Multi-Resolution Modeling in the JTLS-JCATS Federation

Andy Bowers
David L. Prochnow
The MITRE Corporation
7515 Colshire Drive
McLean, VA 22102
bowersa@mitre.org
prochnow@mitre.org

Keywords:
JTLS, JCATS, multi-resolution modeling, MRM

Abstract: *The JTLS-JCATS federation satisfies a warfighter requirement for multi-level training, as articulated in MSIAC's Warfighter M&S Needs Assessment. In the process of developing the federation, the federation team has gained insight into Multi-Resolution Modeling (MRM) necessitated by federating two fundamentally different simulations. The Joint Theater Level Simulation (JTLS) is a hex-based, time-stepped simulation using Lanchesterian equations to adjudicate conflict between aggregate-level objects, typically battalions or brigades. The Joint Conflict and Tactical Simulation (JCATS), on the other hand, is an event-stepped simulation that uses terrain posts, and adjudicates combat between entity-level objects, typically individual vehicles or combatants, using individual shot probabilities. The JTLS-JCATS federation leverages the strengths of each simulation by allowing objects to be represented at the level of resolution needed for achieving training objectives. The federation allows the control of federation objects to pass from one simulation to the other, and it employs mechanisms that permit objects represented at different levels of resolution – and controlled by the different simulations – to interact with one another.*

As reported in a Fall 2002 SIW paper, the federation objectives are to provide a useful capability to the Joint Warfighting Center (JWFC) to support its exercise support program and to document lessons learned in MRM for the benefit of future programs. Since publishing the previous paper and completing a successful prototype federation, the federation has undergone extensive integration testing and an acceptance test process required by the JWFC. Acceptance testing placed the federation in the hands of the JWFC Instructor/Controllers (ICs) and, as a result, provided valuable feedback for exercise support use.

This paper addresses work completed subsequent to our last published paper and focuses on several topics. The authors discuss multi-resolution modeling (MRM) in the JTLS-JCATS federation, including use of shared object ownership as an enabler of MRM. In addition, the paper addresses reuse of federation components for the JTLS-JCATS federation, and potential reuse for future training systems. The paper also emphasizes exercise support enhancements suggested by JWFC ICs

1. Introduction

JTLS-JCATS federation development began in 2002 with the dual objectives of meeting joint training requirements and exploring the concept of Multi-Resolution Modeling (MRM). The federation utilizes the High Level Architecture to link the Joint Theater Level Simulation (JTLS) and Joint Conflict and Tactical Simulation (JCATS). The integrated system will be capable of supporting multi-level exercises integrating large-scale theater operations with small unit and individual combat system actions. In addition, this federation is designed to support analysis of alternative combat systems concepts and employment strategies. A prototype, depicted in Figure 1, was developed and demonstrated in fall 2002. The architecture and fundamental concept was approved for subsequent development with expanded goals. Post prototype development objectives anticipate an integrated system that will provide a capability to concurrently or separately exercise traditional Joint Force staff training audiences together with real-time, warfighter-in-the-loop, live and virtual forces at user-selectable levels of resolution from the entire force to individual systems [BOWERS].

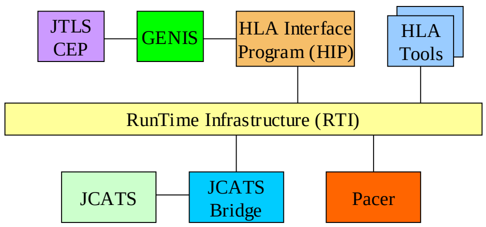

Figure 1. JTLS-JCATS Federation

Over the past year, federating the Joint Theater Level Simulation (JTLS) and the Joint Conflict and Tactical Simulation (JCATS) has continued to surface MRM issues. As defined in a Fall 2002 SIW paper [Bowers], MRM "represents aspects of the real world at more than one level of detail." Indeed, the rationale for the Defense Modeling and Simulation Organization's (DMSO) interest and sponsorship of the federation was, and is, to report lessons learned and make recommendations to the M&S community on MRM. As the federation has matured, the development team has faced fewer challenges related to basic integration and more issues associated with the fundamental differences in the level of resolution inherent in each of the federates.

Coincident with surfacing and solving MRM issues, the development team has increasingly focused on ensuring a useful product for the customer, and co-sponsor, the Joint Warfighting Center (JWFC). JWFC's interest and sponsorship of the JTLS-JCATS federation stem from its charter to support Combatant Commander exercise programs. JWFC has historically accomplished this with a variety of tools, to include JTLS and JCATS as separate simulations. The JTLS-JCATS federation adds another tool "to provide the CINCs a multi-resolution federation of existing simulations that can be used until JSIMS is fielded with the necessary functionality to conduct operations in today's asymmetrical warfare environment." [DMSO] Key to ensuring a useful product was the involvement of JWFC Instructor/Controllers (ICs) in testing federation functionality. The ICs' input resulted in new requirements and changes to existing requirements.

