Battle Damage Computation Server

Hung Tran
Tactical Systems, CAE USA
Tampa, FL
hung.tran@caemilusa.com

ABSTRACT

This paper will present a new approach for handling the battle lethality computation in the context of a Distributed Interactive Simulation (DIS) network training scenario where participants in an exercise are required to broadcast their current damage status. Usually, each simulation handles the computation and the assessment of damage differently, resulting in an "unfair fight" between the participants. Rather than having each simulation performs this computation, an alternate approach would be to delegate the lethality computation to a common processing task implemented on a server. The design of the battle damage server will be described and discussed in this paper.

The advantage of the battle damage computation approach described in this paper is twofold: participants within the network would use a common mathematical model, and simulations being freed from the burden to compute the battle lethality. This approach will help to eliminate the interoperability variances in lethality results and achieve a "fair fight" weapon effect.

ABOUT THE AUTHOR

Hung Tran is a Tactical Systems Group Leader at CAE USA. He joined CAE USA more than 20 years ago and has worked on the modeling and simulation of several Electronic Warfare (EW) systems. Hung was the lead designer of the Computer Generated Forces (CGF) currently used by all USAF C-130J Weapons System Trainers (WST).

Hung holds a Bachelor of Science in Electrical Engineering from Montreal Institute of Polytechnic, Canada.

Battle Damage Computation Server

Hung Tran
Tactical Systems, CAE USA
Tampa, FL
hung.tran@caemilusa.com

INTRODUCTION

Generally, battle simulation lethality consists of two distinct parts: the representation of aircraft intercept conditions and the lethality computation caused by the munitions' detonation (Paul Zarchan, 2013). This paper will focus on the second part, the lethality computation. In the context of a Distributed Interactive Simulation (DIS), simulation players within the network are required to report their current damage status using "fuzzy" linguistic terms such as: slight damage or moderate damage (IEEE, 1998; ISS, 2003). The DIS standard does not neither offer the definition nor specify the method to compute aircraft damage; therefore each simulation handles the computation and the assessment of damage differently, resulting in an "unfair fight" between players. For this specific reason, a common and simple battle lethality computation is required.

The motivation of providing a common and simple battle lethality computation is twofold:

1- Eliminate the interoperability variances in lethality results to achieve a "fair fight" because all simulations within the network will resolve the weapon lethality effect using a common approach.
2- Provide a simple and consistent mathematical method to determine the damage status of the aircraft. Instead of having individual simulation perform this computation, a better approach would be to delegate this lethality computation to a common processing task implemented on a server within the DIS network. This approach will ensure that all participants in the DIS exercise consistently use the same computation model.

The objective of this paper is to present to the Modeling and Simulation community a simple model that can be easily implemented into existing simulation environments so that the lethality battle simulation can be performed consistently by all participants in a networking exercise, therefore the interoperability variance in lethality results can be eliminated. The following fundamental elements of this simulation approach will be described and explained in this paper:

- The computation of the probability of kill for a single detonation (PK_{det}) using the munition lethal range (LR) as a single parameter of the mathematical model. Since the munition lethal range is already used widely by the simulation community in various applications, the introduction of this parameter into the lethality computation can be performed without any confusion.
- The probability of kill cumulative (PK_{cum}) due to weapon detonation. As oppose to most approaches found in the simulation and training literature where the computation of the damage is limited to the computation of the probability of kill due ***solely*** to weapon detonations (Mann, Fisher, Krauss, Lowndes and York., 2006; N. Prabhakar, B. Larihi and C. Sangeetha., 2008), this paper introduces a novel approach that will include the simulation of aircraft component failures into the computation of the probability of kill.
- The utilization of the membership function (fuzzy set) to assess the aircraft damage expressed in natural language by simple mathematical terms. The advantage of using this original approach will be explained and discussed.
- The design of a server within a DIS network to enable a common damage computation for all participants in the exercise.

LETHALITY MODEL

There are many approaches for modeling the vulnerability of an aircraft due to a weapon detonation in the vicinity (Hinrichsen, Kurzt, Wang and Belcastro, 2008; Gautam H. Shah and Melissa A. Hill, 2008). The *vulnerability area approach* (Emil J. E., 1989) is the most simple and convenient method to use in a virtual aircraft simulation environment. This approach divides the aircraft into multiple cells or sectors. For example, the aircraft can be divided into either two sectors (front and rear) or four sectors (front left, front right, rear left, and rear right). The critical components of the aircraft (i.e., engines, flaps, controls) belong to each sector can be damaged if hit by warhead fragments. The effect of damage or malfunction of these aircraft components is normally already modeled and simulated in aircraft simulators; therefore they can be used in the battle damage computation.

