Top 10 DIS V8 Improvements

Lance Call
CAE USA/AFRL
Dayton, OH
lance.call.ctr@us.af.mil

Robert E Murray
SimPhonics, Inc.
Tampa, FL
bobmurray222@gmail.com

ABSTRACT

Distributed Interactive Simulation (DIS) is getting a major overhaul for its next version. The redesign greatly simplifies the protocol formats, reduces the number of Protocol Data Units (PDUs) and makes it easier to add capability, enabling DIS for a more capable and longer future.

The Simulation Interoperability Standards Organization (SISO) DIS Product Development Group (PDG) has nearly completed the DIS Version 8 (V8) draft to be considered for adoption by the Institute of Electrical and Electronics Engineers (IEEE). This updated version will break the backwards compatibility required by previous generations in order to bring several new capabilities and address existing issues.

This paper will cover the top ten improvements that have been made in the DIS V8 standard compared to DIS Version 7 (V7) including:

1. Extensibility through a new PDU Design
2. SISO Reference document number 30 (SISO-REF-030) yearly updates to capability/reusable records
3. Improved Radio Frequency modeling of AESA radar and advanced jammers
4. Combined Parabolic Circular Dead Reckoning algorithm to reduce PDU rates
5. Multiple Entity-Entity State PDU to allow higher entity counts with dramatically fewer PDUs
6. Partial Updates/Split Modeling of systems and reliability in all SIMAN PDUs
7. Extensible Markup Language (XML) schema supports automated marshalling and Wireshark dissectors
8. New PDUs: Application Control, Interactive Identification Friend or Foe (IFF), Laser, and Weather
9. Simplified Gridded Data PDU with weather support
10. Miscellaneous small updates like a 64-bit timestamp and Little-Endian data representation

While DIS V8 will not be on the wire compatible with DIS V7 it will be possible to use gateways to translate all of DIS V7 into DIS V8. DIS V8 is the biggest change to be made to the DIS standard since its beginning, and we will cover the top 10 improvements that have been made in the standard.

ABOUT THE AUTHORS

Lance Call is a Scientist, Systems and Software Engineer with CAE USA at the Air Force Research Laboratory (AFRL). He graduated Magna Cum Laude with a Bachelor of Science degree in Electronics Engineering Technology from Brigham Young University in 1988. He has worked on real-time threat systems, and integration of live, virtual man in the loop, and computer only simulations. He has been responsible for Cross Domain Security systems and rule set development, improving threat systems and integrating simulators with live aircraft systems. He was the SISO Compressed DIS (C-DIS) drafting group editor. He is an IEEE member.

Robert Murray is a computer engineer with over 30 years of experience in high fidelity flight simulation and real-time network implementations. He is a retired Boeing Technical Fellow and is currently working under contract with SimPhonics to help develop the next generation of the DIS standard. Bob has been active in the development of the DIS standard since 1993. He is currently Vice Chair of the DIS Product Development Group within SISO and the Draft Editor. Bob received a BS from the University of Cincinnati in 1983 and MS from Washington University in St. Louis in 1993, both in Electrical Engineering.

Top 10 DIS V8 Improvements

Lance Call
CAE USA/AFRL
Dayton, OH
lance.call.ctr@us.af.mil

Robert E Murray
SimPhonics, Inc.
Tampa, FL
bobmurray222@gmail.com

HISTORY OF DIS

Historically, distributed simulation started with a Defense Advanced Research Projects Agency (DARPA) program called SIMulator NETworking (SIMNET) in the 1980s. This program proved that real-time network interoperability between training simulations was practical and that it greatly enhanced training effectiveness.

Distributed Interactive Simulation (DIS) standardization evolved from SIMNET starting in 1989. The first generation of Protocol Data Units (PDUs) was defined in IEEE 1278.1™-1993 (DIS Version 2). It was a stepping stone between SIMNET and the broader capability that DIS was to become. The standard was greatly expanded in 1995 to create DIS Version 5 (V5), the first version of DIS that was widely implemented.

The 1998 standard DIS Version 6 (V6) added new PDUs but did not change any of the existing DIS V5 PDUs. DIS Version 7 (V7) [see IEEE (2012) and IEEE (2015)], officially known as the IEEE 1278 family of standards, made a significant improvement by adding capability, correcting errors, clarifying rules, and adding five new PDUs. It did this while maintaining almost all backward and forward compatibility with DIS V5/V6. DIS V7 maintained compatibility with previous versions largely by adding new information to previously unused padding fields without modifying the overall format of the PDU.

DIS Version 8 (V8), summarized in Murray (2018), will make extensive changes to the PDU design and structure that will break the backward compatibility with all previous DIS Versions. This is being done in order to address issues identified over 30+ years of use, and to make it more flexible going forward.