This paper starts with an overview of changes in the federation as a result of technical maturation, including new and evolving requirements. We then address Multi-Resolution Modeling and include specific examples of challenges faced by the development team in federating JTLS and JCATS. A section on reuse follows, addressing both reuse that has supported the development team's efforts, and the potential for reuse of the team's products for future federation development. We next address the impact of the JWFC ICs by discussing their input resulting in new and changed requirements. We conclude in section five offering lessons learned and recommendations to date.

1.1 Overview of JTLS-JCATS Changes since Fall 2002

During August 2002, the federation development team conducted a demonstration of the federation prototype for DMSO and JWFC representatives. The purpose of the demonstration was to provide sufficient insight into federation capabilities that the representatives could decide to continue, or curtail, federation development. At the time, the prototype had limited capability:

- Time management worked, but redundant time advance requests (i.e. when an advance request was already pending) limited game speed.
- Object reflection worked for some object types; others supported by the FOM hadn't been tested.
- Object transfer worked, but not as desired; a JTLS unit transferred to JCATS would result in JCATS instantiating all entities within the unit separately rather than the unit as a whole
- Direct Fire between JTLS and JCATS objects worked for limited numbers of objects, primarily due to inadequate data mapping.
- Area Munitions worked for one bomb type dropped by JTLS aircraft on JCATS units; other area munitions hadn't been tested.

In summary, the nature of the majority of the problems at the time was attributable to the early stage of federation development and the insufficient time spent to date on integration. Acknowledging the current limitations of the prototype, the representatives nonetheless recognized the potential of the federation and approved continued development.

Since that time, the development team has successfully resolved each of the above issues, as well as many other limitations. Additional integration time has allowed more data mapping and more testing of a variety of object and data types. In many cases, including new object and data types has surfaced new MRM challenges and opportunities. These are summarized in section 1.2 below. Additionally, the team has had time to develop other, more complex, capabilities. Key among these is a save and restore capability, a necessity for supporting an exercise conducted twenty-four hours a day for a week or more. Another example, discussed in section 1.3, is migrating away from the "box-in-box" implementations of the JTLS and JCATS playboxes around which the federation was developed.

1.2 New JTLS-JCATS Functionality

During prototype development, we postponed developing naval functionality, an obvious shortfall in a Joint federation. The Ship object class existed in the FOM and was the start point for implementing naval functionality. Ship objects can now be initialized in each of the simulations, reflected by each, and transferred from one to the other. We still have FOM modifications to make and test before damaging or destroying a ship in one simulation by munitions originating in the other simulation is possible.

Another object class now providing functionality within the federation is Target. Again this was a FOM supported object class with which we had not yet worked. The JTLS standard database defines more than twenty Target types. We started implementing Targets in the federation by including four of the more than twenty: Anti_Air_Artillery, Sensor_Site, Supply_Storage, and Surface_To_Surface_Missile. The federation can now support reflection, ownership transfer, and cross-RTI damage/destruction of the four types.

With the inclusion of JCATS recognition and use of Supply_Storage targets, implementing supply and resupply functionality became development objectives. These are domains in which the JTLS-JCATS federation aptly demonstrates the advantages in MRM and are discussed in more detail later in the paper.

Unit, an object class addressing the representation of ground units, also demonstrates the advantages in MRM. The development team not only corrected the inadequacies of the implementation of the Unit object class during the prototype, but also enabled an expanded capability for ownership transfer beyond that initially requested for the prototype. We will discuss unit representation as an example of MRM in section 2.

1.3 Migration Away From Box-in-Box Approach

One of the most significant changes in the post-prototype federation has been the gradual diminishment of the importance of the "box-in-box" implementation. The box-in-box conceptual model was agreed upon during the Federation Development and Execution (FEDEP) process as an initial means of addressing the fundamental terrain and algorithmic differences between JTLS and JCATS. Briefly, box-in-box required JTLS representation of a theater-sized geographic area within which JCATS would represent a smaller area of tactical interest. The concept played to the strengths of each of the simulations and, with an associated concept of operations (CONOPS), minimized the possibility of fair-fight issues arising from simulation differences. (A more complete explanation of box-in-box is contained in [BOWERS].)

Though the initial conceptual model suggested that JTLS and JCATS objects would be "separated" geographically by the box-in-box approach, early federation agreements acknowledged the usefulness of allowing some JTLS-owned objects inside the JCATS box. Initially these objects were all Air_Mission objects, one or more of the same type of aircraft with the same mission. In fact, the *only* Air_Mission objects destined for ownership transfer to JCATS would be those requiring aircraft sensor utilization of the JCATS line-of-sight (LOS) algorithm and the JCATS terrain: Close Air Support, Armed Reconnaissance, and Reconnaissance missions.

