A New Approach To Live Entity Interaction Via The HLA

John W. Shockley
Reginald E. Ford
Brian D. Todoroff
Slavko Galuga
SRI International
333 Ravenswood Ave.
Menlo Park, CA 94025
(650) 859-4165, (650) 859-4375, (650) 859-4316, (650) 859-4755
john_shockley@sri.com, reg_ford@sri.com, brian_todoroff@sri.com, slavko@erg.sri.com

Keywords:
Live Systems, HLA Architecture, Live Entity Surrogate, RTI, RTI Reentrancy

ABSTRACT: *Instrumented live players, often called "live simulations," have generally been included in mixed live/virtual exercises via a gateway or live entity surrogate approach to integration. In this approach, the entire instrumentation system or range system is considered as a single HLA federate even though the system may consist of several participants at the individual operator or vehicle level (e.g., a 36-aircraft training system would still only be one federate). This approach has always seemed to be the most technically feasible, considering many of the characteristics and limitations of live instrumentation.*

This paper proposes a new approach to augment the system gateway architecture. In this approach, individual range participants--e.g., aircraft or ground vehicles--are individual HLA federates and an RTI is internalized into the range system. This proposed architecture allows substantially greater flexibility for the user, while accounting for many of the limitations and characteristics of live instrumentation. In this paper we describe the proposed architecture, address the potential advantages of the architecture, and discuss results to date in implementing this architecture on a live ground training system. These initial experiments to examine the feasibility of this architecture have yielded positive results. They have also resulted in specific recommendations--in particular, a recommendation to make the RTI Library reentrant to enable multiple federates on a single process. While we do not propose replacing the existing gateway architecture, we do see that this architecture has promise and may be applicable for a number of live range systems.

1. Introduction and Background

It has generally been assumed that live instrumented systems will be integrated into a High Level Architecture (HLA) federation through the use of a live entity surrogate. This surrogate acts as a gateway between the live system and the Run-Time Infrastructure (RTI), and the live system is considered as a single HLA federate. This approach, illustrated in Figure 1, has been considered as the most technically realizable option, since many of the characteristics that are unique to live instrumented systems--e.g., transmitting data between participants via limited bandwidth radio frequency (RF) links--could still be optimized for the system itself. Other HLA-compliant federates interface at the system level, so the internal system architecture is unaffected.

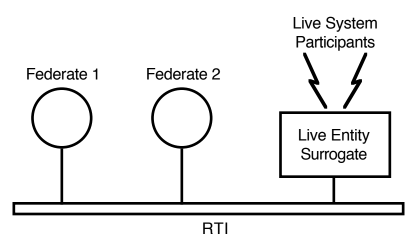

Figure 1 Integration of Live System with HLA via Live Entity Surrogate

This paper discusses the proposed HLA integration of the Deployable Force-on-Force Instrumented Training System (DFIRST), a 60-player live instrumented system for ground force training of National Guard units, developed by SRI International under contract to the Defense Advanced Research Projects Agency (DARPA). DFIRST was first deployed operationally by the Idaho National Guard in October 1995. A second DFIRST is being developed under contract to the National Guard Bureau for the South Carolina National Guard. Expansion of the first DFIRST to a 150-player system is planned under All Service Combat Identification Evaluation Team (ASCIET) sponsorship.

DFIRST interaction with other systems is currently supported by a Distributed Interactive Simulation (DIS) gateway. An effort is now under way to make DFIRST HLA-compliant. Reference 1 documents the first step in the HLA compliance process for DFIRST. In this paper, we presented an initial simulation object model (SOM) for DFIRST and reported some lessons learned on developing a SOM for a live system. For this SOM, we considered DFIRST to be an HLA federate containing several object classes that could each subscribe and publish attributes to the RTI, and concentrated on reflecting the internal operation of the system in the SOM.

Since this initial effort, we have re-examined the requirements for HLA compliance for DFIRST and have considered alternatives to the single federate approach. This paper outlines these alternatives and explores some key issues associated with implementing them in the actual system.

Section 2 presents the proposed architecture alternative to the single federate live entity surrogate approach to HLA compliance for live instrumented systems. Section 3 describes efforts to date to implement the alternative architecture and the lessons learned in doing so. Section 4 reports conclusions and recommendations based on this work.

