A Methodology for the Generation, Storage, Verification and Validation of Performance Data for Modelling and Simulation

Mr Denis Shine, Dr Oleg Mazonka, Mr Lance Holden, Mr David Hemming and Ms Tanya Frank
23 Labs, DSTO Edinburgh, PO Box 1500, Edinburgh SA 5111
AUSTRALIA

denis.shine@dsto.defence.gov.au

ABSTRACT

Defence Science and Technology Organisation (DSTO) provides scientific advice to the Australian Army to support acquisition, concept development and force structure decisions. Combat Simulations have been an integral part of the multi-disciplinary approach employed by DSTO for the past two decades. DSTO uses multiple Combat Simulations, each of which require similar input data relating to weapons, targets and their interaction on the battlefield. However, the representation and interpretation of this input data differs between simulations, making it difficult to represent systems consistently across simulations and thereby reducing the confidence of studies employing several simulation tools.

Previous attempts to solve this issue have faced recurring problems with the paucity of available empirical data to describe weapons, platforms and their interactions. An entity in a Combat Simulation requires complete data on its interactions (e.g. probability of hitting and killing it) with every other entity under numerous environmental and situational conditions. The combinatorial blowout in interactions between all types of entities, and the number of factors modifying these interactions, means this data cannot be gathered empirically. It must, instead, be estimated through modelling. Such models exist, but these models are themselves so complex that their input data and resource requirements mean that results cannot be calculated on the fly.

DSTO have developed an approach to mitigate this problem which is built around three concepts:

- *A database, called the Simulation Repository (SimR) that contains simple, verifiable, fundamental attributes for systems including weapons, ammunitions, platforms and sensors.*
- *A set of algorithms and techniques based on physics to calculate more complex data required by Combat Simulations from the simple formats stored in SimR.*
- *The use of available empirical data as a verification tool at all stages of the data generation process, but not necessarily as a direct input to SimR*

The data generation algorithms allow for the rapid generation of rough order of magnitude performance data - sufficient to distinguish between classes of systems --which is an appropriate level of detail for our purposes. This reduces the data storage burden on SimR, which in turn minimises the amount of verification and validation required.

Our approach allows us to generate and manage the input data for multiple Combat Simulations with a relatively small workforce. We can also respond to changes in data requirements, such as the addition of platforms or weapons to an Order of Battle (ORBAT), within limited timeframes. Most importantly, our approach allows us to use multiple Combat Simulations to analyse a single problem with the assurance that the input data is drawn from the same source.

1.0 INTRODUCTION

Combat simulations are an integral part of the suite of Operations Research (OR) tools and techniques that Joint and Operations Analysis Division (JOAD) uses to analyse land force effectiveness problems. Our approach is centralised around building multi-disciplinary teams, which use multiple interconnected tools and techniques, including combat simulations, to analyse a problem. It seeks to balance Internal Validity (the ability to identify cause-and-effect within a problem solution) and External Validity (the ability to relate the results of an analytical campaign to the real world) [1]. This approach is in accordance with the principles outlined in the GUIDEx [2]. We use several internationally accepted combat simulations, such as the Close Action Environment (CAEn, UK), One Semi-Automated Forces (OneSAF, USA), the Combined Arms Analysis Tool for the 21st Century (COMBAT XXI, USA) and Map Aware Non-Uniform Automata (MANA, NZ).

In order to properly adjudicate combat outcomes, these simulations require massive amounts of input data, describing effects such as weapon trajectories, platform mobility or vulnerability to direct and indirect fire. This input data is represented as large lookup tables, which describe the characteristics of a system at defined points. We have faced a longstanding problem of generating consistent, appropriate data to populate these tables.

This paper describes a unique solution we have developed to provide validated, fit-for-purpose input data for combat simulations. The generation and management of such input data is vitally important to ensure that combat systems are modelled appropriately and valid analytical insights can be obtained. We are cognisant of a number of challenges that we face:

- The data requirements are massive, multi-dimensional, difficult to visualise and difficult to estimate;
- We operate within a work environment characterised by short lead times and shifting requirements;
- We maintain multiple combat simulations, each of which require data; and
- There is a lack of expansive, available empirical data on the subject.