As the federation matured, three realizations caused our reassessment of the implementation of box-in-box. First, we discerned that there were other object types that JTLS should own even though instances of the object were located in the JCATS box. Secondly, we simplified the implementation of the box-in-box from our initial concept requiring a two-hex "boundary" separating areas with different object control rules, really a box-in-box-in-box, to a single boundary. Finally, the JWFC IC use of the federation suggested that, with additional development, we could do away with the box-in-box altogether and rely upon CONOPS to address unresolved terrain and algorithmic differences.

We reconsidered our agreement limiting JTLS-owned objects in the JCATS box to select air missions because of the previously discussed Target object class. During integration testing we realized that an international airport, located inside the JCATS box, was the site of numerous JTLS targets associated with the runways, aircraft shelters, and facilities. These targets, along with their associated BE numbers and data, would not be represented unless either the agreements were changed to allow JTLS-owned targets in the JCATS box or these three additional target types were added as transferable objects.

Simplifying the implementation of the box-in-box occurred because we found that during acceptance testing, the testing done by the JWFC ICs, the problems we anticipated occurring across boundaries between objects owned by JTLS interacting with objects owned by JCATS didn't occur for two reasons. First, the ICs preferred to manually transfer objects from one simulation to the other. Since they had control of the timing of the transfers, they were less likely to rapidly repeat multiple transfers of the same object, a potential issue for automated object transfer. Secondly, the FOM has no provisions for JTLS unit to engage in ground combat with a JCATS unit, so multiple boundaries aren't needed to prevent that occurrence.

Doing away with the box-in-box altogether is closer to realization, but still in the future. Key steps in the process included implementation by Rolands and Associates (R&A), the JTLS developers, of a boolean flag allowing JTLS to transfer objects ownership of objects during initialization based on geographic area. With the flag set to "false," the JTLS users could instead transfer selected objects in the JCATS box during initialization and the federation would start with both simulations owning objects in the JCATS box. This implementation had repercussions on JTLS objects being damaged or destroyed in the box, which are being resolved as of this writing, but the capability exists for initializing the federation with objects owned by each federate – a key federation requirement.

From the JCATS point-of-view, the JCATS Bridge, which serves as middleware between JCATS and the RTI, uses the box to distinguish objects it should notify JCATS about for purposes of reflection or transfer and those that JCATS doesn't need to be aware of. JCATS is informed of objects inside the box, but not informed of objects outside the box. The latter is important for practical reasons. We are federating JTLS and JCATS so that we can represent a theater-level scenario within which tactical-level actions occur. The theater-level scenario, with potentially millions of people and pieces of equipment, must necessarily be represented at a different level of aggregation than the tactical actions involving a few thousand or even tens of thousands of people and items of equipment. The JCATS Bridge accepts all updates from the RTI for objects in the FOM. It therefore knows about all instances of objects throughout the entire JCATS terrain file. If the JCATS and JTLS terrain files are the same size, the JCATS Bridge knows about all of the objects owned by JTLS. If the JCATS Bridge told JCATS to represent all of these objects at the entity level, JCATS would attempt to instantiate millions of entities and, as a result, would exhaust all the memory of its host machine. By distinguishing between objects inside the box and those outside the box, JCATS avoids this issue. LLNL, the JCATS developer, is developing a capability to represent the JTLS objects outside the box, essentially reflecting the JTLS aggregate, not the systems comprising the aggregate. When this capability is complete, and when R&A resolves damage to JTLS objects regardless of location, we will remove the box completely, relying entirely upon CONOPS to address unresolved terrain and algorithmic differences.

2. Multi-Resolution Modeling in the JTLS-JCATS Federation

2.1 Advantages of Multi-Resolution Modeling

Our Fall 2002 paper explains the rationale behind the JTLS-JCATS federation, including the advantages of multi-resolution modeling. For that reason, it will just be summarized here. The ultimate objective for this federation is to produce an integrated JTLS and JCATS simulation system capable of supporting multi-level exercises integrating theater operations with urban, small unit, and entity system actions by simulating forces at user-selectable levels of resolution from the entire force to an individual system, concurrently or separately, to support the joint simulation community.

2.1.1 MRM of Objects in General

"User-selectable levels of resolution" are thus far most noticeable in the ground domain. Both JTLS and JCATS represent air, sea, and space objects as individual platforms or, at most, a few platforms. The requirements for MRM in these domains fall therefore, not in the representation of object types, but rather in functionality. Functional MRM will be addressed in 2.1.4. Objects in the ground domain, in particular ground units, afford ample opportunity for MRM because ground unit representation is so different in JTLS vice JCATS.

Firstly, JTLS ground units are either Aggregate Resolution Units (ARU), for example an Army Artillery Battalion or a Marine Rifle Company, or High Resolution Units (HRU), such as SOF teams. In both cases, JTLS maintains a list of the systems comprising the unit. In JTLS version 2.5, the current version, a combat system cannot be separated from its parent unit and represented as a distinguishable object to which the user can give commands. In JCATS, units are comprised of systems that can be separated into distinguishable objects and given commands.

