A COMMON BEHAVIOR APPROACH TO INTEGRATING HETEROGENEOUS SIMULATIONS

Frederic McKenzie, Brett Butler and Christopher Dean
Rick_McKenzie@cpqm.saic.com
Brett_Butler@cpqm.saic.com
deanc@orl.saic.com
Science Applications International Corporation
Orlando, FL

Avelino Gonzalez
ajg@engr.ucf.edu
Dept. of Electrical and Computer Engineering
University of Central Florida

ABSTRACT

The Advanced Distributed Simulation Research Team (ADS RT) at SAIC-Orlando has been conducting experiments with the interoperability of simulations. One of these experiments focuses on a generic approach for sharing behaviors between Modular Semi-Automated Forces (ModSAF) and Close Combat Tactical Trainer Semi-Automated Forces (CCTT-SAF) with a goal of allowing military units from each simulation to perform together as one unit under the same task organization. It is anticipated that this approach will aid large scale or joint exercises by reducing SAF operator workload and allowing more use of varied simulation assets. This research explores a method of correlating the behaviors of units in different simulations so that they can interoperate with one another while performing unit tasks. The correlation will not be 100% since most simulations have different semantics and were designed for different training needs. An ontology of common generalized behaviors and behavior parameters, a database of behaviors written in terms of these common behaviors, and heuristic metrics are used to correlate specific behaviors from one simulation to specific behaviors for a target simulation. Behaviors are organized into several layers of aggregation down to primitive behaviors. It is this common behavior organization that is used by the closeness metrics to provide a generic methodology for the interpretation of missions and behaviors from one simulation and the initiation of comparable tasks in different simulations.

About the Authors

Dr. Frederic (Rick) McKenzie has been a member of the Advanced Distributed Simulation Research Team (ADS RT) since April 1995 currently holding a Senior Scientist position at SAIC Orlando. For two years prior to joining the ADS RT, he had been a member of the SAF Behaviors team for the Close Combat Tactical Trainer project. He obtained a M.S.E. in Computer Engineering in 1990 and a Ph.D. in Engineering in 1994 from the University of Central Florida.

Brett Butler is employed as a Chief Scientist for SAIC. He currently manages the ADS RT which is performing research into areas of Computer Generated Forces technology, software interoperability, advanced distributed simulation, artificial intelligence, and Distributed Interactive Simulation standards. Mr. Butler has over fifteen years of experience in the field of real-time simulation. His areas of interest include simulation, software engineering, and artificial intelligence. He has authored a variety of papers in these fields. He holds an M.S. degree in Mathematics.

Christopher Dean has been a member of the ADS RT since January 1995. Christopher has seven years of prior development experience creating commercial educational software. Christopher has a B.S.E. in Computer Engineering from the University of Florida in 1994 a M.S.E. from the University of Central Florida in 1996.

Dr. Avelino J. Gonzalez obtained his Ph.D. from the University of Pittsburgh in 1979 in Electrical Engineering. He spent nearly 12 years with Westinghouse Electric Corp. in Pittsburgh, PA and Orlando, FL working in computer applications to electrical power engineering (including the generator diagnostic expert system GenAID). In 1986, Dr. Gonzalez joined the Computer Engineering Department at the University of Central Florida, where he has focused his research and teaching in Artificial Intelligence and knowledge-based systems.

A COMMON BEHAVIOR APPROACH TO INTEGRATING HETEROGENEOUS SIMULATIONS

Frederic McKenzie, Brett Butler, and Christopher Dean
Rick_McKenzie@cpqm.saic.com
Brett_Butler@cpqm.saic.com
deanc@orl.saic.com
Science Applications International Corporation
Orlando, FL

Avelino Gonzalez
ajg@engr.ucf.edu
Dept. of Electrical and Computer Engineering
University of Central Florida

INTRODUCTION

The initial focus of Distributed Interactive Simulation (DIS) application development has been on training of large, joint, or combined forces which is lacking in traditional training. [DIS Steering Committee, 1994]. Since no one simulation can meet all the training needs required for large or joint exercises, multiple simulations must be used that can interoperate with one another seamlessly in a common environment. The DIS protocol was developed to promote interoperability in a heterogeneous simulation environment. Experience has shown that the DIS standards do not address all of the issues associated with interoperability. Although DIS provides standards and guidance for interface definition, communication, environment representation, management, security, field instrumentation, and performance measurement, it does not specify entity representation standards, behavior standards, synchronization standards, or spatial coherence standards (correlation of terrain, resolution correlation and environment correlation such as ambient illumination, buildings, weather, etc.). This research specifically addresses the behavior standards problem and the behavior interoperability of Semi-Automated Forces (SAF) simulations.