When munitions detonate in the ownship's vicinity, three possibilities can result: undamaged, damaged, or destroyed. Damaged aircraft can be simulated by enabling corresponding simulated malfunctions of the affected sectors. The failure of these aircraft components should eventually contribute to the vulnerability of the aircraft over time. For instance, an engine on fire due to a missile detonation in the vicinity will eventually affect the aircraft's ability to evade the next missile attack.

Figure 1 represents the battle lethality simulation block diagram. The *target intercept geometry* consists of the end-game computation, where the detonation parameters, such as detonation location with respect to the target and missile intercept velocity, are determined in real-time. The *warhead model* in the figure represents the fuse operation of the munitions. Upon a warhead detonation, there are three means by which the aircraft can be killed: direct hit, blast, or aircraft components failure. The direct hit case is not considered in this model because it is normally simulated through an Instructor Operational System (IOS) command. A blast kill will happen if the detonation is strong enough to create an overpressure that can destroy the aircraft instantly. The warhead particle kill is due to aircraft components failure.

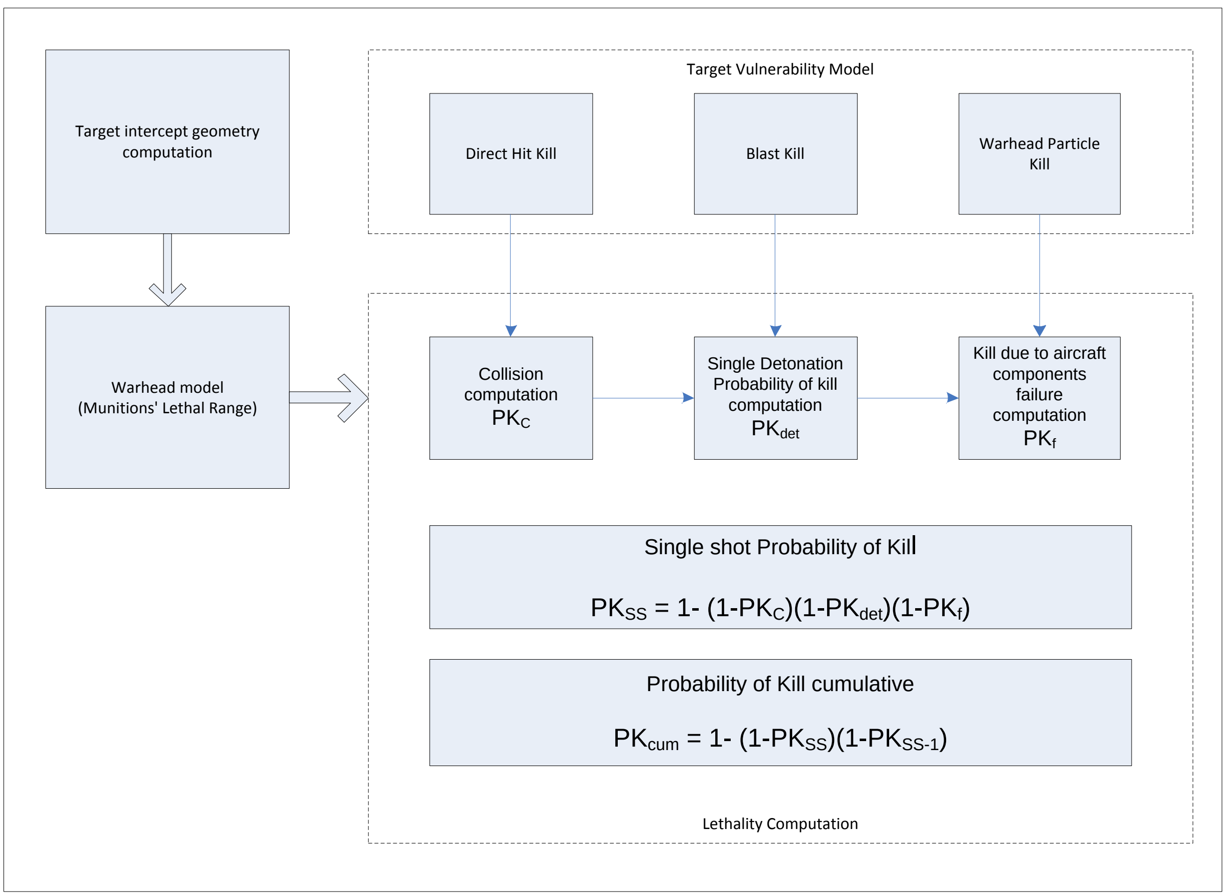

Figure 1. Lethality simulation model block diagram.