Secondly, algorithmic differences between JTLS and JCATS in conflict adjudication are solely in the ground domain. JTLS adjudicates combat between opposing ARUs using Lanchesterian equations. JCATS adjudicates direct fire combat between all individual systems using probability of hit (Ph) and probability of kill (Pk). JTLS uses Ph/Pk for weapons fired by air and naval forces, so the two simulations handle adjudication similarly in these domains.

The practical result of the algorithmic differences is that JTLS ground units typically are comprised of generic system types in comparison to more detailed system types found in JCATS units.

2.1.2 Example of MRM in Object Representation

Table 1 shows a data mapping of an HRU, Operational Detachment Alpha (ODA) 745, an Army Special Forces A-Team. JTLS lists the majority of combat systems comprising the ODA as combat troops, as one would expect, but does not distinguish between each of the combat troops. Each contributes equally to the combat power of the unit as a whole. Other systems in the unit, MG-AGL for example, would contribute a different amount to the combat power of the unit.

JTLS SYSTEMS	#	JCATS SYSTEMS	#
ODA 745		ODA 745	
CBT-TROOPS	11	SOF CDR	1
		SOF OPS SGT	1
		SOF COM SGT	1
		SOF DEMO SGT	2
		SOF MEDIC	2
		SOF SNIPER (12.7MM)	2
		SOF SPOTTER	2
MG-AGL	1	SOF WPN SGT (M249)	1
66MMRKT-AT4	2	JTLS ONLY	0
EXPLOSIVES	3	JTLS ONLY	0

Table 1. MRM of a High Resolution Unit

JCATS distinguishes between far more of the systems in the unit. This is because the systems may own different weapons, e.g. the sniper, which affects the Ph/Pk for that system against other systems, or possess different capabilities, e.g. the medic if casualty play is important, or a variety of other reasons. Even the JCATS systems showing multiple entries, e.g. the two demolitions sergeants, could be modeled as two distinct systems if there were a reason to distinguish between them – exercise of a personnel replacement system, for example.

When ODA745, a named unit in the JTLS database, is transferred to JCATS, the representation of ODA745 in the federation as a whole changes from the level of detail shown in the left two columns of table 1 to that shown in the right two columns. This is MRM. JCATS controls many of the attributes of ODA745 so, for example, the unit can be deaggregated and the members placed in particular positions to take advantage of the JCATS terrain and LOS.

2.1.3 Types of Unit Data Mapping

ODA745 represents one of the three types of data mappings supported by the federation. In this case, the JTLS object name, ODA745, matches a corresponding JCATS unit, i.e. ODA745, and JCATS is able to represent not only the systems in the unit, but the complete hierarchy of the unit. If ODA745 did not exist as a named unit in the database, JCATS would next check the Prototype_Name, an attribute for the name of the unit prototype, against predefined JCATS prototype units. If a match occurs, JCATS can again represent the hierarchy of the "generic" JTLS unit. If no match is made using the Prototype_Name, JCATS will represent the object with the correct attributes, but will not be able to disaggregate the object in a hierarchical manner. Disaggregation to the system level would, of course, still be possible.

The importance of the hierarchy is not evident in the simple ODA example shown in table 1, but is clearly evident in larger ground units. For example, a JTLS armor battalion might include in its system list fifty-four tanks. A JCATS unit representing the battalion's hierarchy would be able to deaggregate the battalion into a headquarters company and three tank companies, further deaggregate the tank companies into a headquarters platoon and three tank platoons, and further deaggregate the platoons into individual tanks. The user has the choice of maneuvering and positioning the unit at any of the different levels of the hierarchy within the unit. Mapping a JTLS unit to a JCATS hierarchical unit is one method of providing "user-selectable levels of resolution."

2.1.4 Example of MRM in Supply Representation

MRM is not only, however, a function of object representation. JTLS and JCATS differ functionally in many areas and these provide opportunities for implementing MRM while, more importantly, providing improved functionality to the user.

For example, JTLS and JCATS differ in their representation of supplies. Depending on the training audiences desires for supply representation, and the resulting database build, JCATS objects consume fuel and ammunition, classes III and V respectively, and carry realistic quantities of each. JCATS also provides functional use of barriers like wire or sandbagged positions, but doesn't explicitly link the use of these to consumption of the class IV supplies necessary to construct them. JCATS obviously represents major end items, like tanks, but does not typically represent the supply chain replacement of class VII. JTLS training audiences are more likely to exercise supply and resupply, albeit at the operational or theater level, so JTLS enables consumption of each of the classes of supply I through X. Although JTLS represents some supplies, e.g. classes I and II, that JCATS doesn't, JCATS typically represents far more ammunition types than does JTLS.

We capitalized on this difference in resolution with respect to supplies in the JTLS-JCATS Federation by ensuring that units originating in JTLS transferred to JCATS continue to consume all classes of supplies represented in JTLS. The means of doing so is shared object ownership and is discussed in 2.2.2.