The approach is built around three main concepts. Firstly, we have developed a database called the Simulation Repository (SimR) [3], which aims to store and manage data appropriate for input into combat simulations. Secondly, we have developed a suite of data generation and estimation techniques that are designed to assist SimR in generating the enormous lookup tables required by combat simulations. Finally, we hold both of these tools to a consistent verification and validation methodology, using empirical data to appropriately assess the realism and validity of generated data.

2.0 PROBLEM DEFINITION

This methodology was developed to tackle four major issues we face with our combat simulation capability: massive data requirements; short lead times; multiple combat simulations; and the paucity of empirical data.

2.1 Massive Data Requirements

With models and simulations becoming more complex, the requirement for ever increasing amounts of data increases the probability for human error within the data. For example, a simulation study requires probability of kill data for each of combination of weapon and target. However, simulations require this data for a multitude of situations, such as moving firers and targets, level of defilade of the target, range to target, aspect angle of target and elevation of the firer. For each combination of these factors, the simulation requires the standard set of kill types – Mobility, Firepower, Mobility/Firepower and Catastrophic.

So if a small study contains 10 ammunition types, 15 target types, 4 motion combinations, 2 levels of cover, 1.5 elevations (since some weapons can be fired from rooftops, but not others), 5 ranges, 8 angles and 4 types of kills, it requires 288,000 data points all stored within enormous lookup tables accessed by the simulation. The possibility of an error being introduced into such a dataset is high. In addition, the overhead associated with changing or updating the dataset when additional data is introduced is also a major issue. Users do not always have access to the datasets allowing them to either challenge the values or confirm what is being used. Local subject matter experts may also have an influence as to what data values should include.

2.2 Short Lead Times

Simulation-based experiments are often beset by shifting priorities which flow onto requirements for the modeller. Often, a new weapon or target is required to be added to the simulation and used within a matter of weeks, which is far too short for detailed field trials or intensive performance and vulnerability modelling. Thus our solution needs to provide the ability to produce a reasonable approximation of systems within a rapid timeframe.

2.3 Multiple Combat Simulations

JOAD maintain several combat simulations as part of our multi-methods approach to analysing land warfare. The Land OR branch maintains or uses three major combat simulations – CAEn, COMBAT XXI and OneSAF, along with a suite of more abstract tools such as MANA. Each simulation requires similar input data describing system performance, however, the exact format, parameters and assumptions tend to differ, often to the point that the same data is not directly relatable from one simulation to another.

Data is often shared between simulation managers which presents problems with data provenance and confidence. Data may have been sourced in one form and through internal processes been manipulated into a format that suits another simulation. In doing this the data may have lost some of its validity, the original value and provenance.

2.4 Paucity of Empirical Data

Collecting Real world data requires live fire events using equipment and ammunition. This can be very expensive. Because of the variety of systems available worldwide no one country can possibly have, or have access to every system or platform. Because of the availability of live fire data some organisations have very good data sets on some systems and then interpret the capabilities of other systems. Because of this practice results in some simulations will have a very low confidence rating. The method for collecting live fire data may not be consistent between countries or organisations, introducing the problem of data validity and consistency. As a final problem the environment which data was collected may not be indicative of the environment or problem space being studied.

3.0 OUR APPROACH

Our approach to solve these problems is based on three pillars – The SimR database, a suite of Data Generation tools and techniques, and finally the use of verification and validation techniques at appropriate points. We describe these in brief below – full descriptions are provided in the following sections.

3.1 SimR

Given that we maintain multiple, similar simulations, there is scope to create a centralised database to store common input data. This has been attempted before within the Land domain, with mixed results. Its main failing was attempting to store all simulation data in full, or at least a covering set of this data across simulations, causing insurmountable issues relating to verifiability and sourcing. Our implementation, SimR, stores only a minimalistic dataset required by combat simulations, relying on data generation tools to produce performance data on demand.

3.2 Data Generation Algorithms

Data generation algorithms and techniques have been developed to provide fit-for-purpose input data for our target simulations, allowing SimR to store a smaller dataset and better enabling consistent data to be generated across simulations. For example, all simulations require trajectory data for weapons systems. Instead of storing this information within SimR, we instead only store simple characteristic data such as muzzle velocity and ballistic coefficient, relying on a ballistics model to produce the more expansive information required by the simulation when required.