BEHAVIOR INTEROPERABILITY

Because of the military's desire to conduct large-scale training exercises and joint force operations, there is a growing interest in the use of SAF in the generation of simulated forces. Coordination between different services employing different SAF systems and the increased use of varied simulation assets requires that the SAF systems be capable of coordinated actions. Because DIS does not support behavior interoperability, it can be difficult for elements simulated by different simulations to coordinate behavior, especially if these elements are under a single task organization. CGF units must be able to be composed of entities that are owned and simulated by different simulations but must act properly under the specified task organization, i.e. each unit must coordinate with every other unit even if they are simulated by different simulations (Figure 1). Entire missions need to be arbitrated under this task organization. This can be a problem since the behavior of the simulations may be of a different fidelity or functionality. Also, different simulations may not even possess corresponding behaviors. Behavior interoperability addresses these interactions in an attempt to achieve the same performance for particular behaviors from different simulations.

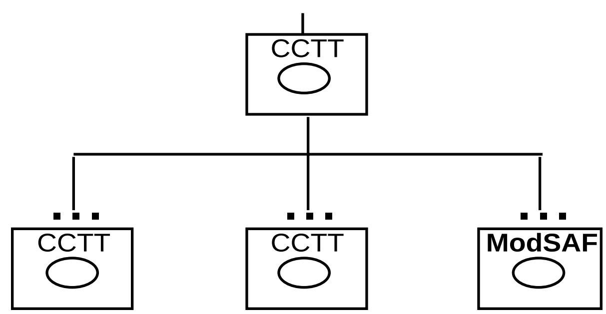

Figure 1
Task Organized Heterogeneous Simulation Units

To address the problem of behavior interoperability in heterogeneous distributed simulations, a common framework is necessary to provide a basis for arbitrating SAF behaviors (Smith, 1995). As illustrated in this research such a framework allows simulation entities, events, etc. from one simulation to be converted from their specific form to a general form and then to the specific form required by another simulation. The extra step of converting to the general common model provides flexibility in that it allows interoperability between different combinations of simulations without having to know the exact combination beforehand. For behavior interoperability, not only is a common behavior framework necessary but some degree of correlation of the behavior is required that can allow different simulations to execute comparable behaviors. This requires that all the necessary components of a behavior must be imitated by both simulations. However, this is not always the case and a best match must be found. For this research, an arbitrated behavior will be the best fit tactical maneuvers for a subordinate unit based on the requirements of its commanding unit.

Not only do behaviors need to be correlated but also the parameters associated with the behaviors. If the parameters of the commanding unit's behavior cannot be correlated with the target simulation behavior then the behavior cannot be executed. Figure 2 shows the chosen approach for behavior interoperability between CCTT SAF and ModSAF.

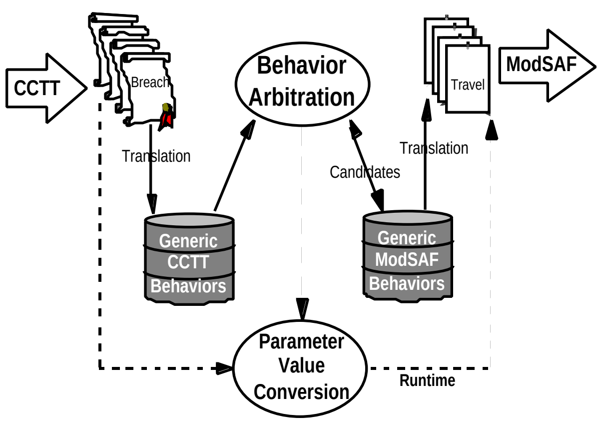

Figure 2
Common Behavior Interoperability Mechanism

As part of the subject research, a methodology was developed that promotes interoperability of behavior among simulations using a common behavior framework, along with heuristic metrics to correlate behavior. A set of closeness heuristic metrics has been defined for both behaviors and their parameters. These metrics will use the general behavior and parameter ontologies to select the behavior for a destination simulation with the best "semantic closeness" to the given behavior from a source simulation. A source simulation contains the commanding unit and a destination simulation represents the subordinate unit.

To satisfy the problem of interoperable SAF simulations, this research involves the development of a general framework for behavior and behavior parameters that facilitates the correlation between tactical procedures. The structure of this framework is domain independent which enables the system to be used with other applications outside military training. Additionally, the system may be used to perform off-line arbitration between known simulations or during run-time to allow any combination of simulations to interoperate. Parameter conversion is performed during run-time.