2.1.5 Accounting for Algorithmic Differences in MRM

MRM must necessarily resolve algorithmic differences between the two simulations. The differences in conflict adjudication, Lanchesterian vs Ph/Pk, have already been noted. Our "resolution" of this difference is adequate as ownership transfer allows the combat to occur in one or the other simulation rather than between the two.

A decidedly more elegant resolution of algorithmic differences between the two simulations occurred in resolving the effects of indirect fire, or any type of area munition, between JTLS and JCATS. Although the FOM interaction includes all necessary and relevant information [Bowers], the effects of the interaction were vastly different depending on whether JCATS area munition(s) were detonating in JTLS or vice versa.

For example, a JTLS artillery battalion firing a battalion ten (ten rounds fired by each howitzer) into JCATS would generate one interaction including, among other parameters, the number of rounds, 180. JCATS receives the single interaction and adjudicates the effects of all 180 rounds as though they were delivered simultaneously, an artilleryman's perfect massed-fire mission.

When a JCATS artillery battalion fired a battalion ten into JTLS, JCATS, abiding by federation agreements, sent an interaction to JTLS for each artillery round. Since the JTLS algorithm calculates attrition based on the number of area munition rounds over a database-set time period, the results of individual artillery rounds were inconsequential.

R&A implemented a solution to group individual rounds, over a database-set time period, based on the logically deduced target unit. The solution accounts for multiple fire missions by multiple units landing in proximity of one or more target units. The solution allocates the effects on target without regard to the firing element, but instead based on location of impact in relationship to the target and the munition type, a realistic implementation. Figure 2 graphically portrays R&A's design for "bundling" area munitions.

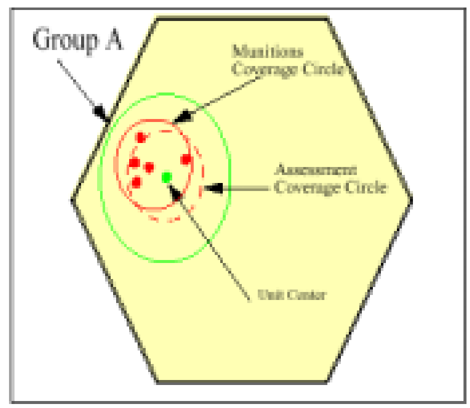

Figure 2 JTLS Area Munition Bundling

2.2 Shared Object Ownership as an Enabler of MRM

The multi-resolution modeling within the JTLS-JCATS federation is made possible by the attribute ownership services within HLA. By exploiting these specialized RTI services, the JTLS-JCATS federation has the capability to transfer control of objects from one simulation to another. That is, JTLS objects can be transferred to JCATS when more detailed modeling of an object is needed, and similarly, objects can be transferred from JCATS to JTLS when a more aggregate representation of the object is appropriate.

To be more precise, using HLA, it is actually the ownership of attributes (the defined components of an object) that is transferred from one simulation to another. Because of different model representations in

JTLS and JCATS, it does not make sense to transfer an object attribute that has no meaning in the other simulation. Therefore, in many cases, it is a subset of the attributes of an object instance that are transferred.

In his Spring 2003 paper, Hill makes the point that very few distributed simulation systems take advantage of the powerful capabilities of object ownership transfer. [HILL] The JTLS-JCATS federation is a notable exception, as attribute ownership services enable MRM within the federation. Hill also discussed the technical difficulties associated with implementing ownership services, primarily identifying shortcomings with doing so using the Distributed Interactive Simulation (DIS) protocol. The JTLS-JCATS federation is proof that ownership transfer can be successfully implemented using HLA, though it too has some complicating features.

HLA allows for both *push* and *pull* methods of transferring attributes. In addition, an attribute push may be *unconditional* or *negotiated*. The JTLS-JCATS federation implements ownership transfers using an unconditional push. One challenge of using this mechanism is ensuring that object attributes do not become unowned for an extended period of time. Obviously, attributes will not be owned by any federate for a very brief time during the attribute transfer process, and that is a tolerable condition. However, if a problem occurs in the transfer process causing an object's attributes to remain unowned, then this is a very serious problem. In this case, neither federate takes control of the object, and it become a sort of zombie. Although the RTI has services for querying attribute ownership, the JTLS-JCATS federation does not invoke them, and it is unclear that use of these services would be practical.

2.2.1 Shared Object Ownership of Units

As previously discussed, JTLS and JCATS represent surface units at different levels. JTLS represents aggregate objects while JCATS represents objects at the entity level. An early design issue for the JTLS-JCATS team was whether to pass data across the RTI at the aggregate level or at the entity level, or both. Passing data at the JTLS level would allow use of the previously existing Federation Object Model (FOM) associated with JTLS but would require that any aggregation or disaggregation of objects occur within JCATS. Passing data at the JCATS level (i.e. entity level) would require extensive FOM changes and would force JTLS to disaggregate its objects prior to sending updates. A third option was to publicize object data at both aggregate and entity levels. This would have the advantage that another federate could subscribe to whatever level it was interested. However, having the same simulation data represented multiple times, albeit in different ways, would add a level of complexity that was beyond the scope of our work.

