VULNERABILITY/LETHALITY SIMULATION ENHANCEMENTS (VLSE)

Russel E. Hauck, Irwin L. Hudson, Edison Atikune
Systems Research Group, Inc.
Orlando, Florida

Arthur Atikune
Nations, Inc.
Orlando, Florida

ABSTRACT

The VLSE Project is one of a number of projects initiated under the Live Fire Test and Training Program. The program is sponsored by the Live Fire Test & Evaluation Office, located within the OSD Directorate of Operational Test & Evaluation (DOT&E), and is being performed in partnership with the Naval Air Warfare Center Training Systems Division (NAWC TSD), the Army's Simulation, Training and Instrumentation Command (STRICOM), and the Air Force Agency for Modeling and Simulation (AFAMS). The common thread amongst the projects is to find ways to more closely integrate training and testing activities, including modeling, simulation, and shared use of data. The primary objective of VLSE is to develop and implement a methodology for more realistic (i.e., detailed) damage assessments in direct fire gunnery simulations through improved use of live fire test data and models. Specific objectives of this effort include:

- Use data and lessons learned from tank-on-tank degraded states initiatives in DSWARS and CASTFOREM to develop a conceptual degraded states approach to damage assessments in training simulations.

- Generate a degraded states demonstration model and integrate it with one or more virtual training systems used by the Army National Guard for tank gunnery training; and with a constructive simulation (i.e., SAF) to demonstrate and evaluate damage assessment representations via a degraded states approach.

ABOUT THE AUTHORS

Russ Hauck is a Vice President with Systems Research Group, Inc. (SRG) in Orlando, Florida and serves as Project Manager for the VLSE Project at SRG. Mr. Hauck has 20 years experience in the training systems community. He received a BS degree with a major in International Relations from the U.S. Naval Academy and a Master's degree in Public Administration from George Washington University.

Irwin Hudson is Program Manager for Information Systems at Systems Research Group, Inc. in Orlando, Florida. Mr. Hudson serves as Project Engineer for the VLSE Project. He received a BS degree in Computer Engineering with a Minor in Mathematics, from Mississippi State University. He is currently working on a Masters in Human Factors Engineering, along with a Masters in Business Administration at the University of Central Florida.

Edison Atikune is a Systems Engineer at Systems Research Group, Inc. in Orlando, Florida. He has extensive experience in the area of 3d Graphics ASIC designs, Visual C++ programming, and system administration. He has worked on both computer hardware and software design. He received a BS degree in Computer Engineering from the University of Central Florida.

Arthur Atikune is a Software Engineer at Nations, Inc. in Orlando, Florida, where he works closely with ModSAF as part of the configuration management team. He is experienced with online database design and implementation, website design, Visual C++ programming, and 3d programming using the Open GL API. Mr. Atikune graduated Cum Laude from the University of Central Florida Computer Engineering program.

VULNERABILITY/LETHALITY SIMULATION ENHANCEMENTS (VLSE)

Russel E. Hauck, Irwin L. Hudson, Edison Atikune
Systems Research Group, Inc.
Orlando, Florida

Arthur Atikune
Nations, Inc.
Orlando, Florida

INTRODUCTION

The VLSE Project is one of a number of projects initiated under the Live Fire Test and Training Program. The common thread amongst the projects is to find ways to more closely integrate training and testing activities, including modeling, simulation, and shared use of data. This effort is based on the following concepts:

- Direct fire training simulations for ground combat vehicles provide only general approximations of hit location, damage assessment, and "probability of kill" – and these are not standardized.
- There is significant potential value in having training simulations more closely replicate the results of actual engagements (i.e., damage assessments that reflect likely outcomes from specific target-weapon pairings).
- Live fire test data and simulation models from the test community can be used to improve the realism of training simulations.

PROJECT OBJECTIVES

The primary objective of VLSE is to develop and implement a methodology for more realistic training (i.e., detailed) damage assessments in direct fire gunnery simulations through improved use of live fire test data and models. Specific objectives include:

- Use data and lessons learned from tank-on-tank degraded states initiatives in DSWARS and CASTFOREM to develop a conceptual degraded states approach to damage assessments.
- Generate a degraded states demonstration model and integrate it with one or more virtual training systems used by the Army National Guard for tank gunnery training; and with a constructive simulation (i.e., SAF) to demonstrate and evaluate damage assessment representations via a degraded states approach.

BACKGROUND

In training simulations, the vulnerability methodology is the means by which the *probability of kill given the target has been hit* ($P_{K/H}$) is generated for each of the target-weapon pairings modeled in the simulation. Typically, these calculations are performed in advance and are resident in a set of look-up tables for the simulation. The values in the look-up tables represent the probability or likelihood that each of the defined kill levels occurred as a result of a hit on the target by a particular weapon; and the resulting damage to the target.

