Proceedings of the 2007 Winter Simulation Conference
S. G. Henderson, B. Biller, M.-H. Hsieh, J. Shortle, J. D. Tew, and R. R. Barton, eds.

USING META-LEVEL ONTOLOGY RELATIONS TO MEASURE CONCEPTUAL ALIGNMENT AND INTEROPERABILITY OF SIMULATION MODELS

Levent Yilmaz

Computer Science and Software Engineering
107 Dunstan Hall, Auburn University
Auburn, A.L. 36849, U.S.A.

ABSTRACT

Engineering large and complex simulation systems is becoming more reliant on the reuse of existing simulation models. While existing technical standards facilitate syntactic and technical interoperability among disparate simulation models, there is still lack of formal methods that enable sound reasoning about the conceptual congruity of models that are selected for composition. This paper suggests a graph-theoretic approach to measure the extent of conceptual congruity of models within a new context. The premise of the approach is based on having contextualized models that provide introspective access to their metamodels. A metamodel associated with a reusable model entails a conceptualization of the domain in which it is originally designed to be situated in. The metamodels are used to instantiate a metagraph and graph distance metrics are used to measure the alignment of metamodels in the context of the new application domain.

1 INTRODUCTION

Simulation interoperability is a longstanding challenge within the Modeling and Simulation (M&S) community (Carr and Myers 2003; Davis and Anderson 2003; Tolk 2004; Yilmaz and Oren 2006). While some engineering disciplines successfully apply component-based approach to build systems (Falkenheiner and Forbus 1991), it has proven significantly difficult to apply in simulation model development (Page and Opper 1999). Interoperability is viewed as the capability of simulations to exchange data and have a common interpretation. Semantic interoperability involves unambiguous interpretation of shared data at multiple levels of abstraction. Composability, which is defined at the *conceptual interoperability* (Tolk 2004) level, is defined as the capability to select and assemble components in various combinations to satisfy user requirements meaningfully (Davis and Anderson 2003; Petty et al. 2003).

M&S community is taking steps to facilitate the improvement of integratability (Guttenberg 2003), interoperability (DMSO 2004; Tolk and Muguirra 2003), and composability (Davis and Anderson 2003; Yilmaz 2004) of simulation models. Tolk (2004) suggests the use of open standards along with explicit delineation of model interdependencies as a prerequisite for a practical solution to composability. The Levels of Conceptual Interoperability Model (LCIM) involves different layers of interoperation from technical aspects to conceptualization.

This paper is based on the premise of explicit specification and use of context (Yilmaz 2004, NRC 2006) to measure the extent of composability. Here “composability” is viewed as the quality of being composable and means to be capable or worthy of being composed. Similar to other terms ending with –ability, for example maintainability, it refers to the object to which it applies and not to the agent (a model composer –human or software) which performs necessary acts to realize the composition of models and/or model components. The composability analysis is separated from the domain of the implementation models via a meta-level framework. The framework involves introspective access to conceptual models of individual simulations participating in the study. Conceptual models are viewed as graphs, and graph distance functions are used to evaluate the alignment of simulations with each other as well as the new domain constraints. The evaluation of consistency of the composed simulation with respect to the domain ontology of the new simulation application is assessed in two phases: First, a consensus summary metamodel is derived from the metamodels of the individual simulation models. Second, distance between the consensus metamodel and the domain ontology is compared.

The rest of the paper is organized as follows. In section 2 we present an overview of related work in the areas of simulation interoperability and composability. In particular, we discuss the LCIM model to provide the context for our method. Section 2 overviews issues in composability and delineates what specific challenge the

1-4244-1306-0/07/$25.00 ©2007 IEEE

proposed method is aiming to address. Section 3 introduces a case study that involves the conceptual model of a simulation study pertaining to UAV coordination mission. The domain ontology of the critical entities and their associations are presented in this case study. Section 4 introduces the elements of the proposed method along with the application methodology. Section 5 elaborates on the application of the method to the case study. Finally, section 6 concludes with the evaluation of the method and a discussion on potential avenues of further research.

2 BACKGROUND

Simulation interoperability involves (Tolk and Diallo 2005) (1) *administration* of information exchange via location, discovery, and retrieval of content, (2) *management* of the content through clarification and standardization of meaning, (3) ***alignment*** of the existing and required content, (4) *transformation* of existing content through aggregation, disaggregation, and transformation operations so that meaningful exchange can be established among disparate simulations.

2.1 The LCIM Model

Current research results led to the development of the “Levels of Conceptual Interoperability Model (LCIM),” which was presented in several papers (Tolk and Muguira 2003). LCIM copes with different layers of interoperation from technical aspects to conceptual ideas, which are the basis for the purposeful abstraction of reality underlying an M&S application. The LCIM introduces seven layers to cope with different aspects of interoperation. These aspects are characterized as follows:

- Stand-alone systems have *No Interoperability*.
- On the level of *Technical Interoperability*, a communication protocol exists for exchanging data between participating systems.
- The *Syntactic Interoperability* level introduces a common structure to exchange information, i.e., a common data format is applied.
- If a common information exchange reference model is used, the level of *Semantic Interoperability* is reached. *Pragmatic Interoperability* is reached when the interoperating systems are aware of the methods and procedures that each are employing. In other words, the use of the data – or the context of its application – is understood by the participating systems.
- If systems have attained *Dynamic Interoperability,* then they are able to comprehend the state changes that occur in the assumptions and constraints that each is making over time, and are able to take advantage of those changes.
- Finally, if the conceptual models – i.e. the assumptions and constraints of the meaningful abstraction of reality – are aligned, the highest level of interoperability is reached: *Conceptual Interoperability*.