3.3 Verification & Validation Techniques

Empirical data cannot possibly cover the breadth of information that we require for use in combat simulations, however it is still the highest quality data achievable and should be utilised. We use empirical data to validate our results at appropriate junctures of the data generation process. Since real world data comes in many forms, this allows us to choose the most appropriate way to use it.

4.0 SIMULATION REPOSITORY

The SimR project was developed with the objective to reduce or remove the ambiguity associated with data. Its purpose is to provide quality data to a wide range of simulations upon request. Rather than store large datasets we felt it easier to, where possible, have the datasets generated in near real time. Figure 1 shows a schematic of the SimR capability.

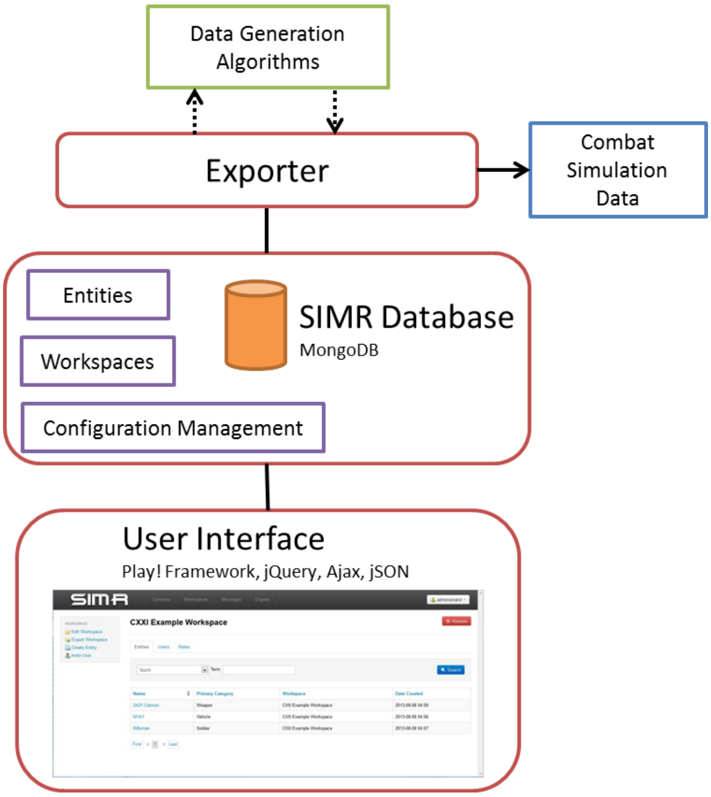


Figure 1 - SimR Structural Layout

4.1 User Interface

The database is accessible through a web interface which also allows users who may be new to the defence environment to browse through differing systems or platforms enabling them to become familiar with them. The user is presented with an easy to view interface with system or platform descriptions, images, performance specifications and capability options. It also permits advanced personnel to locate, view or confirm data fields and values associated with the individual systems or platforms. Descriptions and images are also provided when available as well as information such as performance specifications. Fields have attached metadata regarding its source and the date it was last updated. Finally, SimR is searchable through either a basic or advanced search tool.

4.2 Entities

An entity is a representation of data for an item of interest. This allows entities to represent simulation concepts such as ammunition and weapon up to force organisation and environment. There is no formal structure for what values can be defined for an entity.

A category provides a grouping of related properties for an entity. Each category matches against the common use cases of the simulation tools. Adding categories to an entity allows the simulation exporters to validate that all the required data is available. A category also provides a better name or understanding of properties for a human operator. Thus there is no specific “Tank” entity type – but the tank category type ensures the entity contains the correct information required to describe a tank.

Entities have changes to individual properties tracked. Historical versions of entities can be retrieved at any time. Entities also provide a place to record any providence of the values associated with them. This provides a basis for any challenges against the validity of the entity data.

4.3 Workspaces

The simulation repository groups entities associated with the same study in workspaces. Multiple users can collaborate in a single workspace. Entities can be copied between workspaces; if they are, then users are notified of any changes made to the source entity so they can decide if they want to merge them into their workspaces copy of the entity.