In general, direct fire training simulations of vulnerability for ground combat vehicles consider target and weapon type, range of the target from the firing unit, aspect angle of the target to the direction of flight of the weapon, and area of hit (i.e., for each aspect angle, the target is divided into a number of vulnerability zones). A hit on a specific target type, with a specific weapon type, fired at a specific range, in a specific vulnerability zone, at a specific aspect angle will result in a *probability of kill* assessment. The typical "kill" levels employed in most trainers are: catastrophic (K-KILL), firepower or functional (F-KILL), mobility (M-KILL), and combined mobility/firepower (M/F-KILL). It is also possible to generate a near miss or NO-KILL. The application of vulnerabilities to target silhouettes in this manner has been referred to as *vulnerability mapping*. The effect of a hit on target is assessed in this approach by drawing a uniform (0,1) random number and comparing that outcome to the assigned kill probabilities resident in a set of look-up tables.

The probability of kill (P_K) approach to vulnerability assessment is problematic.[1] These problems have been addressed in some detail within the test community since at least the late 1980's and it is not our intention to duplicate those concerns here, except to establish a rationale for considering a different approach. Probabilities of kill were originated as a means to simplify the very complex business of assessing damage and relating that damage to continued military utility of the target vehicle. But they are an oversimplification that does not allow for more detailed consideration of likely damage to personnel and equipment, because those results have been rolled up early on in the P_K calculations. If trainees sustain a mobility kill, their vehicle might be immobilized in the simulation, but they do not know what equipment has been damaged to cause the M-KILL and they do not have information regarding other systems which might have been damaged by the same hit. In fact, one of the causes of the M-KILL might be a driver casualty, but that information currently is not available to the crew – not even to the driver himself!

Current training simulations do not allow for degradations in one or more systems or for subsequent dynamic assessment of impacts on mission utility. Because degraded states are not considered, our virtual and live training systems do not afford an opportunity to realistically include crew responses to damage or battle damage repair actions. Consequently, our constructive simulations are lacking in dynamic realism. While this level of fidelity might not be a requirement in some current training systems, we believe that a higher level of "granularity" with respect to damage assessment will prove to be of great value – if our goal is to train as we fight – in mission planning and rehearsal and in training at all levels. Indeed, there are many potential advantages to having a more detailed assessment of damage, including the possibility of also considering variations in the vehicle's pre- and post-hit operating condition, variations in the operating environment, the effects of multiple hits, and secondary damage mechanisms and effects. Certainly, these are not new ideas to the testers. Dr. Paul Deitz, Technical Director of the U.S. Army Materiel Systems Analysis Activity (AMSAA), and others in the testing community, have been beating this drum for many years. What is relatively new, however, is the growing realization within the training systems acquisition community (and among training systems users) of the limitations of the current P_K approach and the possibility of employing a higher fidelity and more dynamic scheme for V/L and damage assessments.

DEGRADED STATES VULNERABILITY METHODOLOGY

Between 1988 and 1990, the Ballistic Research Laboratory (BRL – now the Army Research Laboratory ARL) and AMSAA developed a methodology for improved vulnerability assessments in ground combat vehicles, as a replacement for the P_K method; they referred to this as *degraded states vulnerability analysis*. [2]

We are aware of two projects that have attempted to deal directly with the limitations of the traditional P_K approach to vulnerability mapping through application of a degraded states approach. The earlier of the two was the *DSWARS: Degraded States Weapons Analysis Research Simulation* project undertaken in the 1988-90 time frame by Gary Comstock at AMSAA in conjunction with BRL. [3] The second project, the *CASTFOREM Degraded States Project*, was performed in 1994, principally by David Durda and Joe Aguilar at the U.S. Army TRADOC Analysis Center (TRAC WSMR), also in coordination with the ARL Survivability/Lethality Analysis Directorate (ARL/SLAD). [4] The following paragraphs provide an overview of these projects, as reported by the study teams, and the Degraded States Vulnerability Methodology (DSVM).

DSWARS. Using an AMSAA model called GROUNDWARS, Comstock replaced the model's vulnerability calculations (the traditional P_K approach, employing Standard Damage Assessment Lists (SDAL(s)), with a degraded states model developed as an extension of GROUNDWARS. In this DS approach, the functions of a tank were considered as six subsystems: mobility, firepower, acquisition, crew, communications, and ammunition. A list of degraded states and their subsystems was developed jointly by BRL and AMSAA. (see column two of Table 1). The states that indicate a change in performance, rather than complete loss of capability, are intended to be user-defined.