IMPLICATIONS OF BREAKING COMPATIBILITY

The DIS Product Support Group choose to break compatibility with DIS V7 in order to provide all of the benefits that will be highlighted in this paper. These would not be possible without breaking compatibility. This means that simulations that are currently using DIS will require significant software changes in order to send and receive DIS V8 messages. DIS V7 equivalent capabilities can be translated using a gateway. Applications can take advantage of the XML schema and PDU definitions for DIS V8 to auto generate marshalling and unmarshalling code to ease some of the changes. There will be additional information to track internally within the simulation such as AESA radar states, infrared signatures, laser weapons, weather and other information that are not part of the DIS V7 standard. By making these changes new opportunities for advancement will be available to DIS simulations.

IMPROVEMENT 1: EXTENSIBILITY THROUGH A NEW PDU DESIGN

In DIS V7, extensibility was not supported in the PDUs directly. Users often added capabilities to DIS by adding custom datums in the Simulation Management (SIMAN) Data PDU. These additions were typically user specific and undocumented because there was effectively no way to update the IEEE 1278.1 document in a timely fashion. This means that they were not available to the general DIS user community to reuse or to add capabilities.

DIS V8 will adopt a standard new PDU design that specifically supports the addition of commonly formatted extension records to every PDU. This makes every PDU extensible, in a common way that can be documented to encourage reuse.

Every PDU will have a common structure, for example, PDUs will have a common PDU header, followed by a fixed length main PDU body that will contain data that is common to all instances of that PDU which allows for quick filtering. The body will be followed by a list of standard formatted extension records that will contain any remaining information for the PDU (see Figure 1).
The Entity State PDU body for example will contain the location and orientation of the entity, but it will not contain the velocity or acceleration because static (non-moving) entities do not require velocity or acceleration information. The velocity and acceleration will be provided in a dead reckoning extension record only for moving entities that actually require this information. This approach provides smaller PDUs for static entities leading to less network bandwidth. It provides a standard way to add information about the entity such as Articulated and Attached Parts, or possibly in the future information to help model the infrared (IR) signature of the entity. The Appearance and Entity Marking have been moved to an extension record because these fields do not change often, and therefore typically only need to be sent once during each heartbeat interval allowing other updates to omit this information. This makes smaller PDUs and less bandwidth for many updates.

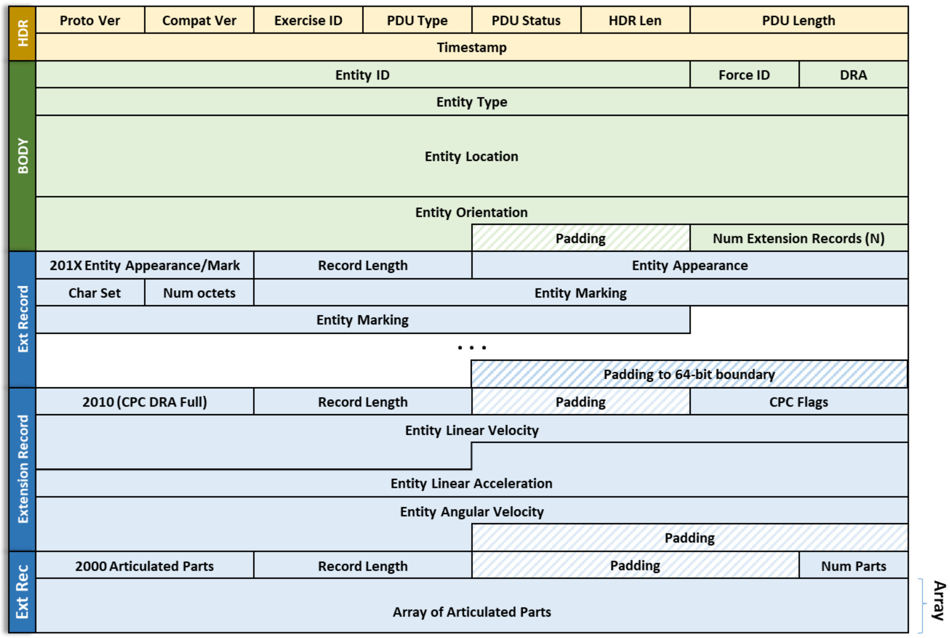

Figure 1 New DIS V8 format with header, body, and extension records

The PDU header contains fields that should allow more future forward compatibility including the compatibility version to indicate the oldest compatible PDU version and an explicit header length so that additional bytes could be added to the PDU header in the future.

Any number of extension records may be added to a PDU up to the maximum DIS V8 PDU size of 8192 (8K) octets. It is recommended however that PDUs should be kept below the DIS standard Maximum Transmission Unit (MTU) size of 1400 octets in order to keep PDUs from fragmenting on Ethernet networks even if they are encrypted for security.