2. Proposed Architecture

Like many other live instrumented systems, DFIRST has an architecture in which the individual participant vehicles are connected to a base station via an RF datalink. Although some functions, such as direct-fire engagement simulation, are handled between participants without base station mediation, the base station is the core of the system architecture and receives data on all participants and all events. For this reason, the DIS and HLA gateways were connected to the base station local area network (LAN), as shown in Figure 2.

While the architecture depicted in Figure 2 would satisfy the requirements for HLA compliance, it does not take advantage of the full potential of the HLA. We have began to consider alternative architectures that would allow the HLA to become a more integral part of the DFIRST system. Such integration would allow us to exploit the potential advantages of using HLA-compliant tools within DFIRST, and at the same time lend greater flexibility for interacting with other federates and federations.

2.1 Description

The key characteristic of the proposed alternative architecture is that HLA and the RTI are brought inside the DFIRST system rather than placed at the boundary between DFIRST and external systems. One variant of the proposed architecture is depicted logically in Figure 3, and physically in Figure 4. In this architecture, each DFIRST live and simulated entity is an HLA federate. In addition, some of the principal functions of DFIRST--including displays, data loggers, and after-action review (AAR) tools--may also subscribe to data as HLA federates. In other variants (not shown), DFIRST live players and simulated entities, instead of each being an atomic federate, are grouped among several federates (e.g., players on RF net #1, players of RF net #2, minefields, artillery units, etc.).

One of the original justifications for the live entity surrogate architecture was to retain critical characteristics of the live system architecture, such as the RF datalinks. The proposed architecture manages to retain these characteristics while defining the system as a set of individual participant federates. As indicated in Figure 3, each participant federate “logically” includes the datalink and entity management portions of DFIRST that are associated with that particular entity. As depicted in Figure 4, however, datalink control processing is performed centrally, as in the original system.

The RTI will supplement rather than replace interaction of participants via direct RF communication. Participants who belong to different RF networks, and so cannot communicate directly with each other, will interact via the RTI. When a federate receives an update from another federate who resides on a different RF network, the update will be evaluated for relevance, and if needed it will be marked for RF uplink. For example, a target must be informed if it has been hit by a munition so that it can assess damage; a shooter needs to know positions of all potential targets within lethality range of its weapons so that it can perform target pairing and hit determination after it fires.

The datalink manager process will be responsible for scheduling uplinks. If more than one Participant Instrumentation Package (PIP) in an RF network