COMMON BEHAVIOR APPROACH

Closely associated with the semantics of a simulation are the structure (syntax) that the implementation uses to describe how the model performs its function [Altman et al., 1994]. Furthermore, Altman et al. [1994] contends that the simulated battlefield lends itself towards hierarchical decomposition and that abstractions are necessary to create a useful hierarchy. The approach taken in this research uses abstractions and hierarchical decomposition to create behavior hierarchies that can be used to compare the similarity of behavior. Behavior is not only composed of sub-behaviors but many times these sub-behaviors represent more general cases of the behavior in question. For example, the Occupy Battle Position behavior is composed of the more general Occupy Position behavior with its corresponding parameters. At the lowest level of decomposition, the behaviors will be considered primitives. Behaviors for military simulation are often expressed as higher level behaviors written in terms of four primitives: move, shoot, search/observe, communicate [Ourston et al., 1995]. In addition to being expressed in terms of these primitives, the behaviors have associated with them a set of parameters, situational triggers for behavior changes (reaction to enemy contact, for example), and in some cases initial and termination conditions.

The idea of common primitives for behavior agrees with various sources in the CGF community. [Smith, 1995] suggests that a common modeling framework is needed to solve the interoperability problem. Similarly, [Altman et al., 1994] contends that a set of unifying semantics are necessary. A set of common behaviors and primitives can provide the unifying semantics necessary for the semantic correlation of behavior for heterogeneous simulations.

For behavior interoperability a common behavior framework is usually not guaranteed. Middleware components can be used that will enable any combination of simulations to be connected together. Figure 3 shows a CORBA-based architecture that utilizes middleware components for interoperability among heterogeneous simulations. Only the parts of this architecture necessary to test the arbitration methodology have been implemented to date. The plug-&-play user interface allows input, browsing, and maintenance of common generalized behaviors from given simulations. These generic behaviors are derived directly from the software code that implements the behaviors and may be generated automatically or interactively. An ODBMS stores the generic behaviors and the behavior and parameter hierarchies used to correlate behaviors. The DIS network connection allows specifications such as CCSIL (Command and Control Simulation Interface Language) to be leveraged as a mechanism for assigning behaviors to units. In such a case, the simulation plugs (ModSAF and CCTT SAF) indicated in Figure 3 would be CCSIL interfaces to their respective object databases. Otherwise, the plugs would be CORBA client interfaces from the simulations' object databases to the Object Request Broker (ORB).

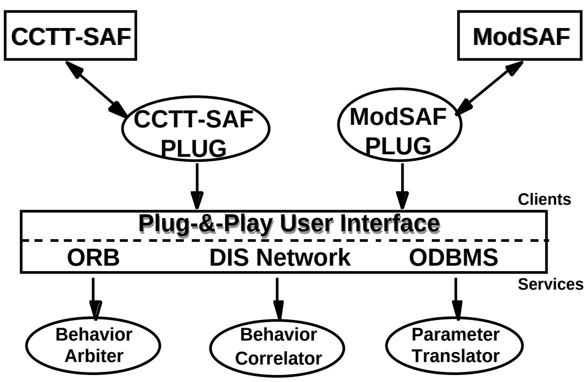

Figure 3
CORBA-based Architecture

In the terms of behavior interoperability, this architecture can be used to translate specific behaviors (from a source simulation) into more general ones which can then be translated into specific destination (destination simulation) behaviors for execution. Once the source behavior is translated into its general form, it can be translated into any of the remaining n-1 simulations without prior knowledge of the pairing. In order to accomplish this, a generic, simulation independent representation of the behaviors was developed. Specific simulation behaviors are translated into behaviors written in terms of general domain behaviors and primitives. An ontology of behaviors and parameters is used to support the similarity metrics. The parameter decomposition and ontology must be completely common between both systems in order for correlation of parameters to be possible. Since the simulations are in the same domain and parameters are not as sensitive to interpretation as behaviors, this is acceptable. This behavior representation allows simulation specific behaviors to be translated to any of the n-l simulation systems. Correlation can be performed only at the primitive level if necessary. However, since simulations can only interoperate to the extent that they share common semantics [Altman et al., 1994], the more behaviors and primitives in common, the better the correlation of behavior.