IMPROVEMENT 2: SISO-REF-030

Updating the official IEEE 1278.1 standard requires broad consensus and is a time-consuming process. The most recent IEEE 1278.1 updates have occurred in 1998 (V6), 2012 (V7), and expected in 2025/6 (V8). This ten-year plus update cycle is not very responsive to the changing needs of DIS users. Because of this, the Simulation Interoperability Standards Organization (SISO) DIS V8 Product Development Group (PDG) decided to create a new standard reference document SISO-REF-030. The IEEE 1278.1 document will standardize all PDUs and the extension records that are deemed to be essential to the usage of the PDU. For example, a dead reckoning extension record is essential to the definition of moving entities and therefore is documented in IEEE 1278.1. Similarly, the Electromagnetic Emission (EE) PDU uses the EE Beam extension record that is essential, and therefore is documented in IEEE 1278.1.

There are many new optional extension records such as the Beam Activity, Azimuth/Elevation List, Jammer False Targets, Angle Deception Parameters, etc., that may be used with the Electromagnetic Emission PDU to model advanced radars as well as advanced reactive jammers. These records, however, are not essential to using the Emission PDU. All optional extension records for all PDUs will be documented in the new SISO-REF-030.

SISO-REF-030 will be managed and updated following the same guidelines and processes that are currently used to update the SISO enumerations reference document SISO-REF-010 [see SISO (2023)]. It is expected that users will be able to propose new extension records in Product Change Requests (PCRs) for consideration and inclusion in SISO-REF-030. The SISO-REF-030 Product Support Group will review the PCRs and recommend changes if needed, and will vote to adopted or reject the proposed changes. Recommended changes will be incorporated into SISO-REF-030. SISO-REF-030 will document the formats and usage rules for those extension records. DIS users can reuse extension records documented in SISO-REF-030 rather than creating custom datums. SISO-REF-030 is expected to be updated on an annual basis. This is expected to lead to more innovation, capabilities, and interoperability in a timely fashion without requiring changes to the IEEE 1278.1 document.

In addition, compressed versions of the new extension records will be documented in a SISO-REF-030 appendix or in SISO-REF-030.2 for use in the Compressed DIS standard [see I/ITSEC (2021) and SISO (2024) for DIS V7 C-DIS] that will be updated for DIS V8, so that extension records can be used in the normal or compressed version of the DIS V8 standard.

IMPROVEMENT 3: RF MODELING

The original emission PDUs were created to model the mechanically scanned radars that existed in the 1990's. Today, however, most new radars are Active Electronically Scanned Array (AESA) radars. Both Combat Air Force Distributed Mission Operations (CAF DMO) and Battlefield Simulation Incorporated (BSI) created custom but conflicting datums to represent AESA radars in DIS V7. Using these messages as a starting point the Advanced Radar and Jammer tiger team created a common set of extension records that allow for much more extensive modeling of AESA radars as well as jammers that are standardized and not conflicting. The tiger team also added new capabilities [see Call (2022) and Call (2023) for details]. A list of the Emission PDU extension records and their intended use cases is shown in Figure 2.

Color Code Key	
	New in DIS V8
ES	Essential
RC	Recommend
OP	Optional

Record Name	Record Type	Radar Use Cases: Basic Radar	Advanced Mechanical Radars	ESA / LPI Radars	High Fidelity - No Local Database	Accurate Detailed Radar Scanning	Detailed Specific Pulse Usage	Jammer Use Cases: Obscuration Jammer	Basic Static State Jammers	Reactive Jammers	Training specific Jamming Display Results	Advanced Jamming
EE PDU - Header	-	ES	ES	ES	ES	ES	ES	ES	ES	ES	ES	ES
EE PDU - Body	-	ES	ES	ES	ES	ES	ES	ES	ES	ES	ES	ES
EE Beam	3503	ES	ES	ES	ES	ES	ES	ES	ES	ES	ES	ES
Jammer Parameters	3504							ES	ES	ES	ES	ES
Universal Scan Data	3505	OP	OP	ES	ES	ES	RC					
Pulse Table Definition	3506		OP	OP	ES	OP	ES					OP
Beam Shape Definition	3507		OP	OP	ES	RC	OP					OP
Fundamental Parameter List	3508		OP	ES	ES	ES	RC					
AZ/EL List	3509		OP	OP		OP	OP					
Beam Activity	3510		OP	ES	ES	ES	RC					
Jammer Activation	3511									ES		
Jammer False Targets	3512								RC	RC	ES	ES
Jammer Triggered	3513									ES		
Blanking Sectors	3514	OP	OP									
Angle Deception	3515										ES	
Range Deception	3516										ES	
Velocity Deception	3517										ES	
False Targets Parameters	3518										ES	

Figure 2 EE PDU associated Extension Records and Use case summary table

IMPROVEMENT 4: COMBINED PARABOLIC CIRCULAR (CPC) DEAD RECKONING

Dead reckoning has been an enabling technology for distributed simulation since the beginning. The relatively simple algorithms in SIMNET were significantly expanded for highly dynamic vehicles in DIS V5. However, the expansion also led to a more complicated set of eight dead reckoning algorithms but little guidance on choosing the best one. There are two categories of these algorithms, the so-called World and Body formulas, named after the coordinate systems where the extrapolation is performed.

