria to the different cases described above.

Table 2. Application of Manifold Representation Criteria to Different Cases.

Abstraction Level	Parallel Representations	Sequential Representations
Single Level	State Correspondence	State Continuity
Different Levels	State Correspondence & Abstraction Transform	State Continuity & Abstraction Transform

Table 3 summarizes the manifold representation criteria and relates them to the anomaly types that they attempt to avoid.

Table 3. Linkage between the Manifold Representation Criteria and the Representational Anomalies.

Manifold Representation Criterion	Affected Representational Anomalies
State Correspondence	State Error & Registration
Abstraction Transform	State Error & Registration
State Continuity	All

Like the functional dependency criteria, those criteria presented for manifold representations are likely incomplete. Research continues to both strengthen and develop completeness for these criteria.

2.4 Substantive Interoperability Analysis

SOMs as specified by the OMT contain considerable information that can support automated or manual testing for substantive interoperability.

Functional Dependencies. Potential functional dependencies can be discovered by examining the Publishable/Subscribable field in the class structures of the SOM. One simulation subscribing to the information that another publishes suggests the existence of a functional dependency of the subscriber upon the publisher. This only suggests the possibility because the subscriber may not use the information due to other conditions (e.g., minimum range relationships) and the information gained from the subscription may not influence the results produced by the subscriber (e.g., its values below the minimum level of tolerance). Despite these extenuating circumstances, the existence of a subscriber-publisher relationship provides a pretty good indication of a functional dependency and permits some level of testing of the dependency representation criterion. A simulation that senses or reacts to another simulation's initiation of an object interaction also suggests the possibility of a functional dependency and that provides additional information to test the dependency representation criterion. If either of these situations is detected then the attribute structures in the publishers and subscribers can provide some information to test their ability to interoperate substantively. Specifically, the units, resolution and accuracy fields can be used to test the dependency representation, representational accuracy, sensitivity consistency and error consistency criteria.

Manifold representations. The SOMs also contain some information to detect the existence of manifold representations. Two simulations declaring their support of the same object-attribute pairs may indicate that they maintain manifold representations at the same level of abstraction. Their also declaring those representations to be either transferable or acceptable further supports the existence of manifold representations, again at the same level of abstraction. The current OMT provides little more information to test the manifold representation interoperability criteria than that.

"Choosing your partners carefully" is the most important axiom for designing meaningful federations. The criteria suggested herein certainly strongly support that axiom. Good federate selection requires a sound federation conceptual model (FEDEP Step 2) that specifies the fidelity, the components of which the SISO fidelity conceptual framework define [2], necessary to accomplish the federation's purpose. Given a good statement of the needs of a federation and comprehensive statements of the candidate federates' representational capabilities, a federation designer can evaluate the capabilities of the candidate federates to meet those needs. Without sufficient information describing candidate federate capabilities the federation designer can neither make an accurate assessment of the sufficiency of those federates to meet the federation's needs nor evaluate whether those federates can interoperate substantively. A federate's SOM is of limited use in this process since, by design, it addresses only the characteristics of the data the federates produce for external consumption or consume from external sources. As shown here and recognized by Dahmann et al. [1], substantive interoperability demands the designer's understanding the internal representations of a federate whether or not they are exposed at runtime. The current OMT specification does not contain sufficient information to gain that understanding because it was not designed for that purpose. It was designed to support technical interoperability but not substantive interoperability.

Thus, the current OMT specification does not provide sufficient information to test all of the criteria for substantive interoperability suggested in this paper. One primary deficiency comes from the OMT's representation of objects as collections of attributes with no characterization of the properties of the functions that use or change those attributes. Without such a functional orientation, as described in the SISO fidelity conceptual framework [2], complete testing for substantive interoperability will not be possible. Adopting such an orientation would require the addition of range and domain fields for each dependent and independent variable for each function that uses or affects a subscribable or publishable attribute. This would provide the means to test the range consistency criterion as described above. Additional fields would also be needed for deviation, precision, and time interval sensitivity and precision to permit complete testing of the interoperability criteria associated with functional dependencies as suggested above.

Further, the current OMT specification provides even less insight into the nature of manifold representations than it does into functional dependencies. The primary shortcoming comes from the fact that manifold representations may not provide information that is either publishable or subscribable and thus would not be specified in the SOM at all. Hiding these details reduces the complexity of federation design through partitioning at the cost of complicating the testing for substantive interoperability. This choice enables manifold representations to abound unseen but not without influence upon the results produced by the federation. As a result, these unseen components can subtly introduce anomalies that affect substantive interoperability. Further, the current descriptions of transferable/acceptable and updateable/reflectable attributes assume that the manifold representations operate at the same level of abstraction. Only by enhancing the current OMT to provide more information about the existence and nature of manifold representations, as well as the existence (or lack of existence) of any abstraction transforms, can overcome this problem. Specifically, the OMT must include information to permit the unambiguous detection of manifold representations and to test the state correspondence, abstraction transform and state continuity criteria as described above.