First, the probability of kill for a single munition detonation can be modeled and computed using the following equation:

$$PK_{det} = e^{-k(\frac{SR}{LR})} \qquad (1)$$

where PK_{det} is the probability of kill due to a single detonation, SR is the slant range between the target and the munition detonation, LR is the weapon lethal range, and "k" is a calibration constant.

The constant "k" in the equation (1) represents a weighting parameter normally used to calibrate PK_{det} based on the munition type and available data. For instance, set the value of the constant "k" equal to 0.69315 to calibrate PK_{det} in a way that if the slant range (SR) between the detonation and the aircraft is exactly equal to the munition lethal range (LR), PK_{det} will be equal to 0.5. An alternate approach to compute PK_{det} is to obtain the lethal range and its corresponding probability of kill from sources such as the Joint Munitions Effectiveness Manual (JMEMS). This alternate approach will involve a common database with the PK_{det} and the lethal range defined for munition/aircraft type pairs of interest.

Secondly, the probability of kill due to aircraft component failure can be computed as follows:

$$PK_f = 1 - \prod_{i=1}^{i=n}(1 - PK_i) \qquad (2)$$

where PK_f is the probability of kill due to "n" aircraft component failures. PK_i is the probability of kill due to the failure of component "i".

The probability of kill due to component failure for each aircraft component can be assigned by subject matter experts (SME) on that simulation device. Obviously, these values must be assigned based on (a) aircraft type and (b) survivability of the aircraft providing a specific failure is present. A representative set-up could look like those shown in Table 1.

Table 1. Vulnerability Area Table with Probability of Kill Due to Component Failure

Sector	Failure 1	Failure 2	…….	Failure n
Front Left	Engine 1 fire (PK = 0.6)	Outflow valve failed (PK = 0.3)	----	MWS FL Sensor failed (PK = 0.4)
Front Right	Engine 4 fire (PK = 0.6)	Battery relay failed (PK = 0.1)	----	----
Rear Left	----	----	----	----
Rear Right	----	----	----	MWS RR Sensor failed (PK = 0.4)

Using the probability of kill for a single detonation (PK_{det}) and the probability of kill due to aircraft component failure (PK_f), the single shot probability of kill can be computed as follows:

$$PK_{SS} = 1 - (1 - PK_{det})(1 - PK_f) \qquad (3)$$

Because PK_{SS} represents the probability that the aircraft is killed either by a detonation or by aircraft component failure, therefore the blast kill can be simulated by comparing a random number with PK_{SS}. If the blast kill did not occur, simulated malfunctions of the affected sector will be enabled and the aircraft is considered damaged.

Finally, the cumulative probability of kill can be defined as the probability that the aircraft is killed due to the current detonation or due to the previous detonations.

$$PK_{cum} = 1 - (1 - PK_{SS})(1 - PK_{cum-1}) \qquad (4)$$

where PK_{cum} is the probability of kill cumulative.

The value of PK_{cum} provides an indication of the vulnerability of the aircraft for the next weapon attack; therefore it can be used to determine the current damage status of the aircraft.

MEMBERSHIP FUNCTIONS OVERVIEW

Fuzzy logic was formulated by Lofti Zadeh in 1965 (Zadeh L. A, 1965) as the way to process elements that are partially in a set. In mathematics, a set represents a collection of elements that shared common characteristics, for example, a set of "big" aircrafts. One can define an aircraft with a length greater than 100 feet as "big". This set can be graphically represented as in Figure 2.