Entities within a workspace can be customised as needed for a specific simulation exporter or to explore different parameters within the experiment space. Customisation can include such things as changing the calibre of a gun, changing the muzzle velocity of a projectile or changing the size or protection levels on a vehicle.

There is a single workspace designated the ‘common’ workspace. This workspace contains a set of managed entities that can form the basis of data used in any other workspace. Users will select an entity they want to use for an experiment and then copy it to their workspace. Once there they can adjust the parameters as needed without affecting any other workspaces copy of that entity. This feature assists in reducing the time it may take in preparing what may be a large study or a series of studies where there is a degree of commonality between systems within each study. The common workspace is managed by a limited group of users authorised to change its contents.

The database has the capability to use previously saved data values from within the central database. If a user wishes to re-run a study within a workspace they are a member of they can select the dates that the database was changed. The user can choose to use previous historical values or re-run an old study using current values. This feature allows previously used studies to be re-used or new studies to use old values.

4.4 Configuration Management

SimR records information about changes to entities within both the central database and workspaces, keeping a record in a similar manner to version control systems. Users are informed of changes to both the central database and workspaces they are subscribed to through the SimR messaging system. This alleviates accountability issues often encountered during simulation management, where changes to entities are made with minimal or non-existent documentation.

4.5 Exporters

When a user has completed building the workplace study, they can use the export function to generate a combat simulation database. The exporter has been setup to package the data into a single compressed file such that the data can be unpacked and placed directly into the simulation. As the export can take a significant amount of time, the user who triggered the export function will be informed by E-Mail when their data can be retrieved. The file remains in the central database both as a record and for future retrieval if necessary. By packaging the simulation data in this manner we can reduce human error associated with building complex, interdependent simulation databases.

An additional feature which can be selected at the time of the export is whether the user wishes to have the exporter generate a Battlebook, which is a detailed listing of the entities within the workspace.

4.6 Interaction with Data Generation Algorithms

The data generation algorithms described in the next section have been designed to work with the SimR exporter. When a user asks SimR to export a database for a particular simulation, this database may need to comprise many weapons and targets, and thus many combinations of the two. Our data generation models by default pair one weapon and one target, requiring multiple independent executions of the model to generate all required combinations. It is thus up to SimR to determine which pairs are required and sequentially call appropriate data generation tools to gather the interaction data. SimR is also responsible for taking the individual results of each pairing and stitching them back together into a working database. Figure 2 shows this interaction visually.