2.2 Conceptual Interoperability

Composability of simulations requires establishing substantive alignment of models across a number of dimensions involving both functional and non-functional aspects (i.e., quality of service). At the functional domain, composability of models at least requires the alignment of conceptual and contextual assumptions of models and such assumptions are consistent with the constraints and objectives of the new application. To facilitate analyzing simulation assumptions regarding the concepts and the context, it is critical to formalize the mission and simulation space (Pace 1999). The simulation’s mission space includes all simulation elements, i.e., the things represented in the simulation and specifies how they interact with one another. It includes assumptions, algorithms, characteristics, relationships (especially interactions with other things within the simulation), data, etc., that identify and describe that item’s possible states, tasks, events, behavior and performance, parameters and attributes, etc. The simulation space part of the simulation concept includes all additional information needed to explain how the simulation will satisfy its objectives. Simulation space characteristics range from identification of specific kinds of computing systems (hardware and operating systems) and timing constraints so that real systems can be part of the simulation (such as hardware in the loop unitary simulations or involvement of live forces in distributed simulations) to the kinds of simulation control capabilities described above.

While the mission space focuses on the functional domain, simulation space includes additional characteristics pertaining to non-functional requirements. To assure composability at the mission space level (i.e., conceptual interoperability) we need to determine if there is a consensus among the used models regarding the conceptual and contextual assumptions. Once the consensus is confirmed, proper indicators are needed to measure the extent of congruity between the consensus model and the conceptual domain model of application The next section introduces the formalism used in capturing domain models.

3 MISSION SPACE MODELS AND DOMAIN ONTOLOGIES

To illustrate mission space specification for a composite model, we use a case study, in which a multiresolution coordinated mission for UAVS is being considered.

3.1 Simulation Modeling of Coordinated UAV Mission

Figure 1 presents the major components of the simulation. The simulation is driven by the MUAV tool that generates inputs for the tactical level. The scenario starts at the low resolution with a number of UAVs sweeping an area that contains multiple targets. Targets are classified as low resolution (i.e., tank battalions) and high-resolution entities (i.e., individual tanks). Individual UAVs can detect and destroy high-resolution entities such as tanks. In the case of a detection of a low-resolution aggregate entity such a battalion, UAVs aggregate into teams by virtue of a team formation strategy to establish multi-resolution entities, called Teams. Each team is composed of multiple UAVs. The high-resolution representation of a team is associated with a COA strategy component that encapsulates the coordination mechanism that implements the engagement policy of the team of UAVs.

A policy includes decision making strategy, task allocation approach, and the COA that is appropriate in the identified situation. There exist multiple strategy components from which the team object can choose at the time of its instantiation. The flexibility in choosing alternative strategy components as COAs facilitates experimenting with alternative realities.