The World formula predicts a straight line or parabolic path. The Body formula predicts a circular or spiral path. The World formula is relatively straightforward, but the Body formula is not well understood and is rarely used in practice. This is unfortunate because the Body formula better predicts the circular path of turning vehicles and munitions. Better prediction means that fewer updates are required to stay within the error threshold. Fewer updates mean fewer PDUs, saving network bandwidth. Even those who did understand the Body formula had difficulty in using it because not all simulations in the exercise had implemented it. This is a case where too many choices hinder interoperability.

DIS V8 establishes a single algorithm for selecting the best formula to use for dead reckoning, first proposed in Murray (2019). The algorithm uses the entity type and current motion values to select the best mathematical formula to match the likely path of the entity. The formulas themselves have not changed but are renamed the Parabolic and Circular formulas to better describe their prediction results. The single algorithm is thus named the Combined Parabolic Circular (CPC) dead reckoning algorithm. This is the only algorithm allowed for moving entities in DIS V8. All simulations are expected to implement it.

To help implementers with the new algorithm, a reference C software implementation has been provided and is available on SourceForge at https://sourceforge.net/projects/dsu. In addition, the algorithm and all of the required math is documented in the DIS V8 standard. A Java version should also be available.

CPC Performance

The authors tested the new CPC algorithm using the Air Force Research Laboratory (AFRL) Network Integrated Constructive Environment (NICE) Computer Generated Forces (CGF) system. We created twelve different use cases including both ground vehicles and aircraft to compare how the CPC algorithm performs relative to the DIS V7 Dead Reckoning Algorithm 4 (DRA 4) which uses Linear Velocity, Linear Acceleration and Rotational Velocity. The PDU count achieved using DRA 4 was used to normalize the data (i.e., DRA 4 count = 100%) in all cases so that it is easy to see how many fewer PDUs were sent using CPC as a percentage rather than graphing actual PDU counts. The use cases include ground vehicles following a highway over a mountain pass or on roads around a town, a helicopter flying a racetrack, aircraft flying racetrack caps with different straight leg lengths, actual aircraft strike routes, and aircraft random 3D flight maneuvers to simulate a dog fight. These were meant to cover a large variety of use cases. Two extreme use cases were also created. The first is an aircraft that flew in a constant circle, and the other an aircraft that flew in a straight line at a constant speed. Lastly the overall average of the CPC updates was compared to the DRA 4 updates. The results of these test are shown in Figure 3. On average the CPC algorithm reduced PDU updates by 18% (82 % of DRA 4). The more time an entity spends turning, the more the PDU count will be reduced. If an entity is constantly turning the PDU count will only be 40% of the updates compared to DRA 4. A 60% reduction in traffic is a significant difference. The test results show that the amount of PDU reduction will be directly related to the amount of time that entities are turning. Ships and munitions would see similar results. Almost all entities, other than lifeforms, turn in circular paths.

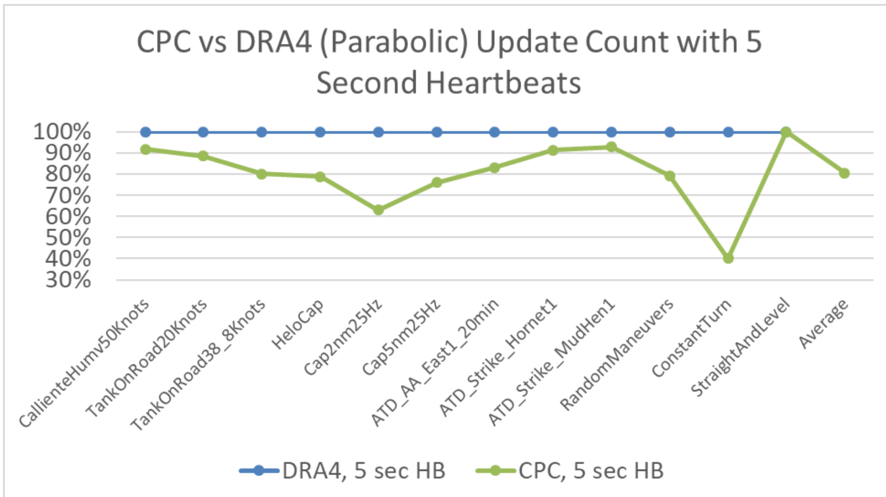

Figure 3 Normalized PDU count DIS V8 vs DIS V7 DRA 4

In DIS exercises the PDU rates increase significantly as entities begin to interact and maneuver. The CPC algorithm will reduce the amount that the PDU rate increases in these scenarios, leading to the ability to push entity counts higher and closer to the limits of the network bandwidth without risking network overload.