Ultimately, the federation team decided to pass data across the RTI at the aggregate level. This maximized reuse and minimized the traffic that had to flow over the RTI. This put the onus on JCATS to perform disaggregation and to re-aggregate data when updating the disaggregated entities. During the effort, JCATS was modified to also represent units as aggregates, and this alleviated much of the burden on JCATS. With this change, JCATS only has to disaggregate an object when a player determines that a greater level of detail is required for a particular operation.

Because data passed over the RTI for most surface objects is at the JTLS level, JCATS must update that object at the same aggregation level after it takes ownership of an object's attributes even if internally it disaggregates the object. If the JCATS user transfers the object back to JTLS, it must obviously be transferred back at the same level.

As previously discussed, JTLS keeps a list of systems in the surface object. These systems typically represent either major end items of equipment or personnel. The equipment-related attributes are represented in the FOM as complex datatypes of variable sizes. Because JTLS and JCATS do not track all of the same individual equipment types, it does not make sense to pass ownership of these attributes. Instead, JTLS always owns these attributes. The data type for the number of each system is real. Before JTLS transfers control of a surface object, it updates another attribute that represents the integer number of manned systems. JCATS uses this integer to instantiate the correct number of combat systems in the unit. When a combat system is damaged or destroyed in JCATS, JCATS sends an HLA interaction to communicate state changes of the individual equipment types that it represents. JTLS in turn makes a corresponding update to the appropriate system list.

Similarly, when a JTLS surface object is transferred to JCATS, JTLS retains ownership of attributes associated with the number of personnel in the object. For purposes of personnel accounting, infantry is explicitly counted while vehicle crews are implicitly represented. If explicitly represented personnel become casualties in a surface object, JCATS sends an interaction to JTLS to pass this information. In turn, JTLS updates the relevant personnel list, to include calculating crew casualties for vehicles that have sustained damage.

Other attributes of the unit objects, location for example, are owned by JCATS. Sharing ownership of object attributes enables MRM because without it repeated transfers of a unit would cause repeated changes to the number of combat systems in the unit as the number changed from real to integer and back. Figure 3 graphically portrays this problem by suggesting successive time increments during which an armor battalion is passed from JTLS to JCATS, deaggregated, and takes casualties before being passed back to JTLS. It takes casualties in JTLS before being again passed to JCATS.

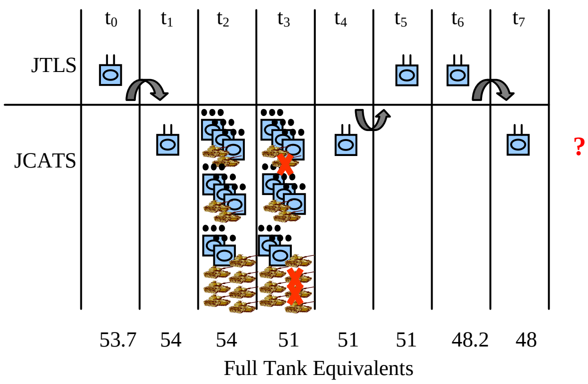

Figure 3 Repeated Object Transfer

Shared object ownership enables MRM by avoiding repeated changes to the number of combat systems in the unit passed from JTLS to JCATS.

2.2.2 Shared Object Ownership of Supplies

As was discussed in 2.1.4, JTLS and JCATS differ in their representation of supplies. It clearly does not make sense to pass ownership of supply attributes from JTLS to JCATS for supply types JCATS does not represent. So, as in the case of combat systems, class VII major end items, JTLS retains ownership of other supply attributes. JCATS consumes classes III and V and reports on-hand quantities using an interaction. JTLS updates the attribute values and also uses the current values and reorder levels, a JTLS parameter, to know when to start resupply of the unit. MRM is enabled by shared object ownership of supplies because the federation as a whole better represents the consumption of supplies than either simulation does alone.

3. Reuse of JTLS-JCATS Federation Components

The JTLS-JCATS federation was constructed quickly and efficiently, largely due to reuse of many of the simulation components. More importantly, the JTLS-JCATS federation is ripe for further reuse, and new functionally could be added through the addition of special-purpose federates.

3.1 Federation Components Currently Being Reused

As indicated in our prior paper, the federation design incorporated elements from the prior HLA experiences of both JTLS and JCATS [BOWERS]. In the past, JTLS was in federations with several systems, including the Enhanced Air Defense Simulation (EADSIM), the Global Command and Control System (GCCS) and NATO C4I systems [PROCHNOW]. JCATS had participated in Millennium Challenge 2002 (MC02), and therefore already had an HLA interface. In addition, the JTLS-JCATS Federation Object Model (FOM) came largely from previous JTLS federation usage. Although the FOM has been modified for additional functionality in JTLS-JCATS, the core object representation is largely unchanged.