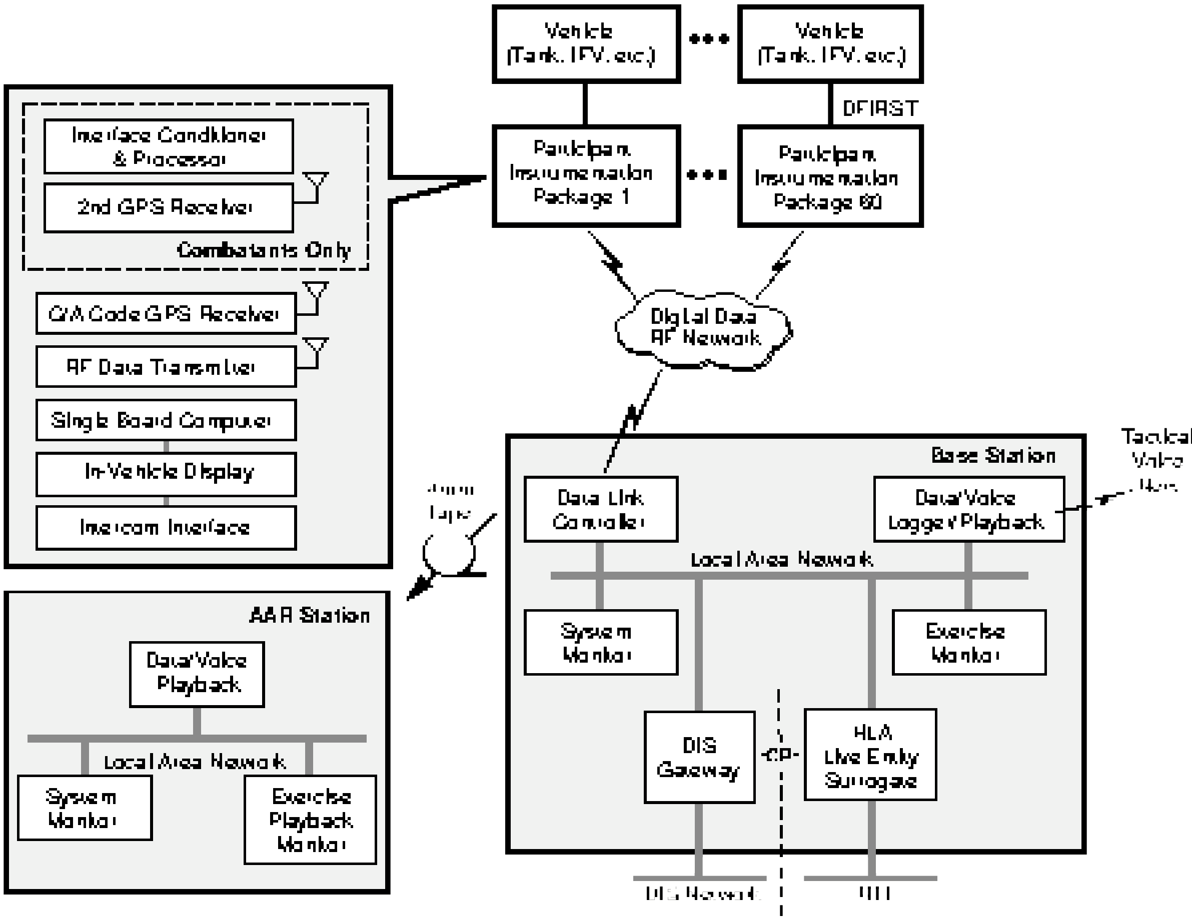

Figure 2 DFIRST System Block Diagram with DIS and HLA Gateways

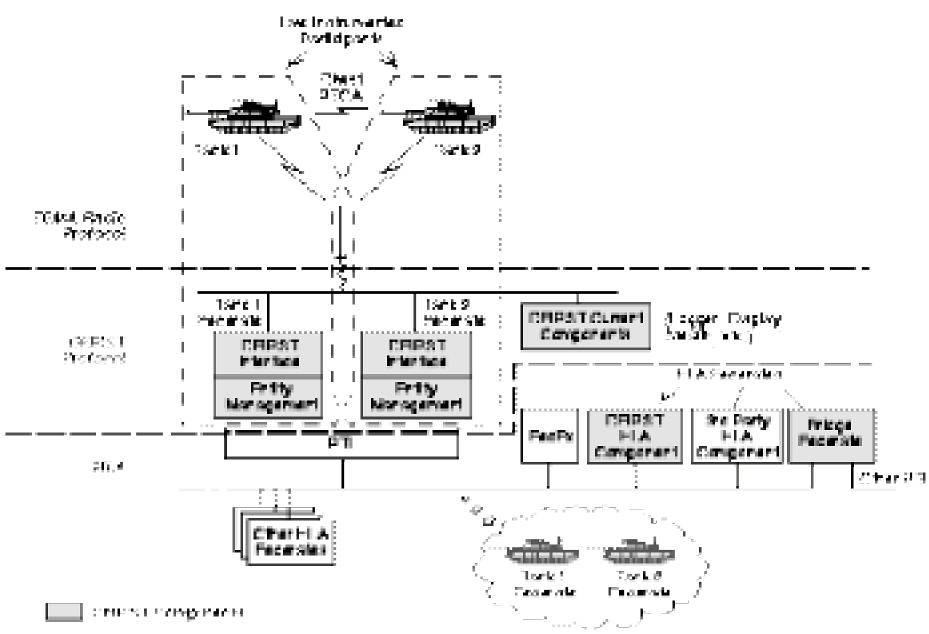

Figure 3 DFIRST System Block Diagram with Alternative HLA Architecture--Logical View