Typical target-weapon pairing conditions are:

- Two *exposures* – fully exposed or hull defilade
- Ten *round dispersion* values (one to ten feet)
- Seven *view* or *aspect angles* of round-to-target (0 to 180 degrees, in 30 degree increments)
- Six engagement ranges (500 to 3,000 meters, in 500 meter increments)

Table 1. Degraded States Options

Subsystem	DSWARS Degraded States	CASTFOREM Degraded States
Mobility	M_0 – No mobility damage M_1 – Reduced maximum speed (slight) M_2 – Reduced maximum speed (significant) M_3 – Stop after time t M_4 – Total immobilization M_5 – M_1 and M_3 M_6 – M_2 and M_3	M_0 – No mobility capability loss M_1 – Driver's vision blocks M_2 – Commander's vision blocks (3,4,5 reduced visibility) M_3 – 30% reduction in speed M_4 – Driver's middle vision block M_5 – 60% reduction in speed (road-wheels/hub) M_6 – Driver's controls M_7 – Driver's service and parking brake M_8 – Total immobilization M_9 – Commander's vision blocks (ALL no visibility)
Crew	C_0 – 0 crew casualties C_1 – 1 crew casualty C_2 – 2 crew casualties C_3 – 3 crew casualties C_4 – 4 crew casualties	C_0 – No crew loss C_1 – Driver C_2 – Commander C_3 – Loader C_4 – Gunner P_0 – No personnel loss
Communications	X_0 – No communications damage X_1 – No internal communications X_2 – No external communications > 300 ft X_3 – No external communications X_4 – X_1 and X_2 X_5 – X_1 and X_3	X_0 – No communications loss X_1 – No internal communications X_2 – No external communications > 300 ft X_3 – No external communications
Target Acquisition	A_0 – No acquisition damage A_1 – Reduced acquisition capability A_2 – Unable to acquire while moving A_3 – A_1 and A_2	A_0 – No acquisition capability loss A_1 – Driver's middle vision block A_2 – Commander's vision blocks AND GPS A_3 – Gunner's primary day AND TIS A_4 – Unable to slew A_5 – Unable to acquire while moving A_6 – Gunner's primary day AND TIS AND auxiliary sights A_7 – Commander's vision block AND weapon
Firepower	F_0 – No firepower damage F_1 – Loss of main armament F_2 – Unable to fire on the move F_3 – Increased time to fire F_4 – Reduced delivery accuracy F_5 – Loss of secondary armament F_6 – F_2 and F_3 F_7 – F_2 and F_4 F_8 – F_3 and F_4 F_9 – F_2 and F_3 and F_4 F_{10} – F_2 and F_5 F_{11} – F_3 and F_5 F_{12} – F_4 and F_5 F_{13} – F_2 and F_3 and F_4 and F_5 F_{14} – F_2 and F_3 and F_5 F_{15} – F_2 and F_4 and F_5 F_{16} – F_3 and F_4 and F_5 F_{17} – F_1 and F_5 (total loss of firepower)	F_0 – No firepower capability loss F_1 – Main armament F_2 – Coaxial weapon or ammunition F_3 – Loader's weapon or ammunition F_4 – Commander's weapon or ammunition F_5 – Gunner's primary sight (DAY) F_6 – Gunner's primary (TIS) F_7 – Gunner's auxiliary sight F_8 – Power traverse OR elevation F_9 – Emergency traverse OR elevation F_{10} – Target range F_{11} – Hydraulic cable to open/close ammo door F_{12} – Reduced delivery accuracy F_{13} – Unable to fire on the move
Ammunition	K_0 – No ammo lost K_1 – Bustle ammo lost K_2 – Hull ammo lost K_3 – K_1 and K_2 K_4 – K kill	
Catastrophic		K_0 – No catastrophic loss K_1 – Bustle ammo lost – no K-kill K_2 – Hull ammo lost – no K-kill K_{10} – True catastrophic kill

Because the total possible number of combinations of these degraded states (75,600) and vulnerability conditions is over 63.5 million, Comstock reduced the number of round dispersions to five (1, 2, 3, 5, and 10) and view angles to four (0, 30, 60, and 90 degrees). This brought the total number of possible combinations down to about 18 million.

For DSWARS, BRL developed a set of degraded states vulnerability files, using the SQuASH (Stochastic Quantitative Analysis of System Hierarchies) model; including degraded states probabilities for a specified target/weapon pairing using the conditions indicated above. All degraded states probabilities for a given range, target exposure, dispersion, and view angle are added to form a cumulative probability distribution that adds to a value of one. A random draw from a uniform distribution (0,1) is used to determine and assign a provisional degraded state to the target. The target's degraded state is described fully by the DS values assigned to each subsystem. As the target receives damage from an initial hit, its assigned DS values will reflect a loss of capability that is likely to have resulted from that damage. (Subsystem and component *fault trees* were developed by BRL for each of the DS "kill" definitions, with components listed in either series or parallel, indicating which components must be lost in order to "cut the tree" – i.e., interrupt the fault tree path.)