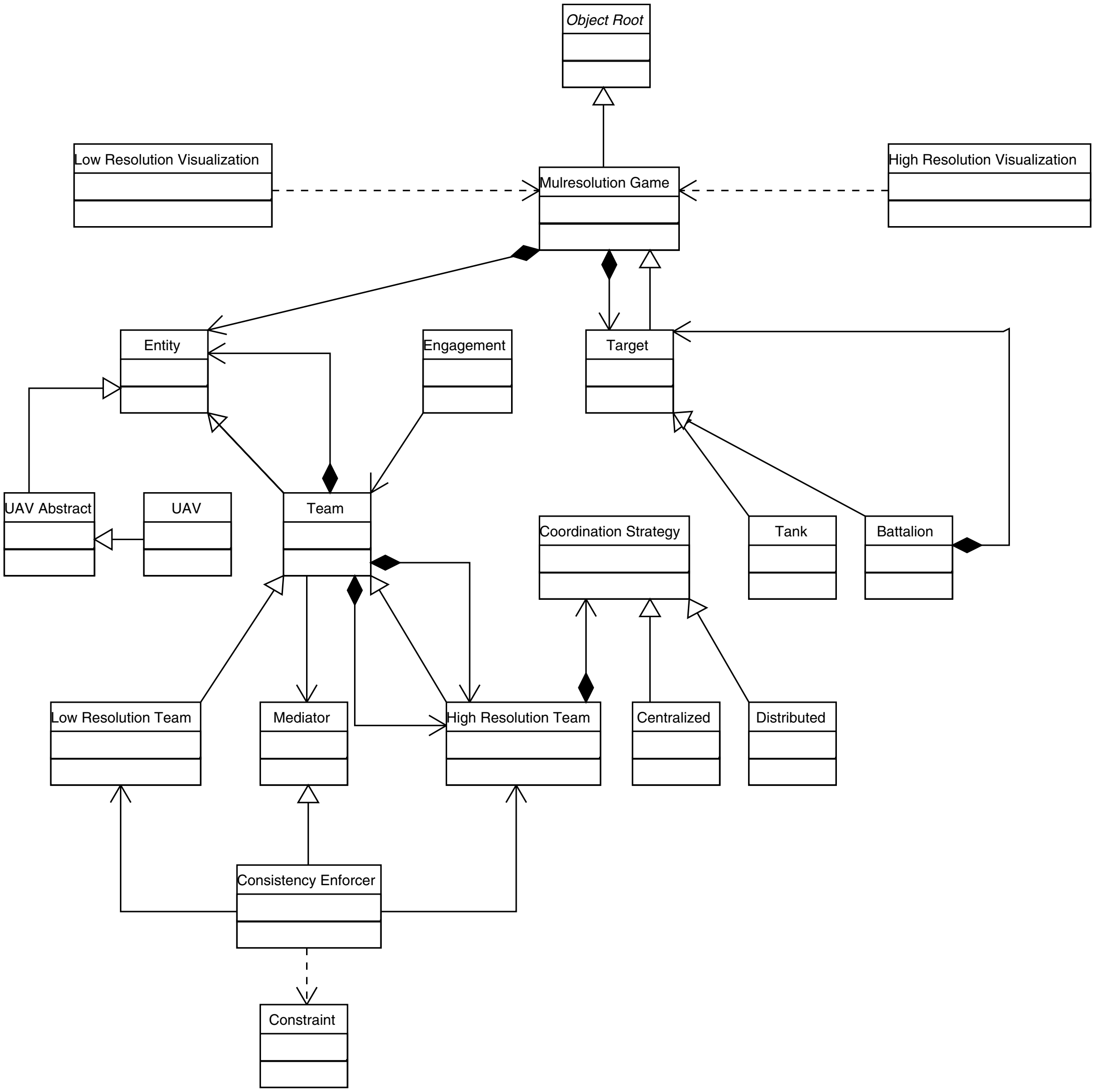


Figure 1: The Mission Space Model

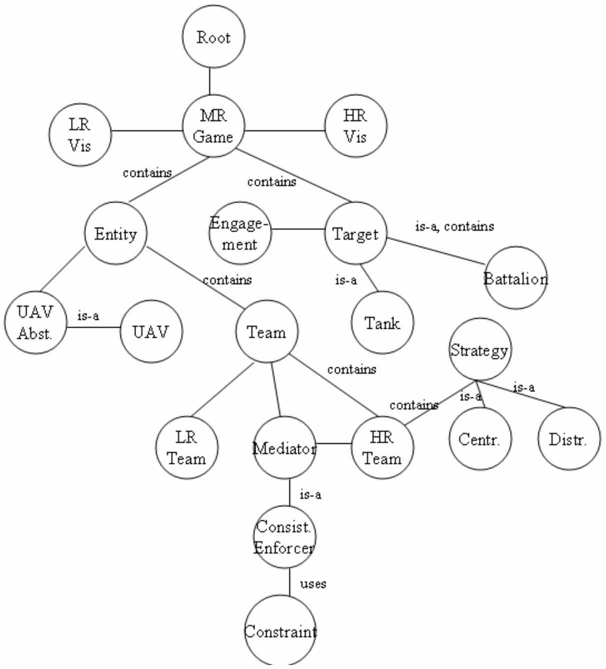


Figure 2: Undirected Graph-based Metamodel

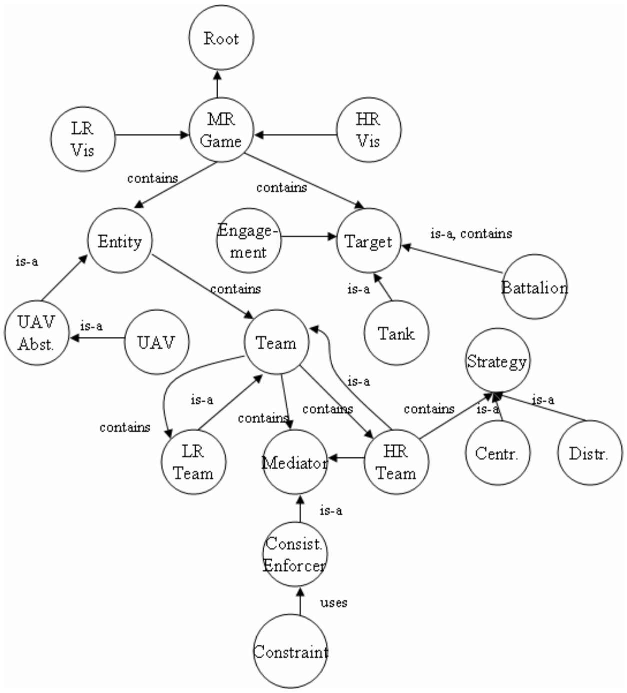


Figure 3: Directed Graph-based Metamodel

3.2 Transforming Mission Space Model into Graph Formalism

In our approach, conceptual models are defined as graphs. A graph G is defined in terms of $V = V(G)$, which is a vertex or node set, and a set $E = E(G)$ of pairs of vertices that are elements of V. If the elements of E are unordered pairs of vertices, then the graph G is undirected. If the elements of E are ordered pairs of vertices, then the graph G is directed. Figures 2 and 3 depict undirected and directed graphs that represent the entities and relations shown in the conceptual model of the problem domain of interest (see sections 3.1 and 3.2). Representation of the conceptual model in terms of undirected graphs without annotations results in loss of information regarding the type and direction of dependency, as well as the cardinality constraints on the associations. While the direction of dependency can be captured via directed graphs, the constraints over the entities and associations are still not represented. The method is limited to measuring the extent of alignment of the consensus model and the conceptual model of the problem domain. Each simulation model is expected to have a metamodel represented in terms of the graph formalism, as shown in Figures 2 and 3.