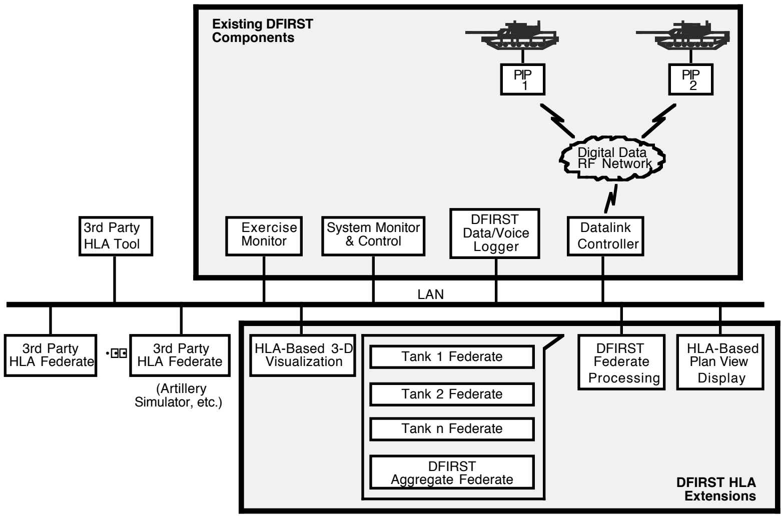

Figure 4 DFIRST System Block Diagram with Alternative HLA Architecture--Physical View

indicates interest in an update, the first request will trigger an uplink message broadcast to all PIPs in the network. An engagement "event" message--e.g., a report that a target was hit by a munition--is guaranteed to be delivered, but because availability of time division multiple access (TDMA) slots for data uplink may be insufficient to send all position updates, requests will be prioritized. Due to bandwidth limitations, a shooter's PIP may have less comprehensive knowledge about positions of potential targets than its federate proxy in the base station. If a shooter's PIP reports "no target" after a weapon fire, the federate proxy may subsequently determine there was an actual target and report the result over the RTI. Section 3.2.3 provides additional discussion of issues related to hybrid RF/RTI communications.

As seen in Figure 3, the proposed architecture assumes that the RTI that is *internal* to the DFIRST system will also be used for interactions with non-DFIRST federates. However, it is possible that when alternative RTIs become available, DFIRST may be part of a federation that uses a different RTI than is used by DFIRST internally. In this case, a second RTI will be run simultaneously with the first and will connect DFIRST with other HLA-compliant systems. This second RTI is depicted in Figure 3 as the "other" RTI.

The proposed architecture may represent a radical departure from the system-as-federate approach indicated by current live entity surrogate-based approaches, but we believe it may be justified by several technical and operational advantages, as explained in greater detail in the following subsection. Although we anticipate advantages for DFIRST in migrating it to the proposed architecture, we recognize that such an architecture may not be suitable for all live instrumented systems. Consequently, we do not advocate blanket abandonment of the live entity surrogate approach to HLA integration for live systems.

2.2 Anticipated Advantages

The key advantages of the proposed architecture are in the areas of expandability of the DFIRST system, flexibility of deployment configurations, and ability to use other HLA-compliant tools to augment those developed for DFIRST.

Expandibility. The current DFIRST system is restricted to 78 players at two messages per second or 156 players at one message per second because of the limitations of the TDMA datalink network. Extensive datalink changes would be needed to handle more players. Additional networks (e.g., using a different frequency or hopping pattern) could multiply the number of players, but would require a mechanism for supporting interactions between players on different networks. A specific DFIRST solution could be implemented, but HLA provides a ready-made solution. The proposed architecture as described in Section 2.1 will enable DFIRST to be expandable to as many participants as can be supported by simultaneous RF networks and network bandwidth.

The proposed architecture also provides more options for expansion design. For example, jumping from the original 60-player National Guard DFIRST to the 150-player ASCIET version requires reducing updates from two per second to one per second if the system is restricted to a single RF datalink. However, an HLA-based design would make it feasible to retain the higher update rate for direct communications by distributing the participants among two 76-player RF networks.

Design flexibility is also enhanced for RF bandwidth allocation. For example, if the system is not limited to a single RF network, it would be possible to reduce the number of players capable of being supported by a single network and use the saved TDMA slots for other purposes--e.g., more bandwidth allocated to base station-to-PIP communications.

Flexibility. A second advantage of the proposed architecture is the flexibility it provides for a variety of DFIRST configurations that might be fielded. For example, although DFIRST is a training system, it has been suggested that it might be used to supplement other systems to support tests. For example, suppose that a test required engagements among a few DFIRST-instrumented participants but had no need for any other DFIRST functionality. Instead of transporting a DFIRST system and an HLA gateway to the test range, we would need to send only the requisite number of PIPs, supported by one or more HLA federate processes (which could run on any available NT host), a master radio, a GPS receiver, and a computer running the datalink control and differential corrections processes under QNX. If the host test range were configured as an open HLA system, DFIRST could be regarded as a plug-in component, to the degree supported by a common federation object model (FOM).