After the initial degraded state has been assessed and assigned to the target, subsequent hits on target also are considered, using the same scheme and by comparing the resulting "new" DS for each subsystem against the currently-assigned DS. A more severe assessed degradation results in a new DS, while a less severe degradation results in no change. New types of damage are added to the current DS and new combinations of damage also are added to result in the more severe set of DS values being assigned.

Because Comstock used a constructive simulation (GROUNDWARS) as the basis for DSWARS, he also had to map each degraded state loss of capabilities into some level of mission degradation. For example, the DS M_2: Reduced Speed (significant) was programmed to mean the following, in terms of performance and tactics:

- Degrade max speed to 30 percent (1.67 mps)
- Stretch line-of-sight times already scheduled
- If unit can still move and fire, cancel any current engagements, cancel any already scheduled detections, and schedule overwatch

Comstock ran a series of scenarios to compare the DSWARS approach to the SDAL approach. In general, his assessment was that a degraded states approach seemed to provide a "fuller and more detailed picture of combat damage assessment." He also noted that complete losses were reduced in the DS cases, but increased partial losses tended to more than offset the total corresponding SDAL losses. Thus, overall, scenario outcomes tended to be correspondent.

CASTFOREM. The Combined Arms and Support Task Force Model (CASTFOREM) is a high resolution, low echelon, combined arms combat simulation model that can be used in air defense, engineering, dismounted infantry, and combined arms conflicts. CASTFOREM is used to evaluate weapon systems and tactics in brigade and below combined arms conflicts. [5]

In 1994, TRAC WSMR performed a CASTFOREM Degraded States Project as a proof-of-principle demonstration of the use of degraded states vulnerability data within CASTFOREM. Similar to DSWARS, this was an effort to use the Degraded States Vulnerability Methodology (DSVM) that had been developed by ARL as a possible replacement for the SDAL approach.

TRAC and ARL developed a set of states (actually, the list includes both capability states and some component damage - see column three of Table 1). As in the DSWARS project, degraded states data from ARL was limited to the M1 tank. TRAC developed the modifications needed to integrate CASTFOREM with the ARL-provided code and also modified CASTFOREM to implement target responses to the degraded states. CASTFOREM also includes a decision table feature that permits modification of responses to degraded states data, thereby allowing users to define states differently for other targets, and this feature was explored during the CASTFOREM DS Project.

TRAC conducted this study using existing scenarios from the 1993 M1A2 Initial Operational Test and Evaluation (IOT&E) and compared the DSVM outcomes to SDAL outcomes (one SDAL data file was used, representing a KE round that is an overmatch to the tank).

Despite some significant acknowledged limitations in this study, the CASTFOREM DS Project team findings included: [6]

- Large scale, high resolution models can be modified to use DSVM data and a decision table based model like CASTFOREM facilitates incorporation of responses to various degraded states assessments.

- It appears that scenario behavior is similar, but that units seem to live longer and fire more ammunition in DS vs SDAL.

- There were fewer K-kills with DS data and the defender seemed to benefit from DS data in the scenarios employed.

- Many questions remain to be answered and further analysis and comparisons with SDAL-based data are needed before the costs and benefits of DSVM can be adequately assessed. They reported: "If we had to vote today on the adoption of Degraded States as the way to go, the answer would be 'No' – not because we don't believe that it is the right way to go, but because there are still many questions that we need to have answered." Some of these questions will be discussed later in this paper.

TRAINING SYSTEMS DEGRADED STATES

We believe that the DSWARS and CASTFOREM DSVM projects have helped point the way toward one possible solution for development of a higher level of damage assessment capability in direct fire training simulations. Our efforts have been directed toward development of a way to develop and demonstrate a DSVM in a training simulation. Principally, this has meant: 1) design and development of a stand-alone, PC-based prototype damage assessment tool; and 2) development of a proof-of-principle demonstration of DSVM integrated (or interfaced) with a training simulation.

Our interest has been in finding an appropriate training simulation with which to interface. We believe that the DSVM should work across domains, since one of the goals of VLSE has been to improve standardization of V/L assessments. We chose to work first with ModSAF, since the SAF environment provides a measure of standardization within the constructive domain and the ModSAF software is readily available. A suitable virtual or live simulation platform, however, has been more problematic due to software and documentation restrictions and our interest in not interfering with ongoing programs. In working with NAWC TSD, we have been fortunate to have developed a working partnership with STRICOM, the Army National Guard, and their contractor, Raydon, Inc., who have agreed to make available to us a version of the A-FIST trainer, which is undergoing a major upgrade at this time.