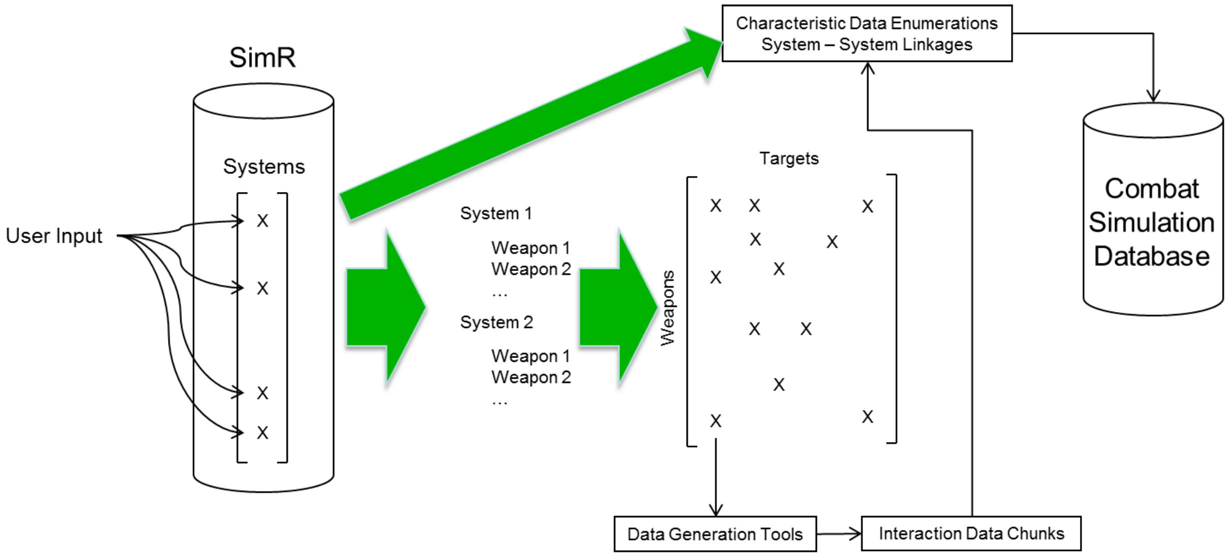


Figure 2 - Interaction Model between SimR and Data Generation Algorithms

5.0 DATA GENERATION ALGORITHMS

SimR is designed as the common storage mechanism for our combat simulations and can thus reduce the data storage burden where data is the same across multiple simulations. For example, each simulation may require knowledge of the top speed of a vehicle, but SimR only needs to store this piece of information once, formatting it as required by the target simulation. This is also true of more detailed data, such as a weapon's ballistic trajectory, however we can take this one more step to even further reduce the data storage burden.

Much of the data required for simulations is technical performance data, such as weapon trajectories and probabilities of hit and kill. In attempting to manage this data, we have encountered some extreme difficulties, such as:

- The data required is extremely intensive, multi-dimensional and very broad;
- Storage of such an expansive set of data is difficult to manage and validate, especially if one requires a generic format;
- Empirical data is very difficult to obtain and in many cases non-existent; and
- Traditional data generation methods lack the responsiveness and flexibility to provide data suitable for use in Army experimentation.

Our approach [4, 5] mitigates these difficulties through a set of rough-order-of-magnitude models. These models are split into three distinct but related components:

- A **Ballistics Model**, which generates trajectories, dispersions and armour penetrations for a wide range of munitions;
- A **Direct Fire Model**, which describes the effects of direct fire munitions on vehicle and infantry targets; and
- An **Indirect Fire Model**, which describes the effects of high explosive munitions on vehicle and infantry targets.

5.1 Data Requirements

The key feature of our data generation models are their limited input data requirements. We are able to produce large volumes of fit-for-purpose performance data from a small set of source data. Data comes in four forms: direct fire munitions, indirect fire munitions, vehicle targets and infantry targets (Table1).

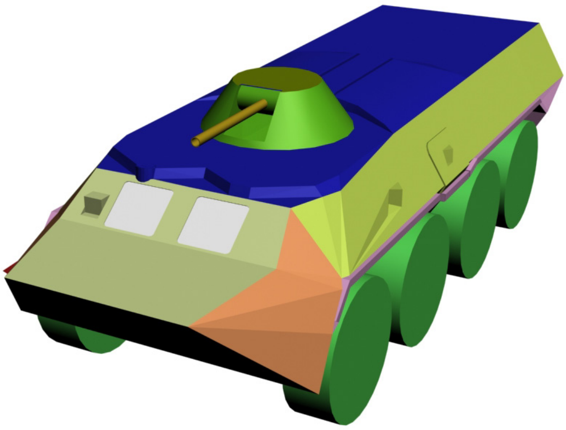
Table 1 - Munition Representation

Direct Fire		Indirect Fire	
Velocity	Muzzle velocity	Charge Mass	Mass of the explosive charge
Ballistic Coefficient	Ballistic coefficient	Case Mass	Mass of the round casing
K-Values	A set of 28 K-Values, as defined in [6].	Munition Diameter	Internal diameter of the casing cylinder
Penetration Equation	One of several simple penetration equations defined in [7]	Case Thickness	Thickness of casing itself
		Mott Constant	Value used for the Mott equation, defined empirically from a defined set of values dependant on explosive and casing
Penetration Parameters	Physical properties of the munition, as required by the equation. Values include mass, diameter, length, material.	Gurney Constant	Initial velocity of fragments, based on Gurney velocities
		Shape Factor	Factor defining the shape of fragments, which is required to calculate penetration
		Alpha	A factor between 0 and 1 defining the "directedness" of the explosive charge. Set to 1 for a totally uniform distribution of fragments, or to near 0 for a directed shaped charge.