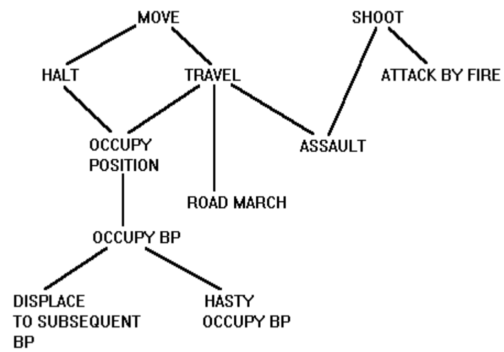

Figure 4
Partial Hierarchy for Tank Platoon Behaviors

Behavior Correlation Metrics

Behaviors are usually represented in terms of lower level behaviors until the primitive level is reached. Behaviors may be represented in terms of more general behaviors or of lower echelon behaviors. In the case of lower echelon behaviors, different behaviors may be assigned to different units. This is not a problem since the higher echelon behavior can still be considered to exhibit these behaviors even though not all lower echelon units exhibit all the behavior. Because there is an infinite number of ways the same behavior can be represented, a simple comparison is not sufficient. When trying to compare and correlate behaviors several metrics can be used to determine how similar they are:

- A source behavior can be found at a lower or higher level of decomposition of a behavior than in the destination behavior. This is defined as the WHERE-IS metric.

- A source behavior can be decomposed into its sub-behaviors which can then be correlated. This is defined as the HAS-A metric.

- A source behavior can be related to a more general or more specific behavior present in the destination behavior. This is defined as the IS-A metric. Note that an ontology such as that shown in Figure 4 is necessary for this determination.

- A source behavior can be related to a similar behavior of the destination. This is defined as the SIBLING-OF metric.

Any combination of these metrics can be used at the various levels of decomposition to determine the semantic closeness of two behaviors. In this context, semantic closeness is defined as the percentage that the destination behavior will perform the desired behavior. There is no guarantee that the chosen behavior will execute the same behavior as the source, only that it will be the best match possible among the available destination behaviors. Many times, behaviors may be essentially the same but are organized differently. Extra behaviors may also be present on either the source behavior or destination behavior. Extra behaviors on the destination behavior do not affect the closeness as it has been defined. Extra behaviors only mean that the destination behavior does more than needed which is acceptable. Extra behaviors on the source behavior do decrease the closeness since the destination behavior may be missing some important functionality.

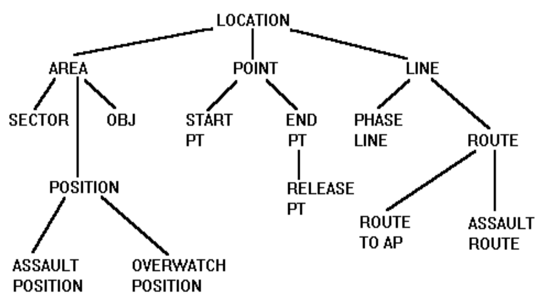

Figure 5
Partial Hierarchy for Behavior Parameters

Parameter Correlation Metrics

In addition to performing metrics when correlating behaviors, metrics must also be calculated for correlating the parameters associated with that behavior. Parameters either are necessary for the corresponding behavior to perform its function or modify how the behavior is executed. Common parameters for military behaviors include speed, formation, platform, route, etc. The metrics define how close the parameters between the two behaviors match. Parameter correlation is only performed for the top level source and destination behavior. The parameters of sub-behaviors are not really significant since as long as the initial parameters correlate, the behavior can be executed. In addition, many times the sub-behavior parameters will be derived internally and have no explicit relationship to the top level parameters.

There are three metrics that apply to parameter correlation, the IS-A, PARENT-OF and HAS-A metrics. The IS-A and PARENT-OF metrics both determine the closeness along an inference path between a source parameter and destination parameter using information shown in Figure 5. The IS-A metric determines if a destination parameter is a child of one of the source parameters. The metric determines the inferential distance between the two. Similarly, the PARENT-OF metric determines if a destination parameter is a parent of one of the source parameters. Unmatched (Additional) parents in a PARENT-OF metric also do not affect the closeness for the parameter. This just means that the parameter is more complex than the source parameter being correlated which is satisfactory. These two metrics can be combined to generate a correlation path from a specific source parameter to a more general parameter and then back to a more specific destination parameter. For example, an ASSAULT_POSITION can be correlated to an OBJECTIVE by following the inference path from ASSAULT_POSITION to POSITION to AREA to OBJECTIVE, where OBJECTIVE is a specific type of AREA. The HAS-A metric determines the closeness along a decomposition path between a source parameter and destination parameter. For example, suppose a ROUTE can be decomposed into a START_POINT and END_POINT. Then, a source ROUTE parameter can be correlated with START_POINT and END_POINT parameters of the destination behavior. The IS-A and PARENT-OF metrics can be combined with the HAS-A metric so that the sub-parameters of a parameter may also be matched with destination parameters.