Vehicle Vulnerability/Crew Casualty Calculator. Our first step was to develop a prototype tool for visualizing the results of a target-weapon pairing and for comparing "damage" assessed by the SDAL-based (P_K) approach with a DS approach (modeled after DSWARS degraded states). We have referred to this prototype device as a Vehicle Vulnerability/Crew Casualty Calculator (V^2C^3). Top-level requirements for V^2C^3 included:

- PC-based vulnerability calculations
- Object-oriented/modular architecture
- Graphical user interface
- Visible damage assessment
- Multiple vulnerability methodologies
- 3D model visualization and manipulation

The prototype system uses Silicon Graphics' Open GL 3D application programming interface for creating and manipulating the target vehicle models. Visual C++ is used to create the GUI interface to run on the PC platform. Minimum system specifications include Pentium 120, 64MB, Open GL capable video card, Windows 95, 98, NT. The system also uses the compartment vulnerability model data obtained from ModSAF and the SQuASH model data obtained from DSWARS and CASTFOREM degraded states studies.

Table 2. V^2C^3 System 3D Functionality and Inputs

3D Functionality	System Inputs
Rotate	Range
Zoom	Exposure
Reset view	Angle
Reset damage	Dispersion
Persistent damage	Ammo type
Persistent shot lines	Target vehicle

The prototype tool uses the exact compartment scheme found in ModSAF and in CCTT-SAF for derivation of P_K *probabilities of kill* and also incorporates a DSWARS degraded states methodology for vulnerability assessments, as a means of visualizing the two approaches. In the course of adapting ModSAF vulnerability software for V^2C^3, we identified some errors in the ModSAF software involving incorrect weapon/target lookups.

The degraded states implementation used in DSWARS was chosen because it has well-documented source code and published results. Since the original DSWARS code was written in the C programming language, the degraded states calculation routines were easily integrated into V^2C^3. The functions necessary to read in the data files and assess damage were placed in a module that is accessed by V^2C^3. Inputs obtained from the user interface are passed to the degraded states module where the damage assessment calculations are performed. The module returns the resulting state to be displayed.

The DSWARS degraded states functions accept the same inputs as the compartment model Individual Unit Action (IUA) routines, with a reduced number of inputs for aspect angle and dispersion, as noted earlier. An advantage of using lookup tables rather than real target geometry and physics is the speed of calculation. Aspect angle, range, dispersion, and exposure inputs are used to reference the appropriate table. A random number is generated using a uniform distribution and the new state is determined. Transition tables for each subsystem are used to combine the old state with the new state and the result is returned to the system.

V^2C^3 displays the degraded states results as a 3D representation of a tank and in a text list. A 3D M1A1 tank model was created with components added to represent the various degraded states. By default, undamaged components are colored green. When a degraded state occurs, the corresponding component's color is changed to yellow for a damaged component and red for destroyed components – as a visual representation of the degraded state. The model can be made transparent to display the internal components. The results also are printed to a text list, which displays the state of all the subsystems and also a list of components. The components were derived from the fault trees developed by BRL when they formulated the degraded states for DSWARS.

We also are working to develop an update to the V^2C^3 system, incorporating the CASTFOREM Degraded States Vulnerability Model. Software has been obtained from TRAC WSMR and we are working with them to leverage off of their previous DS efforts as much as possible. We will port the necessary routines and files to a Microsoft Windows environment, transfer the code to the Visual C++ programming language, and reuse as much of the CASTFOREM DSVM software as possible.

As noted earlier, the degraded states data provided for the CASTFOREM DS study consists of one data file which represents a penetrating tank KE round versus an M1 tank. The data consists of sets of state data for 4-inch square cells overlaid on the silhouette of the M1 tank target. Each square has a data table containing the state vectors created from the results of 10 runs on ARL's S2 vulnerability model. The sum of the probabilities for the state vectors for each cell is equal to one. The state data consists of an 80-character vector, in which each character represents one state. The characters in the vector are set to a "0" or "1" to indicate the absence or presence of the corresponding state (see Figure 1). [7]

The "cell resolution" data that was used in the CASTFOREM DS study provides a far more accurate damage assessment than the data used in DSWARS. The DSWARS study used a "view averaged" data set. While the cell resolution data maintains the independence of each 4 inch square on a tank, view average data takes all of the data cells for each aspect angle and dispersion and averages the results. The averaging of data stops the level of accuracy at a single dispersion value that represents the distance from the centerline of a target, but not the direction. With the cell resolution data, both the distance from the center point and the direction are known, therefore the exact location of the hit can be pinpointed. A *view averaged* assessment of a hit with a dispersion value of 4 cannot distinguish if the hit is on the turret or on a road wheel.

A-FIST Degraded States Application. The proof-of-principle integration of the DSVM with a training simulation involves interfacing the enhanced V^2C^3 prototype damage assessment tool with the A-FIST trainer.