Metamodels can be packaged and distributed along with the simulation models using the introspective metadata wrappers as discussed in section 6. For the sake of brevity and simplicity of illustration of the method, we assume metamodels share a common domain vocabulary. Discrepancies in domain vocabularies can be resolved via manual or automated use of existing thesaurus as preprocessor before the use of the proposed method.

4 A GRAPH-THEORETIC VIEW ON CONCEPTUAL INTEROPERABILITY

As depicted in (Tolk and Diallo 2005), interoperability requires the alignment of the required and available content. Measuring the extent of discrepancy between the available models, as well as the required conceptual model of the problem domain and the agreed upon synthesized model would be an indicator of interoperability at the conceptual level; hence the composability of the synthesized model. In this section we elaborate on the notion of alignment and a solution to measuring it in terms of well-known graph distance metrics.

4.1 Aligning Metamodels of Simulation

Given a set of metamodels depicting the assumptions of reused simulations with respect to the problem domain, exploring how they align with each other involves finding a consensus or summary model. Different strategies may be used to measure the holistic discrepancy within a collection of models.

- *Centrality* aims to minimize the worst case distance between the graph that includes the central nodes, which have elements with minimum eccentricity (Buckley and Harary 1990) among the metamodels, and individual metamodels. Assuming that each model is represented as a node in a graph, eccentricity of a node in a graph is defined as the distance to a node farthest from *v*. Each central node can be characterized as a center, and central nodes prevent high-level dissent or discrepancy of any given node from the collection of nodes.
- Alternatively, one can maximize the agreement between the individual metamodels and the summary graph that is representative of the metamodel of the synthesized simulation. In other words, minimizing the average distance between the given set of metamodels and the derived conceptual model of the synthesized model (consensus metamodel), the *median* metric suggests a consensus model that improves the overall agreement of the collection.

4.2 Meta-level Graph

To measure the alignment of the metamodels of reused simulations, we use an abstract graph representation, the nodes of which are the metamodels. This meta-level graph abstracts away the details of individual metamodels and focuses on (1) dependencies between simulations and (2) derivation of a consensus summary graph (i.e., central or median graph). Figure 4 illustrates the abstraction hierarchy that constitutes the simulation models, their metamodels, and the meta-level graph. Simulation models and their concrete dependencies characterize the concrete implementation level. In our case study, each simulation model is represented by a federate. A dependency between two simulations exists when two entities are said to interact or exchange information. For instance, the tactical and the team federates, in our case study, are dependent, as COA (strategy) components are selected by the tactical federate to instantiate and configure high-resolution team objects governed by the team federate. Also, the high-resolution target visualization federate is dependent on the team and engagement federates, as the visualization is updated based on the change of state in the team behavior and dynamics of the engagement scenario. Each federate has its own metamodel as depicted by the middle layer in Figure 6. The metamodel captures the assumptions of the federate about the context in which it is expected to be embedded. That is, it represents the domain model for which it is originally designed to operate in. The highest level of abstraction is represented by the meta-level graph.

The meta-level graph is defined in terms of the graphs that denote metamodels of the individual simulations. Let G be the set of graphs and $R = R(G) = G \times G$ be a relation over the set of graphs. A dependency $(m_1, m_2) \in R$, if there exists a concrete dependency between simulation s_1 and simulation s_2 , such that s_1 is specified by m_1 and s_2 is specified by m_2 . The meta-graph is defined as $V(M) = G$ and $E(M) = R(G)$. Given that the purpose of the meta-level graph is to identify an agreement (consensus) model, the vertex set, $V(M)$, contains just those metamodels in the given set, and each graph (i.e., node in the meta-level graph) contains exactly the same edges as the original metamodel. The next step in defining the meta-level graph is to identify the edges in terms of relations on the set of metamodels. A relation between two metamodels is predicated on the existence of a dependency between corresponding simulations. If two simulation models depend on each other, then a relation is inserted between corresponding metamodel nodes. The meta-level graph is then processed to determine and assign weights on each individual relation by using graph distance metrics. These metrics measure the extent of similarity in the conceptual views of individual simulations. The metrics can be used in conjunction with a distance threshold value so that relations (i.e., edges in the meta-level graph) can be dropped if the distance between the corresponding nodes is over the designated threshold.

4.3 Graph Similarity Metrics

Graph metrics are functions that compute distances between pairs of graphs. The metamodels in our study are either undirected or directed graphs, so we focus on these two basic types of graphs. The metrics assume a common vertex set, and they are defined in terms of the adjacency matrix representation. The metrics used in this work build on prior work based on symmetric difference approach (Banks and Carley 1994). In this approach, the premise is that the difference between graphs is due to existence or absence of links between nodes. These metrics can be extended in the future to take into account the prominence of nodes as well as multiplicity (cardinality) of associations between concepts. Before presenting the metrics that measure the distance between graphs, we first specify the basic constructs and formal definitions over which the metrics are declared. Given a graph G with vertex set $V = [v_1, v_2, ..., v_n]$, the adjacency matrix $_A$ is a $n \times n$ matrix, where $a_{ij} = 1$ if $(v_i, v_j) \in E(G)$.