Tool Augmentation. The third advantage of this architecture is that it allows HLA-compliant tools to be incorporated as components within the DFIRST system to augment or replace current DFIRST functions. For example, DFIRST data logging and display capabilities were developed to meet the original system requirements. As new requirements are added, including interaction with the virtual world and larger numbers of participants, it is possible that HLA-compliant display and AAR tools may be available that are well-suited to the new requirements and can be adopted without incurring a significant DFIRST development effort.

3. Implementation Efforts to Date

SRI is currently implementing the proposed architecture in DFIRST. We expect that these efforts will demonstrate the feasibility of the architecture, and will complete the process of making DFIRST HLA-compliant. The following sections address the initial demonstration and the lessons learned thus far.

3.1 Initial Demonstration

Our first steps in implementing this architecture involved dividing DFIRST participants into classes of federates. We revised the SOM from the initial version so that it would represent the characteristics of the new architecture.[1] Table 1 contains the revised Class Structure Table for the SOM. In developing this SOM, we concentrated on presenting the DFIRST participants and battlefield in a manner that would be useful and logical from the point of view of an outside federate looking in. Since the principal focus of this paper is the new architecture for incorporating live systems, it does not include the detail of a complete SOM. Table 1 should be sufficient to provide the reader with an overview of the system characteristics that are being implemented.

Figure 5 depicts the logical DFIRST architecture implemented in this original demonstration. As evident in this diagram, the implementation is a simplified version of the overall DFIRST architecture and is limited to two federates interacting over the RTI. Both federates are shooters.

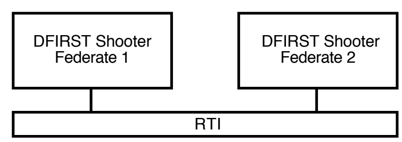

Figure 5 DFIRST Initial HLA Demonstration Architecture--Logical View

Figure 6 depicts the physical view of this demonstration architecture. As shown in this diagram, the DFIRST PIPs are connected via RF datalink to a base station. Three Pentium II processors running the QNX real-time operating system perform the functions of the mini-base station, including datalink controller, system control, map-based display, data/voice recording, and system control. A Pentium II processor running Windows NT™ is the host of the two DFIRST federates and RTI version 1.0. All processors are connected via the DFIRST LAN.

[1] In the initial effort, we developed a SOM for DFIRST that faithfully represented the characteristics and capabilities of the operational system as it currently exists. For this reason, we have revised the SOM to reflect capabilities that will be added to DFIRST to enable anticipated federation interactions.

Table 1. Revised DFIRST Object Class Structure Table

Class 1	Class 2	Class 3	Class 4	Class 5
DFIRST Entity (PS)	PositionedEntity (PS)	LandPlatform (PS)	Shooter (PS)	Artillery_Unit (PS)
				M1A1_Abrams (PS)
				M3A0_BFV (PS)
				M2A0_BFV (PS)
			Nonshooter (PS)	M113A3 (PS)
				OtherVehicle (PS)
				M978_HEMTT (PS)
				M977_HEMTT (PS)
				M981_FISTV (PS)
				M998_HMMWV (PS)
				M1025_HMMWV (PS)
				M1026_HMMWV (PS)
		Munition (PS)	Ballistic (PS)	mm25APDS (PS)
				mm25HEI (PS)
				Coax (PS)
				mm120HEAT (PS)
				M449_A1 (PS)
				M483_A1 (PS)
				M107_CB (PS)
				M107_TNT (PS)
				mm120APFSDS (PS)
			Guided (PS)	TOW (PS)
	AreaEntity (PS)	EnvOffLimitsArea (PS)		
		MineField (PS)		
	TrainingOfficer (PS)			

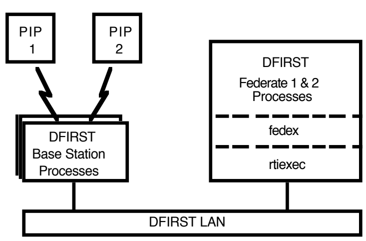

Figure 6 DFIRST Initial HLA Demonstration Architecture--Physical View

In the demonstration, the two DFIRST federates are PIPs configured as M1A1 tank instrumentation. One of the PIPs is set up for normal RF communications, but the other has no antenna and is connected to the base station via cable. As a result, the PIPs can communicate with the base station but not with each other. The PIPs exchange Global Positioning System (GPS)-derived time-space position information (TSPI) and direct-fire engagement interactions via the RTI rather than via the normal PIP-to-PIP direct RF link.