In addition to the core simulations, the JTLS-JCATS federation also reused some important auxiliary components as well. It employs the Pacer federate from the JTLS-EADSIM federation for the purpose of controlling the game speed, and it uses the Federation Management Tool (FMT) for monitoring federation execution.

As the federation this year has evolved to being used by experienced JTLS and JCATS users, there is another type of reuse that was not highlighted in our earlier paper but is highly advantageous. That is the reuse of the simulation interfaces. Despite the new multi-resolution modeling capability, the JWFC users can use the JTLS and JCATS systems on which they have been trained with few differences. For instance, the graphical display of the battle situation on the JTLS user's GIAC display looks the same. Even if a JTLS object has been transferred to JCATS, it still shows up on the JTLS map display as an aggregate, even if the JCATS user is viewing a disaggregated view of the same object on his display. There is a difference that if the JTLS user attempts to send an order to an object being controlled by JCATS, he receives a response indicating that the object is not under JTLS control. Similarly, the JCATS user interacts with the simulation in the same manner that it had before. In both JTLS and JCATS, users can manipulate all objects under the respective simulation's control, and whether the object originated in JTLS or JCATS is transparent to the user.

3.2 Potential Reuse for Future Training Systems

The JWFC has large plans for the JTLS-JCATS federation, while at the same time it recognizes that this federation will not achieve all training objectives. The JTLS-JCATS federation should be considered a serious candidate in the analysis of JSIMS alternatives required by the Program Decision Memorandum (PDM). The JTLS-JCATS federation will also be part of the Joint National Training Capability (JNTC) effort now underway in JWFC.

In whatever capacity it is used, the JTLS-JCATS federation can be adapted for additional federates to meet any special training objectives. The JTLS and JCATS developers have proven that they can use their systems in multiple federations.

4. User-Requested JTLS-JCATS Exercise Support Enhancements

The JWFC ICs participated in the most recent tests of the JTLS-JCATS federation as subject matter experts (SMEs) in operational domains, being retired or reserve military, and in exercise support, being members of the JWFC Support Team. All of their recommendations, when implemented, will clearly improve the federation's capability to support exercises. The recommendations fall logically into two groups: functional improvements and exercise support enhancements. This section addresses both types of requested modifications.

4.1 Functional Improvements

The JWFC ICs found the transfer of weapon loads for air missions from JTLS-to-JCATS inadequate for exercise use. Assigning weapons loads is done very differently in the two simulations and the mapping from JTLS to JCATS currently does not always result in the weapon load launched in JTLS being used by the transferred aircraft in JCATS.

The JWFC ICs recommended developing a capability to transfer objects with embarked units, e.g. a C-130 with airborne troops, and units with attached or detached elements, e.g. a mech.-heavy task force. This functionality has obvious application for both airborne/airmobile and amphibious operations and we have added it to the list of future enhancements.

As a means of avoiding transferring objects with embarked units, the JWFC ICs recommended enabling ownership transfer outside the JCATS box. This would allow combat loading of aircraft or amphibious platforms in JCATS outside the JCATS box. The user could then give the loaded object movement orders, in JCATS, to move into the JCATS box to execute the airborne/airmobile or amphibious operation. This feature contributed, of course, to the discussion on migrating away from the box-in-box representation previously presented.

4.2 Exercise Support Enhancements

The JWFC ICs unanimously preferred manual to automated ownership transfer. Moreover, they requested individual control of the timing of ownership transfer. They could plan and execute the timing, for example, for transferring an air mission from JTLS to JCATS at the Initialization Point (IP) if the JTLS user could send the order transferring the mission while verbally informing the JCATS user of the hand-off. R&A developed a player order to effect the transfer in time for the final test by the ICs, LLNL is in the process of developing the order as this paper is being written.

The JWFC ICs requested use of existing JTLS reports for JCATS-owned objects. The ICs are familiar with the JTLS reports and use the reports to improve the quality of message traffic, both voice and digital, to the training audience. They were interested, therefore, in being able to access information on JCATS-owned objects similar, and in the same format, to the information they were able to obtain on JTLS-owned objects. JTLS users are already able to obtain some reports on JCATS-owned objects. The ICs requested thirteen additional reports be accessible in JTLS on JCATS-owned objects.

The JWFC ICs recommended specific criteria for federation performance, suggesting a 4:1 ratio of elapsed simulation time to elapsed real time as being desirable to run the federation. Although during exercises a 1:1 ratio is used, the faster game speed is desired for periodically accelerating the game clock. For instance, some exercises may employ a 12-hour exercise day, and technical control may need to advance the simulation time at the end of a day to reach a suitable starting point for the next simulation day. In other cases, the game will be run faster than normal to recover from a crash. During federation testing, the federation has run at a game ratio of 1.7: 1, or not quite twice as fast as wall clock time, with a test scenario of around 3,000 JTLS objects. Until recently, basic functionality has been the focus rather than game speed, so additional time will be spent improving game speed.