Incremental Decomposition and Abstraction

The correlation algorithm uses incremental decomposition and abstraction of behaviors to determine the closeness. Each source behavior is recursed into and is compared (via recursion again) to the levels of the destination behavior. Each behavior is decomposed into its sub-behaviors which are also correlated down to the primitive level. The correlation algorithm uses the following high level steps when correlating a source behavior:

1) Check for the presence of the source behavior at the given level of decomposition in the destination behavior.
2) If the behavior is not present, apply the WHERE-IS, IS-A, HAS-A, and SIBLING-OF metrics, using the maximum closeness result.

3) Recurse into the source behavior, performing these steps on each sub-behavior. Combine the results of the sub-behavior correlations and multiply the result by the closeness value determined in one of the two previous steps.

4) Repeat steps 1-3 on the next behavior at this same level of decomposition.

The parameter correlation algorithm follows the same basic steps, with the parameter metrics being applied instead. It is important to note that the names of behaviors can increase the closeness if they match, but behaviors that match in name are not necessarily equal. The closeness must be determined down to the primitive level to determine an accurate correlation (hence the presence of step 3 above). The correlation algorithm uses the semantic closeness metrics defined earlier to determine the behavior closeness value. This value is calculated using closeness factors for each metric along with a few others. These factors may need to be adjusted for a specific simulation in order to optimize the correlation. As a behavior is correlated, the metric that produces the best closeness value is combined with the aggregate closeness value of its sub-behaviors.

The parameter correlation mechanism is a simpler form of the behavior correlation algorithm. Default parameters are defined as those which have preset values within their appropriate simulations and are not required to be set for the behavior to be executed.

Behavior Translation

Before any arbitration can be performed, specific behaviors must be translated into a form that provides a common language for interoperability between the two simulations. Thus when a TRAVEL behavior is being correlated and a TRAVEL destination behavior is found, the system can assign a higher closeness than if the behavior was unknown. As previously mentioned however, the system makes no assumptions about the behaviors being the same because they have the same name. The sub-behaviors are always checked to verify the closeness. The general representation serves as this common language. Examples of translations include converting specific-named behaviors to general names, removing redundant or superfluous behaviors, breaking up aggregate parameter structures, etc. None of these translations are required, they only serve to enhance the correlation with some a priori knowledge about the systems being correlated. This can be done during run-time by a simple set of conversion rules.

CCTT was used as the model for this general representation since it has validated behaviors. Thus, only minor translations were needed for conversion to the general form. ModSAF behaviors, however, require more translation. Figure 6 shows the ModSAF assault behavior as defined by a ModSAF task frame.

```
HALT    (Prepatory Frame)

UNIT_ASSAULT (Actual Frame)
  unit_mixed_travel
    unit_travel
      vehicle_move
    unit_follow_unit
      unit_travel
        vehicle_move
  unit_mixed_targeter
    unit_targeter
      vehicle_targeter
      vehicle_assess
      vehicle_search
  unit_mixed_prep_occupy_position
    unit_prep_occupy_position
      vehicle_occupy_position
  unit_occupy_position
```

Figure 6
ModSAF Assault

```
ASSAULT
  TRAVEL
    vehicle_MOVE
    vehicle_SEARCH
    FOLLOW_UNIT
      TRAVEL
        vehicle_MOVE
        vehicle_SEARCH
  TARGETER
    vehicle_SHOOT
    vehicle_ASSESS
    vehicle_SEARCH
  OCCUPY_POSITION
    vehicle_OCCUPY_POSITION
```

Figure 7
ModSAF Assault in General Form

Figure 7 shows the corresponding behavior in terms of the general representation. The prepatory frame was removed since it is not specific to an assault, and several behaviors were combined and renamed. The developers of ModSAF decided to separate their mixed platoon behaviors (combined tank and mechanized infantry platoon behaviors) from their homogeneous unit behaviors. The mixed behaviors are always assigned to units regardless. This distinction is not needed for correlation so the redundancy is removed.

Parameter Translation

A similar translation is done for behavior parameters as is done for behaviors. Simulation specific translation code is used to rename parameters and decompose complex parameter data structures into individual parameters. Also, the translation must remove parameters that are known to be implementation specific and thus are not a true attribute that defines the semantics of the behavior. Translation to a common parameter model is even more important than the translation of behaviors. If a completely common parameter model does not exist between simulations the parameter correlation may not be possible and thus the behavior correlation will not be possible.