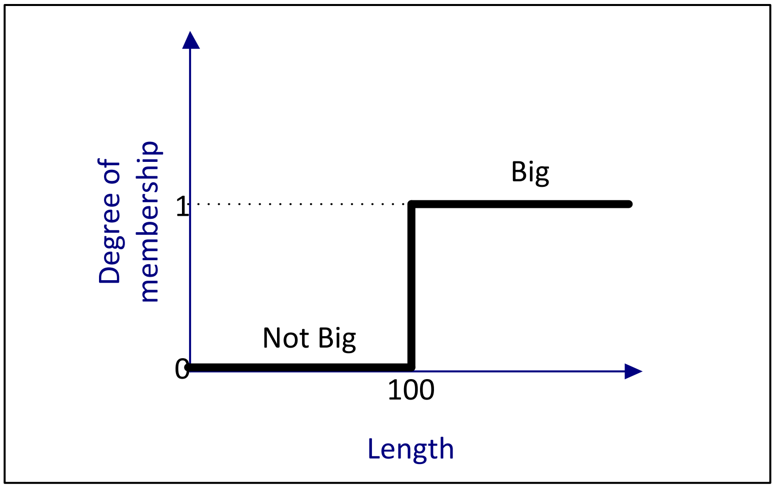

Figure 2. Crisp set – size of aircraft

We can observe that the main characteristic of this set is that an element is either a member or not a member of this set; also, it does not make any distinction between aircrafts with a length of 100 and 200 feet: they are both "big" aircraft. On the other hand, an aircraft with a length of 99.5 feet is "not big" even though it is only half a foot shorter than the cutoff length. This mathematical function works well for binary operations, but not for real world situation such as "moderate big" or "very big".

Compared to the classical set, a fuzzy set allows members to partially belong to the set. This concept is graphically presented as in Figure 3.

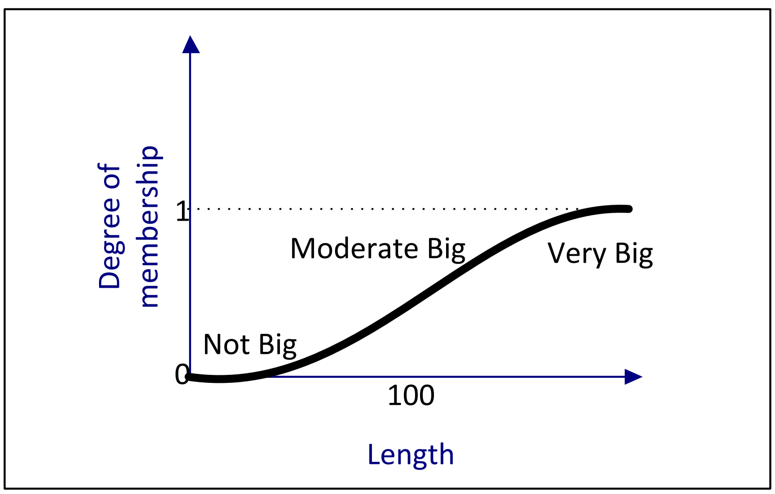

Figure 3. Fuzzy set - size of aircraft

We can observe that there is no clear cut between the "not big", "moderate big" and "very big" categories. The transition between these categories of aircraft is smooth. In order to use the computer to process linguistic variables, such as "moderate big" or "very big", a crisp input, the length of the aircraft, can be converted to fuzzy outputs using a simple membership function similar to the one shown in Figure 3.

In summary, fuzzy logic can be used to handle the concept of partial truth, as oppose to the conventional Boolean logic where an event must be either true or false. In the next section of this paper, we will describe how fuzzy logic can be used to assess the ownship battle damage status in a virtual flight simulation.

DAMAGE ASSESSMENT MODEL

In the context of a DIS network, the aircraft simulation is required to use "fuzzy" linguistic terms to report its damage status: no damage, slight damage, moderate damage, or destroyed. While the conditions "no damage" or "destroyed" are concise and easy to determine, damage conditions such as "slight damage" or "moderate damage" are fuzzy because it is difficult to decide the border between these two levels of damage. This concept becomes important in the context of a "fair fight" between participants in the same networking exercise. To achieve a fair fight of weapon effect and eliminate the interoperability variance in lethality results, two conditions must be met:

1- Participants in the same exercise must use common lethality computation method, and
2- Damage status such as slight or moderate damage must be well defined and consistently applied.

The lethality computation model as described in the previous section provides a simple and convenient way to compute the aircraft battle damage using the concept of probability of kill. This section will describe the damage assessment model.