IMPROVEMENT 5: ENTITY STATE PDUS WITH MULTIPLE ENTITIES

Number of Entities in a single Entity State PDU			
Max PDU Size in bytes	Static Entities	Moving Entities	Accelerating Entities
1452	41	27	18
8144	251	167	111

Figure 4 Entities per PDU summary table

DIS V8 adds new Multiple Entity extension records that allow for the definition of multiple entities in a single Entity State PDU. There are specific records for static, moving, and accelerating entities. This will reduce the number of PDUs needed by the number of entities per PDU (see Figure 4). This is a drastic reduction in PDU count. To illustrate this point to represent 100K static entities in DIS V7 with a 60 second heartbeat requires 1666 PDUs per second (see Figure 5), but to send those same entities in DIS V8 using the Entity State PDU with the Static Multiple State Entity extension record and ~1452 bytes per PDU it takes only 40 PDUs per second and the number of bits per second is reduced to 25% of the DIS V7 rate. This smaller reduction in bits per second is because while the number of the PDUs decreases dramatically, the size of each PDU increases so the reduction in bits per second is less than the ratio of the number of entities per PDU. Still, a 75% reduction in bandwidth is significant. The Multiple Entity extension records do have some limitations. The main limitation is that all entities in an extension record must be of the same type (F-16, M1A1, etc.). Entity State PDUs can contain several multiple entity extension records, each with different entity types. The intent of the Multiple Entity extension records is primarily to be able to add large numbers of entities efficiently. The multiple entity extension record would be a good choice for adding cars in a parking lot, people in a city, or background traffic on city roads. They may also be excellent for creating SAM threat laydowns or Early Warning radar locations where systems of the same type may be grouped together. Perhaps an aircraft formation or tank platoon makes sense to send using the Multiple Entity extension record. It will take some experience to figure out the best use cases and how to leverage these PDUs to reduce bandwidth and increase the numbers of entities that may be simulated.

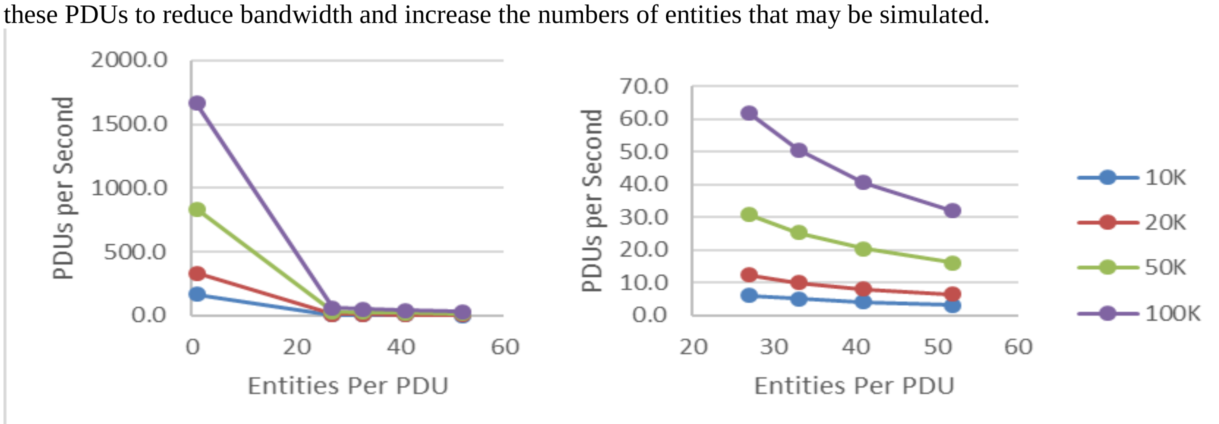

Figure 5 PDU's Per second based on four entity counts (10K/20K/50K and 100K). Left with DIS V7 and one entity per PDU (1666 PDUs/sec@100K entities) included, right with only DIS V8 shown so that the PDUs per second scale can be expanded. DIS V8 supports 41 Static Entities using Multiple Entities per PDU (~40 PDUs/sec@100K entities)

IMPROVEMENT 6: PARTIAL UPDATES, SPLIT MODELING AND RELIABILITY

Because of the new PDU format simulations may now provide only information for extension records that have changed in the PDU or may send some extension records with heartbeats that are slower than the default PDU update rate. For example, information about the infrared signature of an entity may be sent with a heartbeat of 30 seconds while the position updates are provided with five second heartbeats. This minimizes the sending of redundant information and makes DIS V8 more efficient.