EXPERIMENT RESULTS

This research focused on the arbitration of CCTT tank platoon behaviors with that of ModSAF tank platoon behaviors so they could interoperate under one task organization. Only those behaviors that could be assigned to tank platoons via their respective GUIs were considered. A CCTT behavior to be assigned to a ModSAF platoon would be correlated with each ModSAF behavior. The arbitration of the candidate ModSAF behaviors currently consists of picking the correlation with the highest closeness value. When closeness values are low or not significantly different, heuristics and thresholds may be added to further refine the arbitration.

Proof Of Principle

As a proof of principle, each of twelve CCTT tank platoon behaviors were arbitrated among twenty ModSAF tank platoon behaviors. Seven of these behaviors have expected pairings provided by subject matter experts. The remaining five have no corresponding ModSAF behavior. Results from unknown behaviors are subject to interpretation since no agreed equivalent already exists.

An Example Arbitration

As an example, the CCTT Assault An Enemy Position behavior is matched against the 20 ModSAF behaviors. A typical tank platoon assault behavior is concerned with issuing movement and firing commands to its vehicles. These commands instruct the vehicles to perform an on-line attack and occupy the objective. More specifically, the tank platoon closes with and destroys the enemy by overrunning and seizing the occupied enemy position. The tanks move rapidly in line formation under the cover from direct and indirect fire to the far side of the objective. Figure 8 shows the CCTT Assault an Enemy Position behavior. In this case, CCTT is more robust than ModSAF in that it provides for an initial travel to the assault position, allows for the breach of obstacles along the way, and a consolidation and reorganization of forces after the assault has been completed.

<u>CCTT ASSAULT AN ENEMY POSITION</u>:

```
TRAVEL
  vehicle_MOVE
BOUNDING_OVERWATCH
  TRAVEL
    vehicle_MOVE
  vehicle_OCCUPY_POSITION
    vehicle_MOVE
    vehicle_SEARCH
    vehicle_HIDE
      vehicle_HALT
      vehicle_MOVE
      vehicle_SEARCH
SEEK_COVER_AND_CONCEALMENT
  vehicle_SEARCH
vehicle_OCCUPY_POSITION
  …
GENERATE_REQUEST_FOR_IFIRE
CONSOLIDATE_AND_REORGANIZE
    SEEK_COVER_AND_CONCEALMENT
      vehicle_SEARCH
    vehicle_OCCUPY_POSITION
      …
    GENERATE_SITREP
```

Figure 8
CCTT Assault An Enemy Position

For the CCTT Assault An Enemy Position behavior, the following semantic closeness values were calculated for the ModSAF behaviors:

ASSAULT	0.522923
ASSEMBLE	0.264287
ATTACH	0.425562
ATTACK BY FIRE	0.39817
BREACH	0.431557
CHANGE FORMATION	0.265716
CONCEALMENT	0.401982
EXECUTE DELAY	0.419041
DETACH	0.425562
FOLLOW VEHICLE	0.0
HALT	0.264287
HASTY OCCUPY POSITION	0.377462
OVERWATCH MOVEMENT	0.43608
PLOW BREACH	0.431557
PURSUE	0.0
ROAD MARCH	0.422094
SUPPLY	0.400714
TRAVEL	0.422094

TRAVELING OVERWATCH	0.488249
WITHDRAW	0.409559

The highest correlation is with the ModSAF assault behavior with a semantic closeness of 52%. The actual closeness value is not so important as is the relative values between the different ModSAF behaviors. This pairing is the expected match. The semantic closeness values of zero represent cases where required ModSAF parameters could not be correlated. Figure 9 shows the ModSAF Assault behavior.

<u>MODSAF ASSAULT</u>:

```
EXECUTE_ASSAULT
  ASSAULT
    TRAVEL
      vehicle_MOVE
      vehicle_SEARCH
      vehicle_ENEMY
      FOLLOW_UNIT
        vehicle_MOVE
        vehicle_SEARCH
        vehicle_ENEMY
    TARGETER
      vehicle_SHOOT
      vehicle_ASSESS
      vehicle_SEARCH
    OCCUPY_POSITION
      vehicle_ALTERNATE
        vehicle_MOVE
      vehicle_TERRAIN
      vehicle_SEARCH
```

Figure 9
ModSAF Assault Behavior

The common primitives of vehicle_MOVE and vehicle_SEARCH (common toOCCUPY_POSITION) and the TRAVEL behavior are the primary reasons for the correct correlation. For similar reasons, the second and third choices (TRAVELING_OVERWATCH and OVERWATCH_MOVEMENT, respectively) exhibited high semantic closeness values. The presence of these primitives in several OCCUPY_POSITION behaviors offset some of the missing behaviors even though the positions being occupied are very different. The different positions are captured by the parameter correlation but their effect on the overall closeness is much smaller.