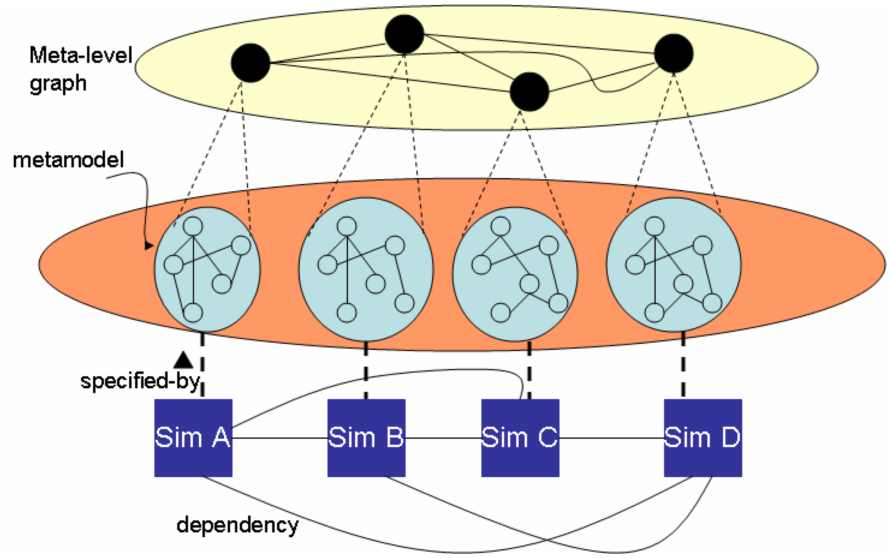


Figure 4: Abstraction Hierarchy

4.3.1 Metrics for Undirected Graphs

In prior work presented by Banks and Carley (1994), a symmetric difference concept is used to estimate a central graph. The strategy used in this approach is to count the discrepancies in the edges of two graphs. Its functional form for undirected graphs is given by the following equation (Banks and Carley 1994)

$$d(G_1, G_2) = \frac{1}{2} tr[(A_1 - A_2)^2], \qquad (1)$$

where the *tr* operator simply sums the diagonal entries in the matrix. Langfield-Smith and Wirth (1992) report an alternative formulation that provides the same results. The following equation is adopted from their study.

$$d(G_1, G_2) = \frac{1}{2} \sum_i \sum_j | a_{ij}^1 - a_{ij}^2 | \qquad (2)$$

4.3.2 Metrics for Directed Graphs

The symmetric difference metric (Banks and Carley 1994) has also been applied to directed graphs. Therefore, we will also be utilized in our approach to measure the extent of dissimilarity between metamodels. When applied to directed graphs, the metric takes slightly different algebraic form. This modified function is given by (Banks and Carley 1994) as follows:

$$d(G_1, G_2) = tr[(A_1 - A_2)^T (A_1 - A_2)] \qquad (3)$$

Similar to the metric for undirected graphs, the above formulation can be equivalently expressed by the following equation (Langfield-Smith and Wirth 1992).

$$d(G_1, G_2) = \sum_i \sum_j | a_{ij}^1 - a_{ij}^2 | \qquad (4)$$

4.4 Consensus Models

Finding central elements that maximizes conceptual agreement (alignment) for a given collection of metamodels would provide means to measure if the synthesized model fits into its new context. The *center* and the *median* of the meta-level graph are two alternatives for computing the consensus model for the collection. The consensus model can then be compared to the application domain conceptual model to determine if the synthesized model is sufficiently close for use in this new context. If the similarity is below a particular threshold, then reused simulations can be updated (tuned) in conjunction with their metamodels until the threshold value is achieved. The threshold depends on the application domain and the extent of credibility and correspondence needed from the synthesized model. Alternatively, one can continue searching for new simulation models, the metamodels of which bring the level of agreement between the specification of the synthesized model and the application conceptual model is sufficient for the purpose of the study.

There are a number of metrics for identifying central elements (Buckley and Harary p.31 1990) reported in the graph theory literature. Two of these are particularly relevant to the study presented here. The center measure that is based on the eccentricity metric enables finding those nodes in the meta-level graph that prevent high-level dissent or discrepancy of any given node from the consensus metamodel depicted by central node. On the other hand, the median metric (Buckley and Harary p.42 1990) minimize the average distance from the metamodels of individual simulation models to the agreement model depicted by the median node. In other terms, the use of the median nodes as agreement models maximize the agreement between the individual metamodels and the summary graph that is representative of the metamodel of the synthesized simulation.

4.4.1 The Center

To determine the center nodes, one has to compute the eccentricities of all the nodes in the meta-level graph and select those nodes, for which the eccentricity value is minimum. Given a meta-level graph there are either weights associate with edges or both weights and directions. The weights depict the distance between two metamodels that is computed using the distance metrics presented in section 4.3. Given a connected graph G, let v be a node of G. The eccentricity of v is defined as

$$e(v) = \{\max d(u,v) : u \in V\} \qquad (5)$$

The radius $rad(G)$ denotes the minimum eccentricity of the nodes. More specifically, v is central node if $e(v) = rad(G)$. The first step in identifying the central nodes is to compute distance metrics for each node in the meta-level graph. Note that using the direct dependencies between the metamodels one can compute the distance metrics, assign weights to edges, and specify them in the adjacency matrix representation of the meta-level graph. The indirect dependencies and associated weights need to be derived by producing the distance matrix in terms of the consecutive powers of the adjacency matrix. The second step is to identify for each node v the node which has the highest disagreement (dissimilarity). The central nodes are then defined as the ones with minimum eccentricity. As such, the central nodes minimize the discrepancy between the rest of the metamodels and the agreement model represented by the central node. It is important to recognize that there can be multiple nodes with the same minimum eccentricity, and each one of these nodes is an element of the set of central nodes. Figure 7 presents the algorithm used to realize the first step of the method defined above. Using the derived distance matrix, we can identify the eccentricities of a specific node v using equation 5. Next, we need to locate the nodes with minimum eccentricity.