Both vehicle and infantry targets are represented as 3D models. The geometry of these models are grouped into parts based on logical groupings, such as parts of the hull, the engine or crew compartments. An infantry model in turn may have body parts represented along with sections of body armour. Each geometry group has metadata assigned to it, which is described in Table 2. This table describes the metadata required for vehicle targets; infantry targets require only a Kinetic Energy (KE) thickness and a probability of incapacitation for each geometry group. On the left is a description of the metadata required for each geometry group, with a visual representation of a target, broken into such groups, on the right.

Table 2 - Target Representation

Data	Description
Name	Name of the geometry part, matching that in the 3D model
Hull/Turret	A descriptor whether this geometry is part of the Hull of Turret of the vehicle (or neither)
Thickness	RHAe thickness of the component, for both KE and Chemical Energy (CE) weapons
Probability of Kill	Probability of Mobility, Firepower, Mobility/Firepower and Catastrophic Kill if this component is penetrated
ERA	A Boolean value representing if this component is protected by Explosive Reactive Armour (ERA)



This set of input data requirements is simple and obtainable in comparison to those for more detailed models and lowers the barrier to introducing a weapon or target into a combat simulation. We are therefore able to build simulation studies using munition and target data of limited detail, introducing more fidelity to input data as required by the study.

5.2 Ballistics Model

Our ballistics model is based entirely on a model developed by the Centre for Operational Research and Analysis (CORA), which developed models of dispersion [6] and ballistics & penetration [7]. These models were developed to produce such data for the Joint Conflict and Tactical Simulation (JCATS) wargame, which is used in Canadian Army experimentation. We have since rewritten and repurposed them, such that they provide data useful for CAEn, OneSAF and COMBAT XXI, which are all in use by DSTO or the Australian Army.

Ballistic trajectories are described with a relatively simple 3 degree-of-freedom model, with a polynomial approximation of the drag coefficient. Many factors, such as yaw, wind or the coriolis effect are ignored or factored out. Penetration is dependent on the ballistics model and is described through a small set of equations that accurately describe various classes of munitions. All penetrations are simplified to equivalents of Rolled Homogenous Armour (RHA)[1], which further simplifies modelling. Dispersion is described through a model that requires 28 coefficients called K-Values. While this seems to be a large amount of data, the paper describing the model provides suggested values for various types of weapons systems, depending on characteristics such as stabilisation and sighting systems.

Figure 3 shows the dependencies between the sub-models. The resulting data is used for two purposes. Firstly, ballistics and dispersion data is used by combat simulations directly, as shown in the black boxes. Secondly, our other two models require data from the ballistics model in order to make probability of kill calculations.