The CCTT parameters were correlated with the ModSAF parameters in the following fashion with their corresponding closeness values:

CCTT UNIT_ID to ModSAF UNIT_ID (SC = 1.0)
CCTT PLATFORM to ModSAF PLATFORM (SC = 1.0)
CCTT ROUTE_TO_AP to ModSAF ROUTE (SC = 0.9)
CCTT ASSAULT_ROUTE to ModSAF ROUTE (SC = 0.9)
CCTT ENEMY_POSITION to POSITION to AREA to ModSAF OBJECTIVE (SC = 0.729)
CCTT TRIGGER LINE to LINE to ModSAF ROUTE (SC = 0.81)
CCTT ASSAULT_POSITION to POSITION to AREA to ModSAF OBJECTIVE (SC = 0.729)
CCTT DEPARTURE_TIME to NO MATCH (SC = 0.0)
CCTT OBSTACLE defaulted (SC = 0.9)
CCTT BREACH_ROUTE to ModSAF ROUTE (SC = 0.9)
CCTT PRE-BREACH_ROUTE to ModSAF ROUTE (SC = 0.9)
CCTT POST-BREACH ROUTE to ModSAF ROUTE (SC = 0.9)
CCTT ALPHA_SECTION ignored (SC = 0.75)
CCTT BRAVO_SECTION ignored (SC = 0.75)
ModSAF LEFT_TACTICAL_BOUNDARY defaulted (SC = 0.75)
ModSAF RIGHT TACTICAL BOUNDARY defaulted (SC = 0.75)
ModSAF SPEED defaulted (SC = 0.75)
ModSAF DISMOUNTED_SPEED defaulted (SC = 0.75)
ModSAF STOPPING_ASSAULT_CRITERIA defaulted (SC = 0.75)
ModSAF SECURE_OBJECTIVE_FLAG defaulted (SC = 0.75)
ModSAF FORMATION defaulted (SC = 0.75)
ModSAF SPACING defaulted (SC = 0.75)
ModSAF X_DI_OFFSET defaulted (SC = 0.75)
ModSAF Y_DI_OFFSET defaulted (SC = 0.75)
ModSAF ASSAULT_REASON defaulted (SC = 0.75)
ModSAF DI_FORMATION defaulted (SC = 0.75)

The results agree with the predictions with one exception that illustrates one inherent problem with the parameter correlation. Destination parameters that are equally related to more than one source parameter cause an ambiguity as to which parameter correlation is the correct one. In this experiment there are five equally related source routes and only one destination route. We know that the ASSAULT_ROUTE is the best correlation but it is unclear as to how the algorithm can determine this automatically. Correlating in the other direction, a single source behavior can be matched against more than one destination behavior. In some cases this may be satisfactory but in other cases it may cause unexpected results and thus the destination parameters should have been allowed to default. Some a priori knowledge code may need to be used to modify the parameter correlation for known problems before assigning the behavior. As an example, code can be used that will check to see if all the routes are the same and if they are, default all the routes except the assault route. Also, the best correlations should take precedence over lesser correlations such as the TRIGGER_LINE in this case. The CCTT TRIGGER_LINE should be ignored since there are better ROUTE correlations. This is a trivial task that can be done when the actual parameter conversions are done. The ordering of the parameters may also be used to specify a priority as a conflict resolution scheme. However this may not always be correct when the simulations being arbitrated are determined at run time.

Experiment Conclusions

Based upon the results of the experiments, it has been shown that the use of heuristic metrics in conjunction with a corresponding behavior and parameter ontology is sufficient for arbitrating CCTT and ModSAF behaviors. Table 1 summarizes the results of the experiments. Out of seven expected matches, six were arbitrated correctly with the one exception due to a deficiency in ModSAF behavior. The remaining five unknown matches were deemed acceptable by subject matter experts under the given constraints. Most of the chosen correlations resulted in closeness values around 50% thus demonstrating the dramatic differences that can be present in externally similar systems.