$$C(G) = \{u \in V(G) \mid e(u) = \min_{x \in V(G)} [\underset{w \in V(G)}{\overset{x}{E}}]\}$$

where

$$\underset{w \in V(G)}{\overset{x}{E}} = \max_w d_{xw}$$

The set $C(G)$ includes the central nodes with the same eccentricity value. If we let

$$C(G) = \{u_1, u_2, u_3, ..., u_k\},$$

where each $u_i, 1 \le i \le k$ is a node that represents a metamodel depicted by a graph. The consensus model is selected by measuring the distance between each element of $C(G)$ and the provided conceptual model M, which specifies the context of the application domain. We use graph distance metrics to measure the distance between the selected central node and M. Let the distance between each node $u_i, 1 \le i \le k$ and M be depicted by $D_{i,M}$, then the central node with the minimum distance is $D_{j,M} = \min_i \{D_{i,M}\}$, and its metamodel is depicted by the node u_j. If we assume the existence of a threshold, δ_A, for the simulation application A, one has to assure that the consensus model of is sufficiently relevant in terms of its conceptual model to the application domain model. This constraint can be evaluated by checking if $D_{j,M} \le \delta_A$. Otherwise, the simulations and/or the application constraints need to be tuned or new simulation models should be located for reuse until $D_{j,M} \le \delta_A$.

4.4.2 The Median

While the centrality measure aims to minimize the discrepancy of individual metamodels from the consensus model, median maximizes the agreement between the individual metamodels and the summary graph that is representative of the metamodel of the synthesized simulation. In other words, minimizing the average distance between the given set of metamodels and the derived conceptual model of the synthesized model (consensus metamodel), the *median* metric suggests a consensus model. Formally, let G be a connected graph. The $dist(v)$ of a node v in G is the sum of distances from v to each other node in G. This is introduced by Harary (1959). The median $Med(G)$ of a graph G is the set of nodes with minimum distance; the total distance $tdist(G)$ is the sum of all the distances

That is, $Med(G)$ minimizes the total distance between any vertices in the graph. Algorithms for the derivation of

$Med(G)$ are presented in (Buckley and Harary 1990). The following set theoretic specification provides a declarative formalization of $Med(G)$.

$$Med(G) = \{v \in V(G) \mid \sum_{w \in V(G)} d(v,w) = \min_{u \in V(G)} [\sum_{w \in V(G)} d(u,w)]\}$$

An important point to recognize is that there can be multiple nodes that exhibit the characteristic of a median node. In that case a strategy that is similar to the one discussed in section 4.4.1 can be used to select the node that minimizes the distance against the simulation application's conceptual model. Furthermore, threshold value can be used to assess the fitness of the consensus model.

It is also important to notice that if $C(G) \cap Med(G) \neq \Phi$, then there exist nodes that act both as central and median.

$$d(Engagement\ , HRV) = \frac{1}{2} \sum_i \sum_j | a_{ij}^1 - a_{ij}^2 |$$

$$= \frac{1}{2}[1+1+1+1+1+1+1] = 3.5$$

Note that while there may not be a direct coupling between two simulations, they may indirectly affect each other via intermediate models that share and disseminate data.

Such nodes are particularly good candidates for being consensus models, as they not only minimize dissent of any given metamodel, but also maximize the agreement between the individual metamodels and the consensus model. The resultant meta-level graph is denoted by an adjacency matrix, as shown in Figure 7. To compute the set of central nodes, the eccentricity values for each node at the meta-level is computed. As denoted by equation , the eccentricity for a node is computed using

$$e(v) = \{\max d(u,v) : u \in V\} .$$

For instance, the eccentricity value for the engagement model is

$$e(Engagement) = \max\{d_{5,1}, d_{5,2}, d_{5,3}, d_{5,4}, d_{5,5}, 3.5\} .$$

Once the eccentricity values are calculated, the next step is to identify the model with the minimal eccentricities. The distance between the agreement model and the mission space model provides insight about the fitness of the synthesized model to the context depicted by the mission. A discrepancy beyond a selected threshold requires either selecting new models for assembly or, if possible, revising existing models and their metamodels to fine tune the consensus model until fitness to mission space is assured.