To categorize the aircraft damage status, logical rules such as those normally found in the Simulation and Training literature can be used:

- ***If the number of aircraft components that failed as a result of a warhead detonation is greater than a predetermined value, then the damage state is "moderate," otherwise the damage is "slight"***. While simple, this rule is obviously inconsistent because the number of aircraft components that failed is irrelevant; the nature of the failures is much more relevant to determine the overall damage state. For instance, a failure of the countermeasure system will increase the vulnerability of the aircraft during the next missile attack. For this specific reason, the computation of the probability of kill cumulative as described in this paper takes into account not only the number of failed aircraft components, but also the probability of kill due to the failure of these components. The resulting probability of kill cumulative can then be used to determine the damage status.

- An alternate rule could be: ***if the computed probability of kill is greater than a pre-determined number, then the damage state is "moderate," otherwise the damage is "slight"***. Assuming that a cumulative probability of kill (PK_{cum}) equal to 0.5 is defined as the cut between "slight" and "moderate" aircraft damage, then a computed PK_{cum} greater than 0.5 will result in "moderate" damage, and a PK_{cum} less than 0.5 will result in "slight" damage. This approach provides a better assessment of aircraft damage because the probability of kill cumulative (PK_{cum}) computed by equation (3) already takes into account the probability of kill due to the aircraft components failure. However, this method can result in conflicting assessments because the computation of two almost identical intercept conditions will lead to two numerically similar PK_{cum} values, but two totally different damage assessment results. In this example, a PK_{cum} of 0.499 yields a "slight" damage state while a PK_{cum} of 0.501 yields a "moderate" damage state.

Membership Functions (Timothy J. Ross, 2010) can be used to assess the aircraft damage status. As explained in the previous section, the main characteristic of a membership function is that the boundaries are not precise. To categorize the ownship damage status, we can define a fuzzy set for aircraft damage status that will contain four levels of damage: no damage, slight damage, moderate damage, and destroyed.

Figure 4 illustrates the concept of using a fuzzy set in comparison to using the traditional crisp set to assess the aircraft damage status. In the traditional crisp set, the aircraft is considered slightly damaged if the probability of kill PK_{cum} is lower than 0.5, otherwise the damage will be considered moderate. For the fuzzy set, a PK_{cum} between 0.2 and 0.8 is belonging to both damage variables. For instance, if PK_{cum} is equal to 0.5, it will belong to both damage variables with a same degree of membership of 0.75 as shown in Figure 2. In other words, it has an equal probability to belong to either the "slight" or "moderate" damage variable. Another example is if PK_{cum} is equal to 0.4, it will provide a membership value of 1.0 for "slight" damage and a membership value of 0.5 for "moderate" damage; therefore it has twice the chance of belonging to the "slight" damage variable than to the "moderate" damage variable. Nonetheless, the "moderate" damage is not excluded from the damage assessment.

The method to assess the aircraft damage status using the membership function is to normalize both membership values and compare with a random number. For instance, in our example of PK_{cum} equals to 0.4, membership value for slight damage is 1.0 and membership value for moderate damage is 0.5, thus normalized membership values for slight and moderate damage will be approximately equal to 0.66 and 0.33 respectively. Therefore, a random number less than 0.33 will categorize the damage as moderate, otherwise the damage is slight.