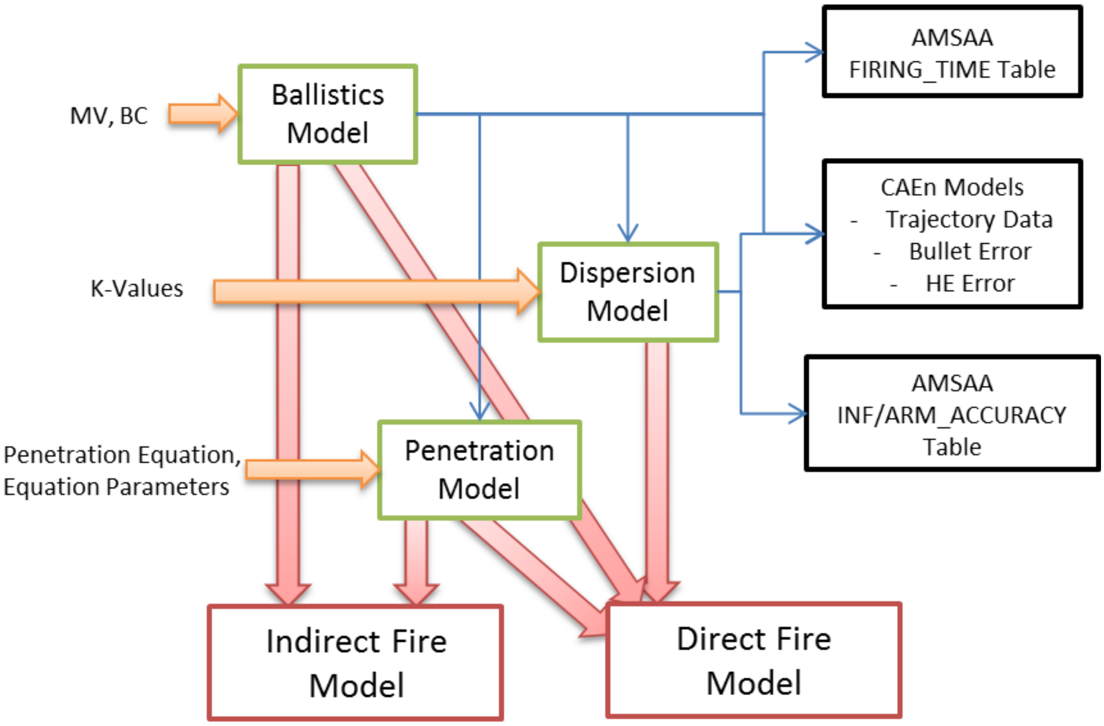


Figure 3 - Ballistics Model Flow and Dependencies

---

[1] RHA was a standard type of armour used on armoured vehicles. It has been rendered obsolete by newer types of armour, but is still used as a benchmark when measuring armour protection and projectile penetration.

5.3 Direct Fire Model

Our direct fire model draws some of its structure from CORA [8], however we have also drawn lessons internally and from other parts of DSTO. All combat simulations need information on the probability of hit and kill for various pairs of weapons and targets. While our model caters for all types of targets, in this section we will concentrate on the more detailed vehicle target model.

Direct fire models have to deal with a number of dependent issues:

- Calculating the probability that a round hits a target, based on the dispersion of the weapon and the presented size of the target;
- Calculating the probability of kill, given a hit, based on the location of the hit point;
- Calculating the above probabilities given a myriad of situational factors, such as the range to target, aspect and elevation angle, relative motion of target and firer and any obstacles the target might be obscured by; and
- Calculating an appropriate weighted average probabilistic outcome, given that we must provide a lookup table of probabilities, not adjudicate individual cases.

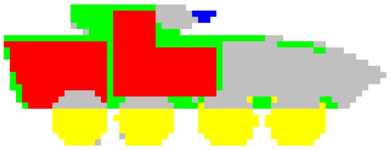
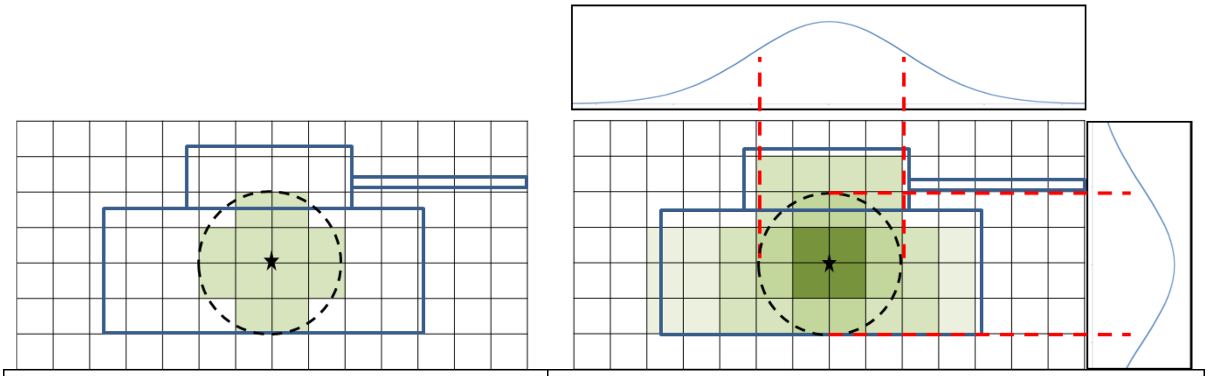


Figure 4 - Kill grid showing different types of kill as different cell colours

Our direct fire model is based on the concept of a kill grid (Figure 4), which is a grid of equal-sized cells overlaid on a target from a given range, aspect angle and elevation. A simulated shot is fired through each cell, the result of which is a probability of kill. These simulated shots are built using the ballistics model described previously, which inform the direct fire model of the angle of fall and penetrative power of the munition. This power is compared with the protection of the vehicle and its various components, which is then translated into the damage this shot would cause to the vehicle.