A-FIST is a tank-appended device that provides full-crew interactive gunnery and tactics procedural training on the Abrams M1 series tanks. During training, the tank is in a non-operational power-off mode, with required power provided separately to the tank controls, indicators, and appended components through a power distribution network contained in the A-FIST instructor/operator station.

Tank-appended monitors provide a visual scene and each member of the crew views the terrain and targets through his own closed-hatch position. In general, A-FIST exercise scenarios are pre-scripted and have been derived from the Army's tank combat tables.

In this planned initial integration with A-FIST, we will be receiving information from A-FIST, but not providing any information back to the trainer.

Simulation packets containing weapon impact parameters will be transmitted and received via a

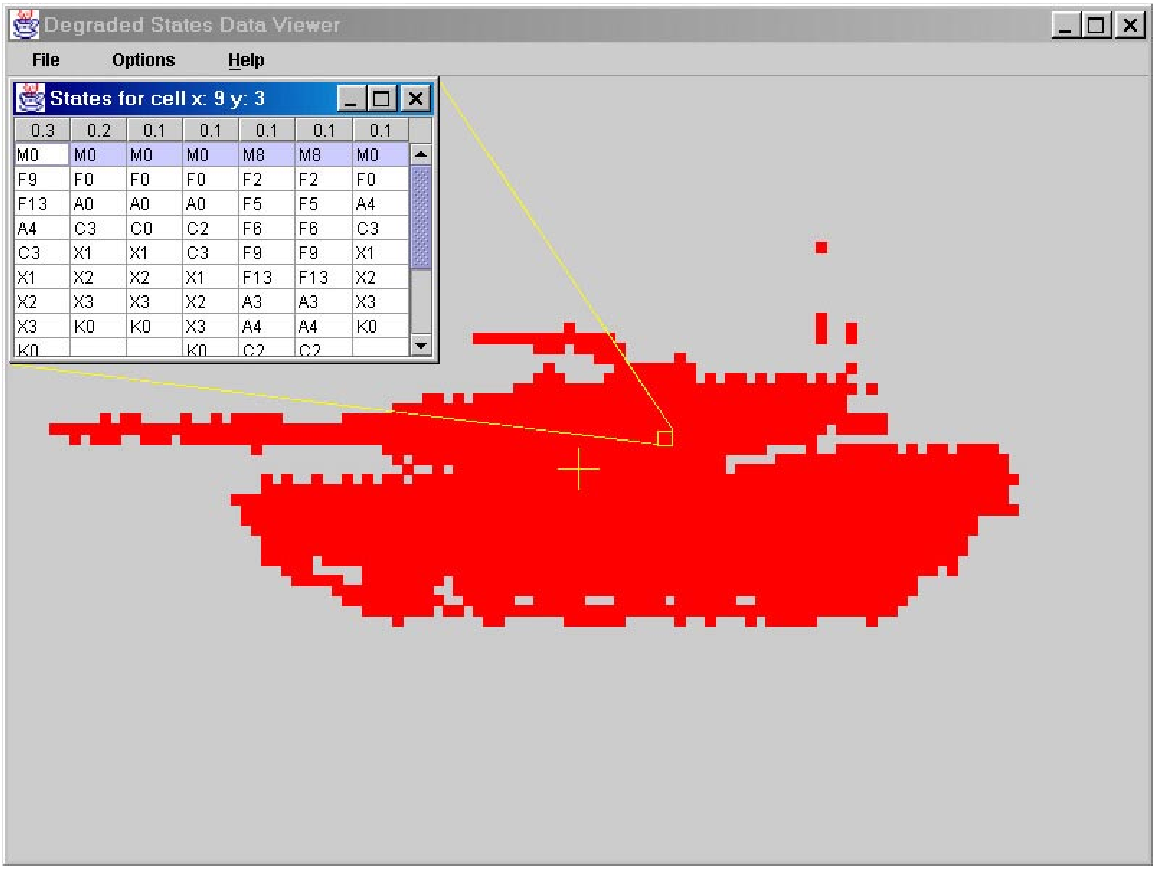

Figure 1. CASTFOREM Degraded States Data Viewer

network connection. The data to be transmitted will be formatted as either a standard DIS packet or in a custom packet format.

The four main procedures that will implement all of the processing are: 1) Listener; 2) Controller; 3) Send_DS_PDU; and 4) Visual Display. Each of these procedures contains various functions and makes procedure calls to process data from the A-FIST trainer and display the outputs via V^2C^3. The Listener will establish a direct network connection with the A-FIST server and receive the desired information packets. Our processor also will establish a direct connection with the A-FIST server and will request the desired information packets directly.