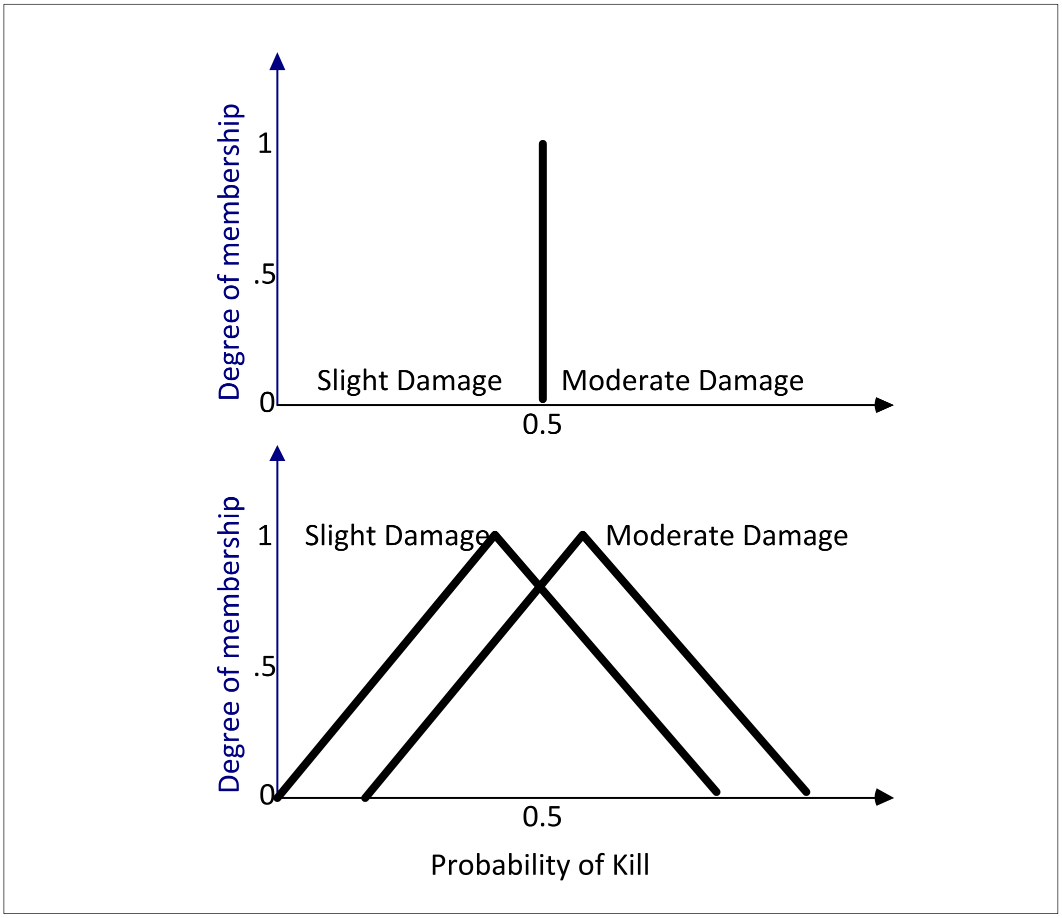

Figure 4. Probability of kill conceived as (a) a crisp variable and (b) a fuzzy set variable

In summary, we can observe that for the traditional crisp set, a computed value of PK_{cum} is exclusively belongs to either "slight" or "moderate" variable; while for the membership function set, it can belong to both variables, and each of them is expressed with a degree of membership.

In the preceding example, we use the triangular shape to represent the damage variables; many others shape, such as trapezoidal or bell, have been used by different applications (George J. Klir and Bo Yuan, 1995). Normally, a specific shape is chosen based on the specific application and availability of data.

DESIGN OF THE DAMAGE COMPUTATION SERVER

The purpose of having a damage computation server is to allow all participants on the network access to a common lethality computation approach. For the lethality computation application point of view, the server accepts requests from simulation participants and returns the computed damage assessment.

The battle damage server is designed for the DIS environment; therefore, it must satisfy the following requirements:

1- It must accept requests and return damage assessments.

2- The lethality computation server must maintain a look-up table that provides the lethal range based on the type of weapon. This look-up table must be expandable for future growth.

3- The lethality computation server must maintain an internal database that will provide the damage assessment state of all participants in the network. When a player subscribes to a DIS exercise, the server signs this player automatically and sets the initial damage status of this player to “no damage”.

4- Any participant can request its damage status at any time during the exercise.

5- Request/Return latency must be minimized for real-time application over a network. Typical maximum latency values should be between 40ms and 100ms to maintain real-time interaction between participants and not to interfere with the fidelity of the simulated exercise (C. Diot and L. Gauthier, 1999).

The participants on a DIS network have the responsibility to monitor warhead detonation events and decide if a lethality computation request is necessary. If a request is made to the server, the request must provide:

1- Specific DIS detonation ID and Ownship Entity ID. Using the information of the detonation PDU and entity PDU, the probability of kill for a single detonation PK_{det} as described in the equation (1) is computed.

2- The probability of kill due to aircraft component failure PK_f. This probability must be computed by simulation participants based on aircraft simulated malfunctions that are currently enabled. Using this information, the server will compute PK_{SS}.

3- The server automatically computes the damage of all subscribed players based on the mathematical model described above and stores the results in its internal database.

The design of the lethality server should be flexible enough to be able to be used in a stand-alone, as well as a networked application. We considered two types of servers: a subroutine that can be called within simulation for a stand-alone application or a network server on a DIS network.