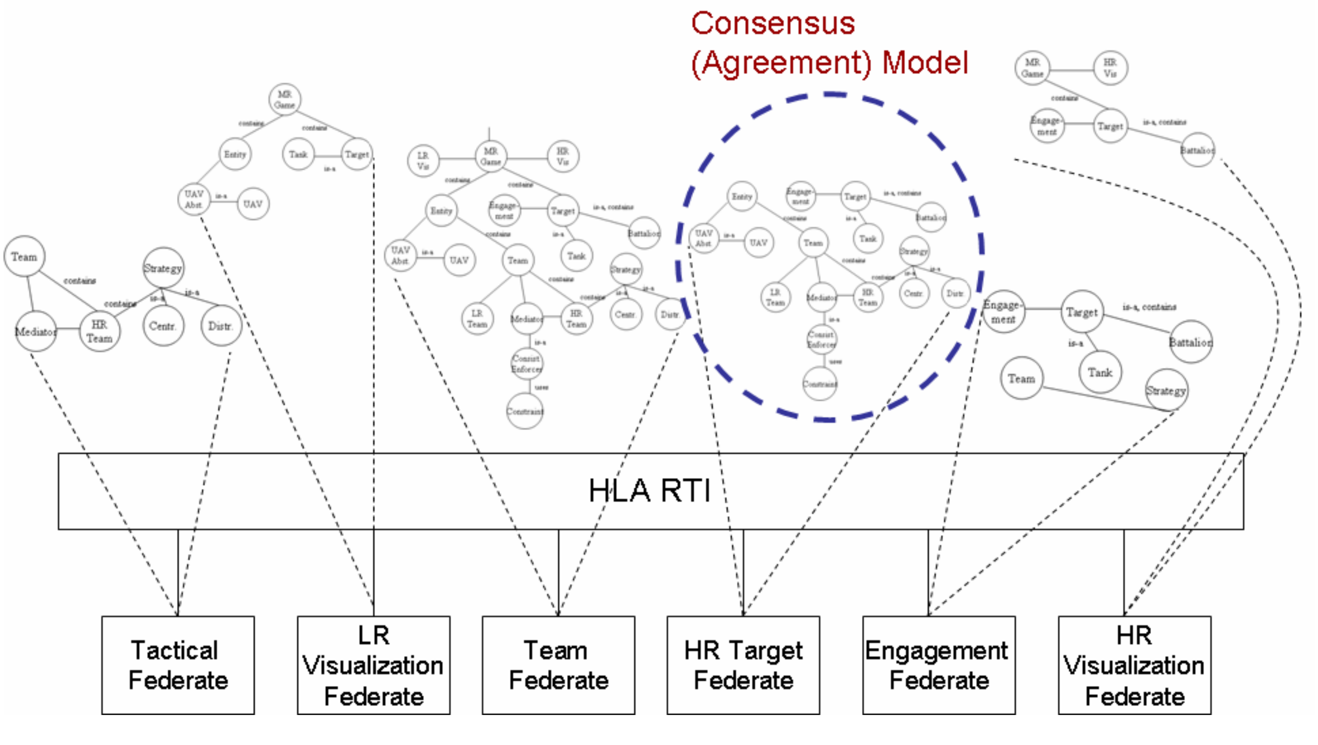
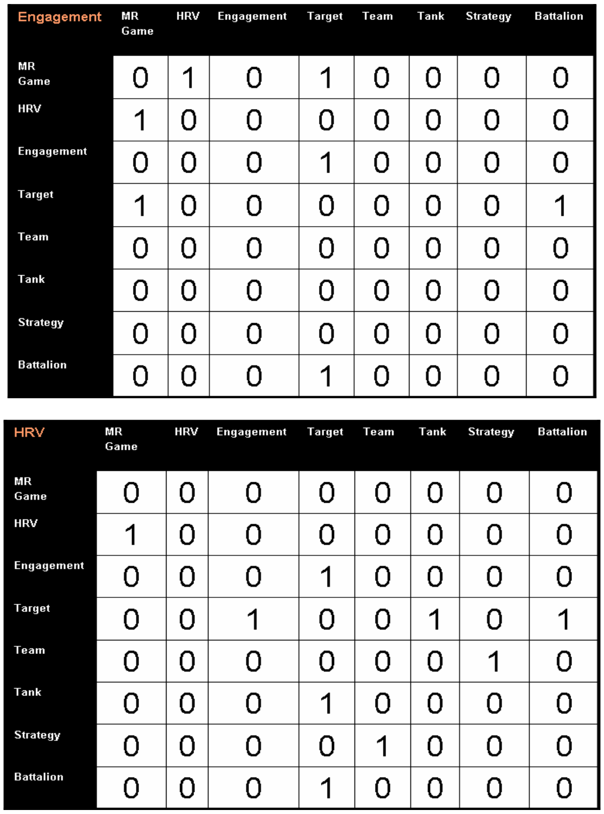


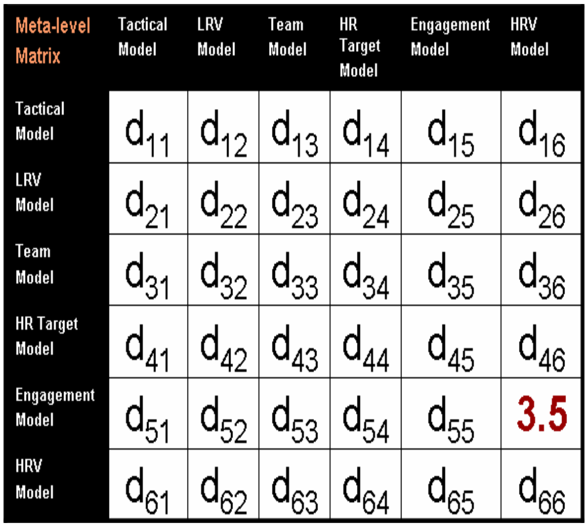
Figure 5: The Consensus Model



Engagement	MR Game	HRV	Engagement	Target	Team	Tank	Strategy	Battalion
MR Game	0	1	0	1	0	0	0	0
HRV	1	0	0	0	0	0	0	0
Engagement	0	0	0	1	0	0	0	0
Target	1	0	0	0	0	0	0	1
Team	0	0	0	0	0	0	0	0
Tank	0	0	0	0	0	0	0	0
Strategy	0	0	0	0	0	0	0	0
Battalion	0	0	0	1	0	0	0	0