The information packets contain "detonation" and "entity-state" Protocol Data Units (PDUs). The detonation PDU(s) is used to run the V/L analysis on the hit. The entity-state PDU(s) is used to compare the results obtained by A-FIST and the DS result. The captured PDU(s) are sent to the PDU Queue, where it will wait until the Controller procedure retrieves it for further processing. The Controller uses parameters entered by the user to determine what actions are performed on the PDU(s) obtained from the PDU Queue. Next, the Controller checks the queue and extracts the PDU at the top. It then determines the type of PDU and performs the appropriate action. Entity-state PDU(s) are sent to the Logger and Detonation PDU(s) are processed by the V/L methodology specified by the user. The result is either sent to the Logger or to the Send_DS_PDU procedure. When a V/L result is sent to the Send_DS_PDU procedure, it is formatted for transmission over the network, and then sent out. The Logger stores the entity states and V/L calculations. The data stored inside the PDU is accessed through a visual interface built using Microsoft Foundation Classes (MFC). The data will be displayed in graphical and textual formats with various methods to sort and query the data.

SOME DS TECHNICAL ISSUES

Cumulative Damage/Multiple Hits. A major limitation of vulnerability data stored in table format is the lack of cumulative damage algorithms.

This limitation applies to all V/L methodologies being used in simulations today, including the DS data used in the CASTFOREM study.

The terms *cumulative damage* and *multiple hits* refers to the change in target geometry due to prior damage; not to the accumulation of damage vectors. Current simulations assess every hit on a target as if that target were undamaged. The inaccuracies of this method should be obvious. Clearly, a hit on an area of a tank that was left armorless by a previous hit would be far more effective than a hit on a fully armored tank. Since degraded states are not independent, some combinations of individual assessments may imply the wrong state. For example, an initial assessment of a particular component in a parallel path of a fault tree appears to be damaged. However, the function that sets the state of the target does not get affected since the parallel component is not damaged. Each time the target gets damaged, the current methodologies react as if the target were a new object with all of its components intact. Therefore, if the second component on a parallel fault tree is damaged, but the first is not, incorrect states will be determined. In essence, there isn't any bookkeeping performed on the individual components from one assessment to the next; resulting in a second assessment producing a state vector where the function is not affected. The lack of cumulative damage algorithms means that units tend to live longer and more ammunition is expended. The only way around this problem is to introduce a component level/cumulative damage model. Currently, the SQuASH S2 vulnerability model, developed by ARL/SLAD is the only identified solution to this problem.

It should be reiterated that, in current training simulations, the effect of multiple hits is not considered, and a second or third hit on target is assessed as if it were the first hit on an undamaged target. The current method of accounting for damage from multiple hits is to perform a logical [AND] of the old state with the new damage state. This method keeps the highest severity damage state, but this state will not affect the analysis of the next hit. All methods that use lookup tables, including those used in the DSWARS and CASTFOREM studies, are limited to this and currently, we are following this scheme, as well.

Target Artificial Intelligence (AI). One of the results of the CASTFOREM DS study was the realization that a fairly extensive rewrite of CASTFOREM software would be required to accommodate the higher level of detail available from degraded states damage assessments, especially in terms of decision rules affecting target behaviors. This is a problem with constructive simulations. In V/L terminology, there needs to be a better mapping of assessed damage (degraded performance) into mission outcomes (mission utility). [8]

The constructive simulation domain has several difficulties for incorporating a degraded states methodology. Changing a unit's movement speed and rate of fire is relatively straightforward to accomplish. The difficulty comes in the unit behavior. If a tank can only move at half its normal rate, should it complete its mission to the best of its ability or should it retreat for repairs? With the old M/F/K states, the number of behavioral combinations is easy to map out. Expanding the number of damage states to 50, however, creates many more situations that must be considered. A computer AI must now determine if the loss of the gunner is serious enough to stop an assault. What strategy is going to be employed if a tank loses the ability to traverse the turret in the middle of a firefight? The sheer number of possibilities makes this a daunting problem. In the CASTFOREM study, this was dealt with by mapping the degraded states to the SDAL metrics. For example, loss of the main gun and the ammunition for all secondary weapons would result in an F-kill. From there, the old behaviors were used. This rolling-up of the data created some difficulties in assessing results of scenarios that were run.

Of course, in virtual and live simulations, where there is a crew-in-the-loop, crew responses and interactions with the vehicle and scenario should provide for the "mapping" from degraded performance to mission outcomes. A virtual simulation, therefore, should be a preferred application for evaluating the effectiveness of a degraded states methodology. (The issue there will be the best way to present casualty and damage/degraded states information to the crew such that it does not introduce factors of negative training.)

A-FIST Targets. Our DS application in A-FIST represents yet another challenge because A-FIST is a gunnery procedures trainer. Targets in A-FIST move along a predetermined path and do not shoot back at the trainees. The training emphasis is on gunnery procedures using standard gunnery exercises and on minimizing own vehicle exposure. There is no mechanism currently in A-FIST to significantly affect target capabilities without major changes to the trainer. As with a constructive simulation, it would take a major rewrite of the A-FIST code and also a change to the system architecture to incorporate DS damage assessments.