The partial updates also allow one system to update one set of extension records, while a second separate system updates a different second set of extension records. Receivers combine all of the updates into a single object state.

With DIS V8 it is no longer necessary to have two versions of SIMAN PDUs in order to have reliability. All SIMAN PDUs may optionally require reliability, and other PDUs may have reliability added using extension records.

IMPROVEMENT 7: XML SCHEMA SUPPORTS MARSHALLING AND WIRESHARK DISECTORS

A major advantage of a simpler and consistent PDU syntax is that it will enable a formal syntax description language, referred to as machine-readable DIS syntax. This language in an Extensible Markup Language (XML) format that describes the exact on-the-wire binary protocol of every PDU.

The machine-readable XML language enables automatic code generation of the marshalling software and Wireshark dissectors. It also allows custom records to be formally described and communicated to all participants in a DIS exercise.

By limiting PDUs to a single form of variability, the complexity of formally describing the syntax is greatly simplified. Likewise, automated software processing of the format is also easier. To be clear, the XML language is not used to convey run-time DIS information, but instead to describe the syntax of the binary PDUs that carry the information.


```xml
▼<PDU name="Entity State" PDUType="1" protocolFamily="1" baseLength="96">
   <FixedRecordField name="PDU Header" type="PDUHeaderRecord"/>
   <FixedRecordField name="Entity ID" type="EntityIdentifierRecord"/>
   <EnumField name="Force ID" type="enum8" enumTableUID="6"/>
   <EnumField name="Dead Reckoning Algorithm" type="enum8" enumTableUID="44"/>
   <FixedRecordField name="Entity Type" type="EntityTypeRecord"/>
   <AdaptiveRecordField name="Entity Appearance" type="EntityAppearanceRecord" discriminant="Entity Type"
   field1="Entity Kind" field2="Domain"/>
   <AdaptiveRecordField name="Capabilities" type="CapabilitiesRecord" discriminant="Entity Type" field1="Entity
   Kind" field2="Domain"/>
   <FixedRecordField name="Entity Location" type="WorldCoordinatesRecord"/>
   <FixedRecordField name="Entity Orientation" type="EulerAnglesRecord"/>
   <FixedRecordField name="Entity Marking" type="EntityMarkingRecord"/>
   <NumericField name="Padding" type="uint32"/>
   <NumericField name="Padding" type="uint16"/>
 ▼<ExtensionRecordSet>
     <CountField name="Number Of Extension Records" type="uint16"/>
     <ExtensionRecordFields/>
   </ExtensionRecordSet>
 </PDU>
▼<ExtensionRecord name="Articulated Parts Extension Record" recordTypeEnum="2000">
   <EnumField name="Record Type" type="enum32" enumTableUID="66"/>
   <NumericField name="Record Length" type="uint16"/>
   <NumericField name="Padding" type="uint8"/>
 ▼<Array>
     <CountField name="Number Of Articulated Parts" type="uint8"/>
     <FixedRecordField name="Articulated Part" type="ArticulatedPartRecord"/>
   </Array>
 </ExtensionRecord>
```


Figure 6 Partial example DIS V8 Entity State XML

Figure 6 is an example of a partial syntax description of an Entity State PDU. It conforms to a schema (not shown) that dictates the format and field types that are allowed in PDUs, fixed records, and variable records.

IMPROVEMENT 8: NEW PDUS

DIS V8 adds the following new PDUs: Application Control, Identification Friend or Foe Interactive (IFF-I), Laser (formerly Designator), and Weather among others.

The Application Control PDU is used to request control of an application, when data is required from an application, or when an application is required to change its parameters. The data required is typically for the express purpose of subsequently controlling the application. The Application Control PDU provides a way to control loggers, simulators, gateways, and other applications in order to allocate resources for a particular exercise, to monitor system health, to coordinate system operations, to modify filter settings, to set and restore checkpoints and to deallocate resources at the end of an exercise.

The IFF-I PDU will support more detailed modeling of IFF interrogations than the current locally regenerated IFF approach can support. It will be an optional addition to the existing IFF capabilities.

The Laser PDU is based partially on the DIS V7 Designator but has been overhauled and modified significantly to support not only laser designators but also laser ranging, jamming, weapon guidance, laser pointer/marker as well as target illumination.