Figure 5 represents the schematic of the callable subroutine approach. Aircraft damage status requests can be done via an appropriate Application Program Interface (API) call. In this configuration, the damage computation server has no knowledge and information of the DIS network, therefore the simulation client must provide data contained in the munition detonation and entity PDU.

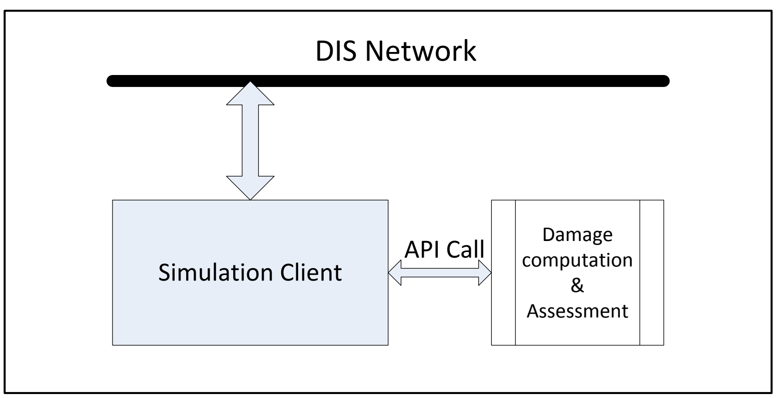

Figure 5. Callable subroutine within the simulation

Figure 6 represents the schematic of the networking approach where the server is connected directly to the DIS network. In this configuration, the damage computation server is connected directly to the DIS network, therefore only the IDs of munition detonations and entities are required for the server to compute the damage assessment.

Generally, data latency depends mainly on the computer processor speed; however for this specific application, network traffic represents an important factor that will affect data latency.

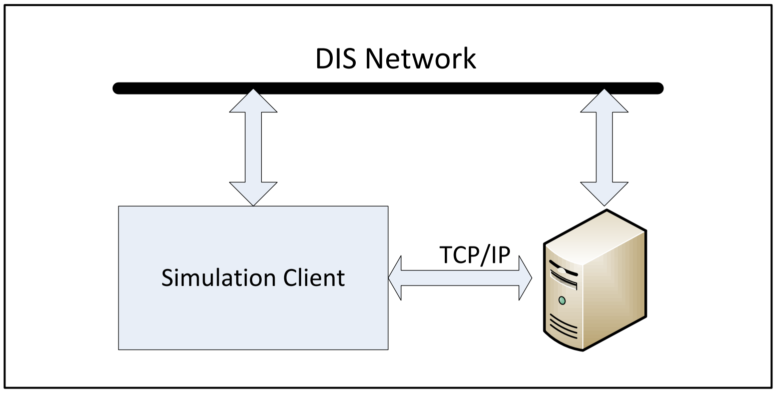

Figure 6. Server on DIS network

Informal tests performed in a lab environment demonstrated that data latency is negligible when using the callable subroutine architecture. The server was able to return the damage status within the same iteration frame that the request was made. With the server connected to the DIS network, test data shows that latency increased, but the server is still fast enough to handle several consecutive requests; the latency rarely exceeds one iteration frame.

EXAMPLE OF SIMULATION

The callable subroutine approach was used to illustrate the lethality computation approach presented in this paper. A missile engagement simulation was performed with a C-130J Virtual Weapon Systems Trainer (WST) with the following parameters: the ownship flying straight and level at an altitude of 5,000 feet, with an airspeed of 120 kts and a generic Surface-to-Air Missile (SAM) site located approximately 5 miles from the ownship at about 2 o'clock engaging the ownship using a short-range infrared missile. For this simulation, a missile lethal range (LR) of 25 ft. is set into the server look-up table for this munition.

Table 2. Results of lethality computation obtained from a stand-alone simulation

Engagement	Intercept Range (ft.)	PK_{det}	PK_f	PK_{SS}	PK_{cum}	Damage
1	51.5	0.2398	0.10	0.3158	0.3158	Slight damage
2	48.2	0.2628	0.10	0.3365	0.5461	Slight damage
3	49.3	0.2549	0.15	0.3667	0.7125	Moderate damage
4	47.2	0.2702	0.23	0.4380	0.8384	Destroyed

Table 2 represents the simulation results. We can observe the following:

1) Depending on the intercept aspect and velocity, the intercept range between the detonation and the ownship is approximately 50 ft., which is the warhead's proximity fuse value used for this simulation.
2) As expected, the cumulative probability of kill increases after each engagement resulting from missile detonation.
3) The cumulative probability of kill after the second engagement is approximately equal to 0.55; nonetheless, the damage assessment is still qualified as "slight damage". If the conventional crisp set was used in this simulation, the damage assessment would be "moderate damage".
4) The ownship is destroyed after the fourth engagement.