For the Fall SIW presentation, we expect to have successfully demonstrated two-way engagements between live instrumented participants configured as individual HLA federates. The participants will be PIPs in automobiles operating in the SRI parking lot.[2]

3.2 Findings

Designing and implementing the demonstration have resulted in several findings that may be of value to the community. The key findings (each discussed in further detail in this section) are:

- Alternative architectures are achievable and appear to be consistent with the RTI specification, but support provided by the current RTI implementation imposes many limitations, as noted below.
- Because the Win32 C++ RTI library as currently implemented is not reentrant, the goal of using

[2] Actual implementation and demonstration are under way as this paper is being written. If the demonstrations are complete in time for the hard-copy submittal of this paper, they will be included. If not, they should be complete in time for the paper presentation in September.

HLA as a method for improving system scalability can be achieved only by awkward workarounds.
- More effort is needed to ensure that RTI-based engagements between DFIRST participants match those supported by direct RF communications as closely as possible. Ensuring a fair fight between DFIRST participants and members of non-DFIRST federations raises additional issues.

3.2.1 Feasibility of Alternative Architectures

The first finding tentatively validates our approach to using an alternative HLA architecture to link live systems. Based on our efforts to date, we believe that such alternatives will be feasible and will provide the advantages for DFIRST that we envision. Depending on the characteristics of the system, other live systems may benefit from similar advantages. However, although we see how alternative architectures can be implemented, we also discovered during this effort a potentially severe limitation in the current implementation of the RTI. This limitation is discussed, along with its implications for live systems, in Section 3.2.2.

3.2.2 RTI Library Reentrancy

Although our initial demonstration was limited to two live participants, and thus would not stress system computing resources, we kept in mind that conserving resources was essential to our ultimate goal of boosting DFIRST scalability beyond 150 vehicles. If each federate ran as a separate process, the burden of context switching and the 2 MB RAM used by each instantiation of the RTI library would quickly overtax our four pentium processors (only one of which is currently allocated to PIP communications control). Our intended design was to have each DFIRST federate run as a thread in a single executable process. However, we discovered that hosting more than one federate in a process is problematical because the RTI library as currently implemented is reentrant for only five of its routines (and none of these five is among the frequently called workhorses of a federation execution).

If more than one federate is hosted in a process, there may be multiple simultaneous calls to the RTI library. If the RTI library were reentrant, it could handle these calls as they came in. However, the lack of reentrancy in the current C++ RTI prevents multiple threads from sharing the RTI library. Each time one federate makes a call to the library, any other federate attempting to make a call to the RTI library will receive an error until the current request is completed. A wrapper class could be implemented to provide thread-capable calls to the RTI. However, because these would be blocking calls, each federate would need to wait for its predecessor to finish completely before it could begin. Such a limitation would induce unacceptable latencies in a real-time system.

Because we have found nothing in the HLA specifications to prohibit reentrancy, we have inferred that this existing limitation is probably an implementation decision rather than a planned characteristic. Our information is derived from documentation for RTI version 1.0 for NT, but the documentation we have seen for newer versions of the RTI indicates that this limitation remains.

Because most HLA implementations to date are virtual-world simulators that are quite naturally modeled by a one-federate/one-process architecture, reentrancy of the RTI has generally not been an issue. However, we understand that some programs have discovered this limitation previously and have had to develop workarounds that they consider to be clumsy.

Although we are not the first to run across this problem, we can see why it has not been a burning issue among the early HLA adopters, most of whom work in the realm of virtual simulators. A virtual simulator is in some respects equivalent to a PIP in a live system such as DFIRST. Where the PIP processes data from its instrumentation, the simulator provides data on what the simulated entity is doing, and in addition the simulator must also run other simulations (e.g., environmental effects) that in a live system are "modeled" very well by the real world in which the live participant is operating. When HLA-related processing requirements are added, it may be natural for the virtual simulator processor to implement these using a single-process/single-federate approach, as it would be for a PIP if the PIP were directly connected to the RTI. However, in a live system such as DFIRST, the PIP federate is connected to the RTI at the base station rather than in the PIP. Because the base station needs only to manage system entities rather than simulate them, the amount of processing power required on the base station for each HLA DFIRST entity is small. With such small processing requirements, it makes sense to share one computer among many entities. In this situation, the memory and context switching burdens associated with having a separate process for each entity become significant. The ideal solution would be to simulate each entity as a separate federate implemented not in a separate process, but in a thread within a master process.