A new Weather PDU is being designed that is largely based on the SISO-STD-013-2014 Common Image Generator Interface (CIGI) standard [see SISO (2014)]. The CIGI weather messages have been condensed and the geometry definitions have been separated from the actual weather description contents data in order to make the messages more flexible as well as using less bandwidth. CIGI only defines rounded rectangular regions. DIS V8 has added the ability to define rectangular, circular, tubular, polygonal and linear regions. We have added linear velocity vectors and rotational rates to the regions. We have also created a way for multiple regions to be combined into a group so that they can move and rotate together or have common layers and weather definitions. This PDU will allow users to specify cloud layers, atmospheric parameters, maritime surface conditions, wave state, and terrestrial surface conditions in order to better synchronize the weather representation in all DIS participants simulators. It will support the definition of a set of current weather, or it may also be used to create dynamically changing weather. DIS simulations that currently use CIGI for their IG interface should be able to ingest this data and easily convert it to CIGI messages in order to get the desired visual weather effects. The information can also be used internally by sensor simulations and platforms to create more realistic training effects.

IMPROVEMENT 9: GRIDDED DATA

In addition to the new weather PDU based on CIGI previously discussed, the Gridded Data PDU has been simplified to support weather data and trench/terrain modification data that is fully gridded. Many historical and live data sources for weather contain data that is organized as gridded data tiles. It was decided to support this type of historical and live data by defining a simplified (compared to DIS V7) set of gridded data that may contain weather data for each grid tile. For example, Figure 7 shows a 10° by 10° notional coverage area for a database. Each tile in the grid would be assigned a unique number (see the small numbers 1 to 100 in Figure 7). In this example, each degree of latitude/longitude geocell has a PDU with 36 (6x6) 2D samples of data. This represents a sample size of 10’ or roughly 10.0nm = 18520 meters. The Gridded Data PDU supports this type of tile definition with each tile information being able to have desired characteristics. We have proposed a basic 2D Wind extension record that describes wind speed, wind gust speed, and direction for each defined tile. We have also considered other common weather data such as humidity, temperature, barometric pressure, etc. Future weather users may propose additional extension records for gridded data weather to be added to SISO-REF-030.

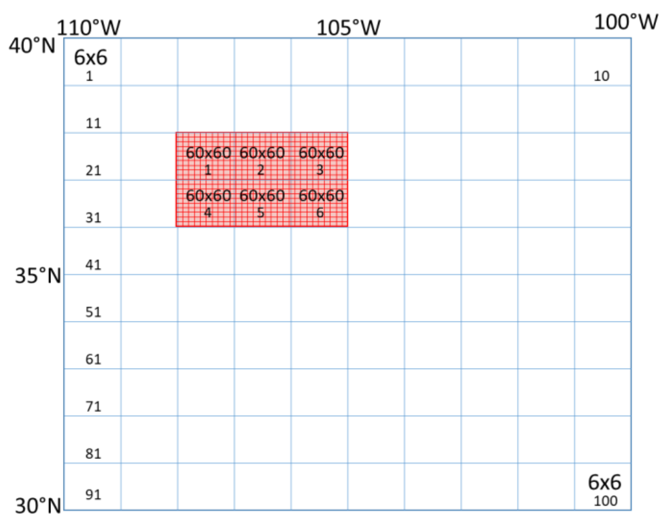

Figure 7 10x10 geocell gridded data example with 100 tiles. Most tiles contain 36 samples (6x6) but high-resolution tiles use 3600 (60x60) samples.

In addition to 2D Lat/Lon tiles it is possible to create 3D samples in order to create a 3-dimensional grid where each tile would then be a 3D box rather than a flat 2D rectangle.

Note that in Figure 7 the size of the red sample grids that have tiles with 3600 (60x60) samples. This represents a higher fidelity area of weather (1’=1.0nm=1852 meters). The Gridded data PDU supports the concept of higher resolution data insets where more fidelity is desired in certain areas while using lower fidelity tiles to cover a larger area.

Figure 7 is just an example and does not infer that all grids must be of this size. The user defines the size of each 2D or 3D tile and how they tessellate together. Each PDU defines a complete tile. How tiles are tessellated is not explicitly defined. Grids may not be fully populated with tiles if they are not needed or applicable.

Current estimations of bandwidth usage indicate that it would be possible to update gridded weather that has Humidity, Visibility, Precipitation Type and Rate, Temperature, Barometric Pressure and 3D Wind with a grid resolution of approximately 1 cubic Km (0.5' Lat/Lon with 20 vertical layers (max altitude 20000 meters)) that is refreshed every five minutes would require 107 PDUs per second and 4.73 Megabits per second per 100 Lat/Lon (10x10) Geocells (see Figure 7).

The gridded data may also be used to represent trenches, ocean temperature/salinity or other data that is organized into a grid. The goal in DIS V8 was to simplify the complex, but very flexible gridded data definition into something that might be less flexible, but more straight forward to use in the three identified possible use cases of weather, ocean environment modeling, and trenches. The second goal was to provide extension records for basic gridded weather.