The JWFC ICs started development of a Federation rules and procedures document covering the use of the federation in an exercise environment. The document will include some topics that are currently covered in the Fed Agreements Doc, e.g. initialization sequence, and topics more germane to exercise use, e.g. procedures for ownership transfer (operational, not technical, implementation), etc.

5. Conclusions

Over the past year, the JTLS-JCATS federation has evolved from a promising prototype to a system that is close to meeting critical joint training requirements at the JWFC. With the demise of JSIMS and the subsequent uncertainty in future joint training, the JTLS-JCATS federation is the premier candidate for supporting exercises requiring MRM.

Involving JWFC ICs and SMEs in the integration testing and development cycle has significantly improved the federation's value for future JWFC exercises. Firstly, JWFC ICs and SMEs suggested improvements resulting in software changes providing additional functionality in the federation as a whole. Secondly, their involvement allowed parallel evolution of a Concept of Operations and operating procedures for use of the federation during exercise support.

The timing of JWFC ICs involvement was good. Involve them too early and integration isn't adequate to support operational scenarios of sufficient size and complexity that they are able to appreciate using the functionality of both federates in one construct. Involve them too late and you forgo their expertise in exercise support requirements in contributing to developing a useful, and usable, federation.

Although the initial scenario and conceptual model agreed upon during early stages of the FEDEP to guide development of the prototype were extremely useful [Bowers], maturation of the federation has required changes to these initial concepts and agreements. The lesson learned is that early FEDEP work is valuable, but one should anticipate and plan for change.

Data mapping continues as an area of critical concern as it promises significant effort during exercise preparation for typical JWFC-hosted exercises. We have recommended an automated data mapping capability be developed to support not only this project, but other JWFC-sponsored efforts as well.

The HLA was an extremely effective means of integrating the JTLS and JCATS. Shared object ownership was a critical HLA capability in enabling MRM and therefore the technical success of the JTLS-JCATS federation development. HLA also enabled the critical success from a user standpoint; the operator's use of each simulation has changed little as a result of federating even though the capabilities of the federation as a whole are significantly more than the sum of the separate simulation capabilities.

Acronyms

ARU	Aggregate Resolution Unit
BE	Basic Encyclopedia
CEP	Combat Events Processor
CONOPS	Concept of Operations
CSL	Conflict Simulation Laboratory
FMT	Federation Management Tool
FOM	Federation Object Model
GCCS	Global Command and Control System
GIAC	Graphical Input Aggregate Control
HIP	HLA Interface Program
HRU	High Resolution Unit
IC	Instructor/Controller
JCATS	Joint Conflict and Tactical Simulation
JFCOM	Joint Forces Command
JNTC	Joint National Training Capability
JTLS	Joint Theater Level Simulation
JWFC	Joint Warfighting Center
LLNL	Lawrence Livermore National Laboratory
LOS	Line-Of-Sight
MRM	Multi-Resolution Modeling
ODA	Operational Detachment Alpha
Ph	Probability of Hit
Pk	Probability of Kill
R&A	Rolands and Associates
SME	Subject Matter Expert
SOF	Special Operation Force

Bibliography

[BOWERS] Bowers, A., Prochnow, D., Roberts, J., "JTLS-JCATS: Design of a Multi-Resolution Federation for Multi-Level Training," Proceedings of the Fall 2002 Simulation Interoperability Workshop, Orlando, Florida, September 8-13, 2002.

[DMSO] Defense Modeling and Simulation Office (DMSO)/Modeling and Simulation Information Analysis Center, "Warfighter M&S Needs Assessment of the Unified Commands and Selected Supporting Commands," 17 November 2000

[HILL] Hill, F., "Transfer Ownership – A Marvelous Idea That Is Sitting Out the War," Proceedings of the Spring 2003 Simulation Interoperability Workshop, Kissimmee, Florida, 30 March – 4 April 2003.

[PROCHNOW] Prochnow, D., Furness, C., Roberts, J., "The Use of the Joint Theater Level Simulation (JTLS) With the High Level Architecture (HLA) to Produce Distributed Training Environments," Proceedings of the Fall 2000 Simulation Interoperability Workshop, Orlando, FL, September 17-22, 2000.

Author Biographies

ANDY BOWERS is a Senior Modeling and Simulation Engineer at MITRE Corporation's Joint Warfighting Center site. A retired U.S. Army officer, Mr. Bowers holds a Bachelor's degree from the United States Military Academy and a MS in Operations Research and a MS in Civil Engineering from Stanford University.

DAVID L. PROCHNOW is a Lead Software Engineer in the Information Systems and Technology Division at the MITRE Corporation and is currently the technical lead on several High Level Architecture programs. At MITRE, Mr. Prochnow previously contributed to the ALSP project and to logistics simulation efforts. While at BDM International and Control Data Systems, Mr. Prochnow developed software for various corps-level and theater-level wargames. He received a B.S. in Computer Science from the University of Virginia in 1983.