Table 1.
SUMMARY OF EXPERIMENTAL RESULTS

#	CCTT SOURCE	MODSAF RESULT	MODSAF EXPECTED	SEMANTIC CLOSENESS	ACCEPT-ABLE
1	ASSAULT ENEMY POSITION	ASSAULT	ASSAULT	0.522923	YES
2	ATTACK BY FIRE	ATTACK BY FIRE	ATTACK BY FIRE	0.607225	YES
3	BOUNDING OVERWATCH	OVERWATCH MOVEMENT	OVERWATCH MOVEMENT	0.554897	YES
4	TRAVELING OVERWATCH	TRAVELING OVERWATCH	TRAVELING OVERWATCH	0.744768	YES
5	TACTICAL ROAD MRCH	BREACH	TACTICAL ROAD MRCH	0.51583	NO
6	TRAVEL	TRAVEL	TRAVEL	0.899357	YES
7	CONSOLIDAT REORGANIZE	DELAY	<NONE>	0.489362	YES
8	OCCUPY BP	ASSAULT	<NONE>	0.589559	YES
9	PASSAGE OF LINES	TRAVELING OVERWATCH	<NONE>	0.39317	YES
10	PLATOON DEFENSIVE MISSION	ASSAULT	<NONE>	0.540253	YES
11	PLATOON FIRE AND MOVEMENT	ASSAULT	<NONE>	0.528677	YES
12	HASTY OCCUPY POSITION	HASTY OCCUPY POSITION	HASTY OCCUPY POSITION	0.594519	YES

SUMMARY AND CONCLUSION

This research has shown that SAF behaviors can be arbitrated with behaviors from different simulations so they can interoperate with one another to support simulation training. Specific source behaviors are translated to a form in terms of common general behaviors which are then correlated with any desired destination simulation behavior without prior knowledge of the pairing. As the experiments show, the correlation may not be 100% since the simulations may have different semantics. The experiments do show that the use of heuristic metrics in conjunction with a corresponding behavior and parameter ontology is sufficient for arbitrating heterogeneous simulation behavior.

This research has shown that using a database of CCTT behaviors and ModSAF behaviors written in a general form, a common ontology of behavior parameters, and a set of heuristic metrics, that CCTT and ModSAF tank platoons can interoperate under one task organization. Of the seven known pairings experiments, six showed the expected results. Even though the correct ModSAF behaviors were selected, many of the closeness values were quite low. This is an illustration of how simulations that appear similar externally can actually be very different in their internal semantics. As mentioned previously, the one failed experiment was not due to an error in the correlation algorithm but due to the drastic difference in robustness of the supposedly same behavior. The five unknown pairings produced arguably acceptable results when considering that there was no corresponding ModSAF behaviors for these CCTT behaviors. Often 100% interoperability of simulations requires complete reengineering of one of the simulations to the extent that it is no longer beneficial to use two different simulations.

This research has shown that a less aggressive form of arbitration with a simple behavior representation can indeed satisfactorily correlate behavior in most cases. This has the potential to reduce the SAF operator workload in large-scale exercises. As the state of the art in CGF increases, these semantic interoperability issues will become the dominant factor in the pursuit of large-scale and joint exercises. This research is but the first step towards the heterogeneous simulations of the future.

FUTURE WORK

The focus of this research has been on the arbitration algorithm and its supporting components. The actual run-time interfaces and parameter conversion routines have yet to be developed. There were also were several issues addressed in the arbitration algorithm. Specifically, how to handle source parameters that correlate to more than one destination parameter equally, and destination parameters that correlate to more than one source behavior. Both can cause unexpected behavior when the behavior is executed with these parameter conversions. Also, more research is required to study when to allow parameters default instead of being correlated. Of course, if 100% correlation is desired then an extension of this work is needed that allows simulations to be data driven and share behavior primitives.

ACKNOWLEDGMENTS

The authors would like to thank Christina Bouwens for her support on this paper.

REFERENCES

Altman, Marty; Kilby, Mark; Lisle, Curtis, "On a Shared Environment Concept for Distributed Simulation," Proceedings of the 11th Workshop on Standards for the Interoperability of Distributed Simulations, Institute for Simulation and Training, Orlando, FL, Sept 26-30 1994, pp. 535-543.

Dean, Christopher, "The Semantic Correlation of Behavior for the Interoperability of Heterogeneous Simulations", Masters Thesis, University of Central Florida, May 1996.

DIS Steering Committee, "The DIS Vision: A Map to the Future of Distributed Simulation Version 1", Institute for Simulation and Training, May 1994.

Ourston, Dirk; Blanchard, David; Chandler, Edward; Loh, Elsie, "From CIS to Software," Proceedings of the Fifth Conference on Computer Generated Forces and Behavioral Representation, Institute for Simulation and Training, Orlando, FL, May 9-11 1995, pp. 275-285.

Smith, Roger D., "The Conflict Between Heterogeneous Simulations and Interoperability," Proceedings of the 17th Interservice/Industry Training Systems and Education Conference (CD-ROM), The American Defense and Prepardness Association, Albuquerque, NM, November 13-16, 1995.