HRV	MR Game	HRV	Engagement	Target	Team	Tank	Strategy	Battalion
MR Game	0	0	0	0	0	0	0	0
HRV	1	0	0	0	0	0	0	0
Engagement	0	0	0	1	0	0	0	0
Target	0	0	1	0	0	1	0	1
Team	0	0	0	0	0	0	1	0
Tank	0	0	0	1	0	0	0	0
Strategy	0	0	0	0	1	0	0	0
Battalion	0	0	0	1	0	0	0	0

Figure 6: Adjacency Matrices for the Engagement and High Resolution Visualization (HRV) Models



Meta-level Matrix	Tactical Model	LRV Model	Team Model	HR Target Model	Engagement Model	HRV Model
Tactical Model	d_{11}	d_{12}	d_{13}	d_{14}	d_{15}	d_{16}
LRV Model	d_{21}	d_{22}	d_{23}	d_{24}	d_{25}	d_{26}
Team Model	d_{31}	d_{32}	d_{33}	d_{34}	d_{35}	d_{36}
HR Target Model	d_{41}	d_{42}	d_{43}	d_{44}	d_{45}	d_{46}
Engagement Model	d_{51}	d_{52}	d_{53}	d_{54}	d_{55}	3.5
HRV Model	d_{61}	d_{62}	d_{63}	d_{64}	d_{65}	d_{66}

Figure 7: Adjacency Matrix for Meta-level Graph

5 CONCLUSIONS

This paper suggests a graph-theoretic approach to measure the extent of conceptual congruity of models within a new context. The premise of the approach is based on having contextualized models that provide introspective access to their metamodels. A metamodel associated with a reusable model entails a conceptualization of the domain in which it is originally designed to be situated in. The metamodels are used to instantiate a metagraph and graph distance metrics are used to measure the alignment of metamodels in the context of the new application domain.

REFERENCES

Banks, D. and Carley K. 1994. Metric inference for social networks. *Journal of Classification*, 11: 121-149.

Buckley, F. and Harary, F. 1990. *Distance in graphs*, Addison Wesley, Redwood City, CA.

Carr, F.H and Myers, L. 2003. Interoperability and reuse through a modeling and simulation common operating environment," In *Proceedings of the Simulation Interoperability Workshop 035-SIW-016*, Spring 2003.

Davis, K. P. and Anderson A. R. 2003, *Improving the composability of department of defense models and simulations*, RAND Technical report available at http://www.rand.org/publications/MG/MG101/ (last accessed April 2007).

Defense Modeling and Simulation Office (2004). "Composable Mission Space Environments," available at https://www.dmso.mil/public/warfighter/cmse (last accessed April 2007).

Falkenhainer, B. and Forbus, K. 1991. Compositional modeling: Finding the right model for the right job. *Artificial Intelligence*, 51:95–143.

Guttenberg, J. 2003. The elusive nature of components. In *Improving the Composability of Department of Defense Models and Simulations*. RAND Technical report, available at http://www.rand.org/publications/ MG/MG101/ (last accessed April 2007).

Harary, F. 1959. Status and contrastatus. *Sociometry*, 22: 23-43.

Langfield-Smith, K. and Wirth, A. 1992. Measuring differences between cognitive maps. *Journal of the Operational Research Society*, 43(12): 1135–1150.

National Research Council 2007. Defense modeling, simulation, and analysis: Meeting the challenge. The National Academies Press. http://www.nap.edu/catalog/11726.html [last accessed April 2007].

Pace, K. D 1999. Development and documentation of a simulation conceptual model. In *Proceedings of the Fall 1999 Simulation Interoperability Workshop*, March 15-19, 1999.

Page, E., and Opper, J. 1999. Observation on the complexity of composable simulation. In *Proceedings of the 1999 Winter Simulation Conference,* 553-560.

Petty, D. M., Weisel, W. E., and Mielke, R. R. 2003. Computational complexity of selecting components for composition”. In *Proceedings of the Software Interoperability Conference*, 03F-SIW-073, Fall 2003.

Tolk, A. and Muguira, A. J. 2003. The levels of conceptual interoperability model. In *Proceedings of the 2003 Fall Simulation Interoperability Workshop* Orlando, Florida, September 2003.

Tolk, A. 2004. Composable mission spaces and M&S repositories - Applicability of open standards”. In *Proc. Of Spring 2004 Simulation Interoperability Workshop (SIW)*, available at http://www.sisostds.org.

Tolk, A. and Diallo Y. S. 2005. Model-based data engineering for web services”. *IEEE Internet Computing*, 9(4): 65-70.

Yilmaz, L. 2004. On the need for contextualized introspective simulation models to improve reuse and composability of defense simulations. *Journal of Defense Modeling and Simulation*, 1(3): 135-145.

Yilmaz, L. and Ören, T. 2006. Prospective issues in simulation model composability: Basic concepts to advance theory, methodology, and technology. *The MSIAC's M&S Journal Online*, vol. 2, September 2006, pp. 1-7.

AUTHOR BIOGRAPHIES

LEVENT YILMAZ is assistant professor of Computer Science and Software Engineering in the College of Engineering at Auburn University. Dr. Yilmaz earned his Ph.D. and M.S. degrees from Virginia Tech. His research interests are on advancing the theory and methodology of simulation modeling, agent-directed simulation to explore dynamics of socio-technical systems, organizations, and human/team behavior, and education in simulation modeling. Dr. Yilmaz is a member of ACM, IEEE Computer Society, Society for Computer Simulation International, and Upsilon Pi Epsilon. His webpage is available at <http://www.eng.auburn.edu/~yilmaz>