CONCLUSION

This paper proposed a new approach to compute the ownship battle simulation lethality using the single shot probability of kill (PK_{SS}); this simple mathematical model required only the munition lethal range (LR) as a single parameter of the model. The mathematical model described in this paper introduced the utilization of the simulated aircraft component failure into the computation of the probability of kill, as opposed to most approaches found in the simulation and training literature where the computation of the damage is limited to the computation of the probability of kill due ***solely*** to weapon detonations.

Critical aircraft components such as engines, countermeasure systems, surface controls, etc. that contribute directly to the aircraft vulnerability are included in the model, making the computation of the probability of kill more accurate. For instance, assuming a scenario where a missile is detonated at its lethal range; if the probability of kill due to aircraft components failures was not taken into account in the computation of the probability of kill, the computed probability of kill cumulative (PK_{cum}) would be underestimated by approximately 25% after the first missile detonation and as much as 45% after the second one.

Furthermore, the ownship battle damage assessment was determined using a fuzzy set instead of using a conventional crisp set. The utilization of a fuzzy set in the lethality computation is necessary because we are dealing with an ***assessment*** problem that required an underlying mode of reasoning which is approximate rather than exact. Because of that, the assessment of the lethality damage using the method proposed in this paper is more balanced as it reflects normal human reasoning.

In a networking exercise that involved many participants, one of the most challenging aspects is "***fair fight"*** where the damage status of players must be fairly assessed. This fair fight aspect can be addressed if all participants used a common approach. Rather than having each participant performing their own damage computation, this paper proposed to delegate this lethality computation to a common processing task implemented on a server. The server can be either within the simulation or within the network. In a DIS exercise context, if the proposed battle damage server is used, the interoperability variances in lethality can be eliminated and a fair fight can be achieved. Based on our internal marketing data, we estimate an approximate number of 20 to 25% of simulators in-service do not have a battle damage model implemented within the simulation. Hence, using the damage server as described in this paper will constitute a quick turnaround to be able to participate in a networking exercise without having to spend a significant amount of effort to develop a lethality computation model.

Finally, we continue our work to expand and adapt the battle damage server so it can be used within the High-Level Architecture (HLA) exercise network.

REFERENCES

John Mann, David Fisher, Matt Krauss, Erik Lowndes, Allen York (2006). An Analysis of Engagement Algorithms for Real-time Weapon Effect. Journal of Defense Modelling and Simulation Application, Vol 3, No 3.

N. Prabhakar, Bidisha Larihi and C. Sangeetha (2008). Damage Assessment Software Program. Defense Science Journal, Vol 58, No 6.

Paul Zarchan. (2013). Tactical and Strategic Missile Guidance. Sixth Edition. Progress in Astronautics and Aeronautics series volumes.

Emil J. Eichblatt. (1989). Test and Evaluation of the Tactical Missiles. Progress in Astronautics and Aeronautics series volumes.

Institute of Electrical and Electronics Engineers, Inc. (1998). IEEE 1278.1A Standard for Distributed Interactive Simulation - Application Protocols.

Institute for Simulation and Training (2003). Enumeration and Bit Encoded Values for Use with Protocols for Distributed Interactive Simulation Applications.

Gautam H. Shah, Melissa A. Hill. (2012). Flight Dynamics Modeling and Simulation of a damaged Transport Aircraft. American Institute of Aeronautics and Astronautics.

Ronald L. Hinrichsen, Alex G. Kurzt, John T. Wang and Christine M. Belcastro. (2008). Modeling Projectile Damage in Transport Aircraft Wing Structure. American Institute of Aeronautics and Astronautics.

Zadeh L. A. (1965). Fuzzy Sets. Intl J. Information Control.

George J. Klir, Bo Yuan. (1995). Fuzzy Set and Fuzzy Logic, Prentice Hall PTR.

Timothy J. Ross. (2010). Fuzzy Logic with Engineering Application, Wiley.

Diot, C. and L. Gautier (1999). A Distributed Architecture for Multiplayer Interactive Applications on the internet. IEEE Network 13(4).