A-FIST represents a convenient starting point for an exploration of a DS methodology in training simulations. The A-FIST tank gunnery trainer uses a similar method for determining how a tank is damaged. Several "hit plates" are drawn on the tank and hits to certain plates will destroy a target. Specified gunnery doctrine manuals from the National Guard determined the plates that will destroy a tank when hit. Shot data from A-FIST can be projected onto the CASTFOREM data to determine an accurate degraded state. This would allow for better analysis of hits on a tank.

Following the initial DS A-FIST development and integration, subsequent phases of this project include plans to provide DS feedback (i.e., reduced capabilities) to the manned modules and to explore applications of DS in an A-FIST tank-on-tank scenario or mode of operation and in other compatible National Guard gunnery trainers. We also plan to investigate ways of introducing some Semi-Automated Forces (SAF) type decision rules into A-FIST for target maneuvering after it has received damage.

Future Directions. As noted earlier, we believe that a degraded states vulnerability methodology has significant potential for providing a higher level of vehicle damage and crew casualty assessments. Our project is focused on demonstrating the efficacy of such an approach and exploring related issues. We are working with limited data and degraded states definitions developed for the two previous DS initiatives. While this will work for our near-term objectives, it will be important to develop a standardized set of degraded states definitions in order to more fully develop and evaluate this concept. AMSAA and ARL/SLAD are the originators of the DS methodology. They developed and have maintained the relevant supporting data and models. As a result, we believe that AMSAA and ARL/SLAD should be tasked and funded to further explore the development of DS – especially as it relates to training systems applications and issues, as well as to test models and simulations.

In summary, DS issues to be explored and which should be further developed include:

- Development of a standardized, and perhaps expanded, set of DS definitions for a wider variety of ground combat vehicles; and implementation of those degraded states in a virtual trainer. Standardization also should permit some level of user modification.

- Development of methods for better consideration of accumulating damage resulting from multiple hits.

- Development and application of decision rules to support the implementation and evaluation of DS in constructive simulations (and in those virtual/live simulations which could benefit from more intelligent or interactive targets) in order to map degraded states to mission utility.

- Development of methods for providing information to the crew regarding assessed damage and crew casualties, and for providing ways for the crew to respond, including battle damage repair and crew work-arounds. This also includes development of approaches for more realistically simulating both primary and secondary damage effects.

- Development and demonstration of a DS approach to V/L in other existing virtual simulations, and further evaluation of its utility.

REFERENCES

1. Dr. Paul H. Deitz and Dr. Michael W. Starks, *The Generation, Use, and Misuse of "P_{K} s" in Vulnerability/Lethality Analyses*, U.S. Army Research Laboratory, Survivability/Lethality Analysis Directorate. Expanded from a paper presented at the 8th Annual TARDEC Symposium, held at the Naval Postgraduate School, Monterey, CA, March 25-17, 1997.

2. John M. Abell, Mark D. Burdeshaw, and Bruce A. Rickter, *Degraded States Vulnerability Analysis: Phase II*, U.S. Army Ballistic Research Laboratory, BRL-TR-3161, October 1990.

3. Gary Russell Comstock, *DSWARS: Degraded States Weapons Analysis Research Simulation*, U.S. Army Materiel Systems Analysis Activity, Aberdeen Proving Ground, MD, August 22, 1990.

4. Joe L. Aguilar and David Durda, *CASTFOREM Degraded States Proof-of-Principle Project Summary Report Input (Unpublished)*, U.S. Army TRADOC Analysis Center – WSMR, White Sands Missile Range, NM, June 1995.

5. Material Systems Analysis Tools: Models and Simulations, web page http://amsaa-www.arl.mil/core/mat_tool.html.

6. Doug Mackey, *CASTFOREM Vulnerability Representation Workshop*, U.S. TRADOC Analysis Center – WSMR, White Sands Missile Range, NM, Presentation at the Vulnerability Representation Workshop, held at NAWC TSD/ STRICOM in Orlando on April 14, 1999.

7. David Durda, *CASTFOREM Degraded States Data Viewer Software,* U.S. Army TRADOC Analysis Center – WSMR, White Sands missile Range, NM, 1999.

8. Paul H. Deitz, *A V/L Taxonomy for Analyzing Ballistic Live-Fire Events, Proceedings of the 46th Annual Bomb & Warhead Technical Symposium*, 13-15 May 1996, Monterey, CA also appears in Modeling Ballistic Live-Fire Events Trilogy, U.S. Army Research Laboratory Technical Report ARL-TR-1274, December 1996.