IMPROVEMENT 10: MISCELLANEOUS UPDATES

In the 1990's when DIS was created the most common computers were Big-Endian (most to least significant byte order for multi-byte numeric values). Today, most computers used in simulation are Little-Endian (least to most significant byte order). This means that most DIS V7 applications today must swap the current Big-Endian network byte order into the local byte Little-Endian order when marshalling or unmarshalling data from the network. DIS V8 will change to Little-Endian byte order in order to make marshalling and unmarshalling data simpler as well as faster and more efficient for most applications.

DIS V8 will use a 64-bit timestamp with units in µsec instead of the one hour 32-bit timestamp used in DIS V7 and earlier. The default epoch is 1 Jan 1970 00:00:00 UTC. This is the standard UNIX epoch most commonly used by computer systems. This new timestamp will improve the time resolution from 107 µsec to 1 µsec. It will also eliminate issues with the time rolling over every hour. The timestamp will include the date as well as the time. This will allow for computation of moon phases and ephemeris calculations. One potential issue with this approach is that now the Date, Time zone, and Daylight savings settings on computers using a DIS interface must be coordinated across all exercise participants. It is expected that most DIS exercises will select the zero-time offset time zone (Greenwich Mean Time) and no Daylight savings as their standard time settings. It will be possible to select other Epoch's by exercise agreement.

Point, Linear, and Areal objects have been merged with entities. The Point, Linear, and Areal Object PDUs are now extension records in the Entity State PDU. Environment objects now use Entity IDs as the unique identifying number. All references to Object IDs have been removed. This should reduce confusion about when to use environment objects or entities and ensure that IDs are unique.

CONCLUSIONS

The following summarizes the conclusions for the top 10 improvements:

1. The extensibility of DIS V8 will allow for adding new capabilities to DIS without needing to modify the IEEE 1278.1 standard which should lead to more innovation.
2. SISO-REF-030 will allow for faster updates of documented capabilities. This should lead to more reuse of existing extension records and increase the chances the updates will be interoperable.
3. DIS V8 will enable much better modeling of AESA radars and advanced jammers in a consistent, interoperable fashion.
4. The CPC dead reckoning algorithm will reduce network updates and reduce the size of network bandwidth peaks making better usage of available bandwidth.
5. The new multi-entity extension records will support larger exercises with fewer PDUs and less bandwidth.
6. General rules for partial updates for all PDUs improve bandwidth efficiency. Duplicate SIMAN PDUs are no longer required for reliability.

7. An XML machine-readable DIS syntax description enables significant automation of creating and processing DIS PDUs.
8. New PDUs add new capabilities and higher fidelity.
9. Simplified Gridded Data PDU with support for dynamic terrain and real-world weather data will allow users to use real-time weather or historical weather data to create more realistic training.
10. Miscellaneous updates for ease of use, faster computation, simplification, and less redundancy will provide better performance and make DIS V8 less confusing, especially for new users.

REFERENCES

IEEE. (2012). IEEE Std 1278.1™-2012 *IEEE Standard for Distributed Interactive Simulation – Application Protocols*. The Institute of Electrical and Electronics Engineers, Inc.

SISO. (2023). SISO-REF-010-2023 *Reference for Enumerations for Simulation Interoperability*. Simulation Interoperability Standards Organization.

IEEE. (2015). IEEE Std 1278.2™-2015 *IEEE Standard for Distributed Interactive Simulation – Communication Services and Profiles*. The Institute of Electrical and Electronics Engineers, Inc.

I/ITSEC. (2021). Interservice/Industry Training, Simulation and Education Conference 2021 Paper 21274 *SISO C-DIS Techniques and Performance*. National Defense Industrial Association.

SISO. (2024). SISO-STD-023-2024 *Standard for Compressed-Distributed Interactive Simulation (C-DIS)*. Simulation Interoperability Standards Organization.

SISO. (2014). SISO-STD-013-2014 *Standard for Common Image Generator Interface (CIGI) 4.0*. Simulation Interoperability Standards Organization.

Call, Lance. (2022). *AESA Radar and Advanced Jamming – A CAF DMO Approach*. Simulation Interoperability Standards Organization paper 2022-SIW-006.

Call, Lance. (2023). *Advanced Radar and Jamming in DIS V8*. Simulation Interoperability Standards Organization paper 2023-SIW-001.

Murray, Robert. (2018). *Gen3: What's Up with DIS Version 8*. Simulation Interoperability Standards Organization paper 2018W-SIW-028.

Murray, Robert. (2019). *A Simpler Dead Reckoning Algorithm with Better Performance*. Simulation Interoperability Standards Organization paper 2019-SIW-034.