3. Conclusions & Recommendations

While critical to overall simulation interoperability, technical interoperability does not in itself guarantee the creation of a meaningful federation application. Federation designers must address the issues associated with achieving substantive interoperability in their federation designs since the purpose of the federation application drives the nature and degree of the interoperability needed.

To this end, this paper has identified four fundamental anomalies that disrupt substantive interoperability and two classes of criteria to detect the possibility that two simulations operating within the same federation may produce those anomalies. The current OMT specification [5] supplies some of the information necessary to detect substantive interoperability problems arising from both functional dependencies and manifold representations but not all that is needed. Further, simply adding additional fields to the current OMT structures will prove insufficient since it lacks the basic organization necessary to completely identify and characterize both functional dependencies and manifold representations. The following improvements would be needed.

- Add additional structures to permit describing functions that accept subscribable information and produce publishable information along with the information describing the fidelity of those functions as described in the SISO fidelity conceptual framework [2].

- Add additional structures to permit describing representations of the same object instances in different places within the federation as well as information that describes the natures of those representations and the conditions under which those representations may be executed.

Augmenting the OMT with this additional information and structure creates but one possible option for aiding the design of substantively interoperating federations. Considering the immense amount of work that has already been invested in the design of the HLA infrastructure, these are relatively minor changes. This research suggests that the existing RTI could easily support these changes with little or no changes to its design or function.

Another option is to formally incorporate the necessary information into the federation conceptual model. The federation conceptual model defines the federation's representational requirements and is critical to federation design. It supplies the information needed to select federates based upon the characteristics of their representations and their abilities to meet the needs of the federation application. Thus, it necessarily contains all of the information needed to test federate interoperability through the criteria defined above. Defining a standardized conceptual model framework that includes the information needed to test the substantive interoperability criteria might be considerably easier than augmenting the current OMT standards.

Incorporating the suggested changes would enable automation of a significant task that is undoubtedly performed manually during federation design and development. Such automation would significantly improve the validity of the resulting federations as well as reducing the cost of the effort necessary to reach that validity.

Finally, the research that resulted in the criteria for substantive interoperability described above, like the SISO fidelity conceptual framework itself [2], is incomplete. The suggested criteria do not form a mathematically complete set so testing all of the criteria does not guarantee substantive interoperability by any means. This work does, however, show a promising and potentially cost effective path toward reaching substantive interoperability and, in the end, valid simulation federations through the HLA.

4. References

[1] J. Dahmann et al., "HLA and Beyond: Interoperability Challenges," Paper No. 99F-SIW-073, Proc. Fall 1999 Simulation Interoperability Workshop, Orlando, FL, 12-17 September 1999, np.

[2] D.C. Gross et al., "Report from the Fidelity Implementation Study Group," Paper # S99-SIW-167, Proc. Spring 1999 Simulation Interoperability Workshop, Orlando, FL, 14-19 March 1999, np.

[3] S. Y. Harmon, "Analytical Determination of Interoperability between Distributed Simulations," Proc. 1996 Electronic Conference on Interoperability in Training Simulation, Falls Church, Virginia, on the Internet @ http://www.mystech.com/~smithr/elecsim96/analytic, April 1996, np.

[4] S.Y. Harmon, Analysis of Manifold Representations, Unpublished Manuscript, December 2000.

[5] Defense Modeling and Simulation Office, High Level Architecture Object Model Template, version 1.3, Defense Modeling and Simulation Office, Alexandria, VA, 5 February 1998.

5. Acknowledgments

This work has been generously supported by funding from the VV&A Program of the Defense Modeling and Simulation Office.

Author Biographies

SCOTT HARMON is president of Zetetix, a small business specializing in modeling complex information systems. Mr. Harmon has been developing rigorous techniques for the validation of simulation federations and human behavior representations.

SIMONE YOUNGBLOOD is a member of the Senior Professional Staff at the Johns Hopkins University Applied Physics Laboratory. She is currently on assignment at the Defense Modeling and Simulation Office (DMSO) serving as the Technical Director for VV&A. Ms. Youngblood currently serves as chair for the Simulation Interoperability Workshop's (SIW)VV&A Forum and co-chair for the DMSO VV&A Technical Working Group. Ms. Youngblood has supported several VV&A efforts, including the Navy's Naval Simulation System, the Joint Warfare System (JWARS), and the development of the current DoD VV&A Recommended Practices Guide.