Alternatively, one could also simulate all entities using only one federate. This, however, limits the usefulness of many of the data distribution features of the RTI by forcing subscriptions to be shared by all entities being simulated. This means that each entity's HLA interface must deal with all entities of interest to any other entity, which drastically increases the search space for interactions between participants and multiplies the required processing power and time. This once again introduces unacceptable latency in a real-time system.

Another option for us that is consistent with the current RTI is to group together some number of live participants as one federate hosted in a single process. For purposes of data management, the group of vehicles will be considered as an aggregate--all will publish and subscribe as a unit--so there will be no conflicting simultaneous calls to the RTI library. While this option is not as appealing to us has having multiple federates in a process, it does appear to enable us to retain some of the advantages we envisioned for DFIRST.

Overall, we believe that the RTI library needs to be reentrant in future versions of the RTI.

3.2.3 Ensuring a Fair Fight in a Hybrid RF/RTI Communications Environment

As we began to determine how to implement the RTI-based engagements, we realized some key "fair-fight" issues needed to be addressed. The first is derived from the design of DFIRST engagement simulation for direct-fire engagements, in which a statistical real-time casualty assessment (RTCA) is based on the published vulnerability table for the intended target vehicle. Target pairing is accomplished by identifying target candidates within the firing cone of the shooter and, if there is more than one candidate, selecting one based on a number of factors, such as which target poses more of a threat to the shooter. Vulnerability tables associated with the identified target are used by the shooter's PIP to determine whether the target was hit. If hit, the target consults other tables to determine the level of damage.

For RTI-based interactions between DFIRST participants, the shooter-target pairing is accomplished in the same manner, provided the shooter has received position data for all potential targets via RF uplink. However, in cases where RF bandwidth limitations preclude providing all the needed position data regarding potential targets to the PIPs, target pairing and hit determination will need to be performed in the base station, where all information provided via the RTI is available.

The other key fair-fight issue concerns latencies in communications that support engagement interactions. For example, if a weapon fire/hit event is reported to a target after the real weapon would have arrived, the target is given an unrealistically long interval to shoot back. Because current DFIRST direct-fire engagement interaction is communicated directly between the PIPs, lost communications are relatively infrequent, and it may be possible to repeat a lost communication and still remain within the required latency. However, implementing these same engagement interactions via the RTI will necessarily incur some latency that is not present in the current method. As long as communications are reliable, we believe that, despite some additional latency, we will still be able to implement engagement simulations via the RTI that will result in a fair fight. However, because communications are inherently imperfect in an RF environment, the RTI will not support a fair fight as reliably as will direct radio communications.

Because of the latency concerns, an operational implementation would try to ensure that those participants most likely to engage each other (e.g., because of location or strategy) would be on the same DFIRST datalink channel. Although in most cases the RTI would support interaction with a level of fidelity comparable to that provided by direct communications, and communications latency would not exceed weapon delivery time, it is unavoidable that the lossiness of RF communications would sometimes result in lesser fidelity and excessive latency.

The latency and bandwidth issues discussed above raise another issue with respect to the use of an RTI internally in a live system. We have stated that one of the reasons for such an architecture is to use the RTI data distribution and filtering capabilities to help manage the RF datalink resources--e.g., limit the data to individual federates based on their subscription requirements. Since we will be using the RTI in this manner, we have realized that the subscription characteristics of a live system are different from those of a simulation.

More specifically, because the bandwidth of a live system is more limited than that of a LAN and because the communications medium (i.e., RF datalink) is subject to dropouts, filtering and prioritizing of updates needs to consider impacts on overall interactions when some updates may be lost. For example, a federate subscribing to attributes of vehicles of interest would assign the highest priority to updates generated by simulated vehicles (assuming that these updates might be used to inject a simulated view of the vehicle using some form of visual stimulation) in order for them to be depicted. The second highest priority might be for updating potential target vehicles within the shooter's lethality range, so that the RTI-based engagements are better facilitated. The priorities will need to be established in consideration of the needs to present a consistent view of the battlefield to the participants and some of the characteristics of the engagement simulations (i.e., taking into account added latency for RTI-based engagements). Finally, the individual federate cannot establish priorities because it cannot manage the datalink. There needs to be some datalink management capability that has knowledge of the "big picture" to determine highest priority updates for a given federate.

3.2.4 Ensuring a Fair Fight: Live System RTCA

As described above, DFIRST uses a statistical RTCA algorithm. Such an algorithm is common for live systems, since the accuracy of tracking instrumentation is usually insufficient to guarantee a realistic RTCA based on a ballistic flyout of a single round.[3] While such an algorithm could be employed, it would not be satisfactory because too many results would be dominated by tracking errors,[4] rather than by the effectiveness of the crew.

Although DFIRST shooter position and gun-pointing angle could be used as the basis for initiating a ballistic flyout of a single round for reporting to federates, mixing statistical RTCA methods in DFIRST with ballistic simulation in a federate would result in an unfair fight. The two different methods would result in different outcomes, given the same engagement parameters, even if the ballistic flyout were based on error-free position and pointing-angle data.

For these reasons, we have chosen to use the same RTCA algorithm for both direct DFIRST and RTI-based engagements. The DFIRST FOM specifies information exchange to support statistical RTCA, including accuracy parameters of position and gun-pointing angle data, and reference to the standard AMSAA tables DFIRST uses for hit determination and damage assessment.

We believe that these issues are typical of a broad class of live-virtual interactions and mixed types of engagements. Similar reasoning, therefore, needs to be employed in other live interactions implemented via HLA to ensure a fair fight.

4. Conclusions and Recommendations

Based on the discussion presented in this paper and the results of the demonstration, we have the following conclusions and recommendations:

- Architectures for HLA integration of live systems other than the Live Entity Surrogate appear to be feasible and, depending on the system involved, may yield several technical and operational advantages.
- Despite the apparent feasibility and advantages for DFIRST of an alternative architecture, we recognize that such an architecture may not be suitable for all live instrumented systems. Therefore, we do not advocate blanket abandonment of the live entity surrogate approach to HLA integration for live systems.
- Future versions of the RTI should have a reentrant library to enable multiple federates to be hosted by a single executable process. This capability will greatly aid in HLA integration of live systems, without limiting the executable-per-federate architecture.
- Issues that were raised during design and implementation of our demonstration served as a reminder that successful integration of live systems with virtual and constructive simulations requires careful consideration of the characteristics of the live system.

---

[3] One notable exception is no-drop bomb scoring (NDBS) used on air training ranges. In this case, a single impact point is determined based on the aircraft's position and velocity at bomb release. In this type of simulation, however, the statistical tracking errors can be matched to the statistical ballistic dispersion of the bomb. Such a relationship does not hold for tank main gun rounds.

[4] DFIRST accuracy is 1-2 m in vehicle position, and 2-3 milliradians in gun-pointing angle.

5. References

[1] J.W. Shockley, R.E. Ford, and G.V. Lucha, *HLA Compliance for a Live Instrumented Training System: A Case Study*, 986-SIW-163, Spring Simulation Interoperability Workshop, Orlando, FL, March 9-13, 1998.

[2] DMSO, High Level Architecture *Interface Specification, Version 1.1* , 12 February 1997.

[3] DMSO, High Level Architecture *Run-Time Infrastructure Programmer's Guide*, *Version 1.0, Release 3*, 14 November 1997.

Author Biographies

JOHN W. SHOCKLEY is a Senior Research Engineer at SRI International. Over the past 15 years, he has worked at SRI on a broad variety of projects--primarily in the areas of test and training range instrumentation systems for the Army, Navy, and Air Force. He began working on modeling and simulation aspects of these systems and has since participated in DIS/HLA standards development activities for the past nine years, concentrating on integrating live and virtual systems.

REGINALD E. FORD is a Senior Systems Analyst at SRI International. Over the past 18 years, he has worked on many Navy, National Guard, and Marine Corps test and training range systems. This work includes software specification and development, technical studies, and system test, evaluation, and improvement. He is the lead software engineer for DFIRST.

BRIAN D. TODOROFF is a Research Engineer at SRI International. He has experience in use of Microsoft Windows 95™ and Windows NT™ for real-time and quasi-real-time applications for control and simulation. He developed the DIS interface for DFIRST and is currently working on the effort to make DFIRST HLA-compliant.

SLAVKO GALUGA is a Research Engineer at SRI International. For the past eight years, he has been involved in developing and enhancing military simulation software, with a primary focus in mission planning and analysis for the Air Force. His work has included software optimization, mathematical modeling, and multi-platform system integration and test.