This method allows us to reach a point where we have an array of kill probabilities across the vehicle. The other component of the direct fire model are a series of results processors, which are modules that translate this raw data into formats appropriate to specific simulations. Each simulation requires that this data be formatted and interpreted differently, requiring a distinct processing method. For example, CAEn adjudicates vehicle vulnerability using a single Shot Kill Probability (SSKP) file, which contains a set of probabilities of kill, given a shot. To produce this data, one has to combine the probability of hit and kill into a single value. Since dispersion is represented as a bivariate normal distribution, exporting this data to CAEn requires that this distribution be overlaid onto the kill grid, providing a weight to each cell.

OneSAF and COMBAT XXI both use the Army Materiel Systems Analysis Activity (AMSAA) Individual Unit of Action (IUA) file format, yet despite this they actually require slightly different data. OneSAF requires that we specify a probability of kill, given a hit, for a series of set dispersion values from 1 to 10 feet and therefore requires a similar method to the CAEn results processor, although with any misses removed. COMBAT XXI requires that we specify the probability of kill for a series of concentric rings around the centre of the target, ranging from 1 to 10 feet. Figure 5 displays visually how OneSAF and COMBAT XXI adjudicate probabilities of kill and the subsequent impact on our data generation methods.



COMBAT XXI	OneSAF
When Adjudicating a Kill: Computes location of hit and selects data from appropriate dispersion. **Data Generation Impact:** Compute uniform average result for points within each concentric range ring.	**When Adjudicating a Kill:** Takes weapon dispersion and samples from nearest IUA dispersion value. **Data Generation Impact:** Integrate a bivariate normal distribution over target with a standard deviation matching each dispersion value.

Figure 5 - Interpretation of AMSAA IUA Data by COMBAT XXI and OneSAF

5.4 Indirect Fire Model

Our indirect fire model takes into account three sources of damage when calculating probabilities of kill:

- Pressure – air pressure caused by the blast damages internal organs;
- Translation – kinetic energy is transferred to the target, throwing them into a hard object (such as the ground); and
- Fragmentation – Pieces of shell casing are propelled and penetrate the target, damaging it.

Pressure and translation are assessed only for infantry targets. Relatively simple physics can be used to assess the air pressure of a round and also its translational effects on a human target. More difficult is deciding exactly how much pressure or gained momentum begins to cause lethal effects. We used empirical data [9], along with several assumptions to develop a probability of kill curve based on the properties of the explosive and range to target.

Fragmentation was a far more difficult problem to solve and required the development of a mathematically intensive model. We first required knowledge of exactly how many fragments a particular explosive produced, how large they might be and their velocity. Research on these topics exists and our approximations are based on Mott theory [10] to describe fragment size distributions and the Gurney equation [11] to describe their velocity.

These approximations give us a set of fragments, each with an initial velocity. If we apply these to a target using a kill grid then our problem becomes similar to that for direct fire, with the additional dimension of fragment size. These are passed to a module called the fragment combiner, which performs the detailed mathematics required to sum the many individual probabilities that a fragment of a particular size might damage a target in a particular location over all possible sizes and locations on the kill grid.

Combining these fragments is mathematically intensive but not terribly difficult if we assume that fragments are uniformly spherically distributed from the explosion point. However, this is almost never the case. Most shells are cylindrically shaped and disperse most of their fragments towards the sides of the cylinder. Some high explosives are actually shaped charges, transmitting most fragments in one direction. Fragment distributions are the subject of a multitude of experiments, which use witness plates to gather empirical data on the subject. However, like most such empirical data, its availability is spotty at best. Therefore, we introduced a value called the asymmetry factor, which represents the proportion of a sphere that the fragments are constrained to. This allows us to easily (albeit roughly) represent most high explosive rounds, but greatly confounds the fragment combination problem, solutions for which are discussed in [12]. Figure 6 shows this combination process in flowchart form.

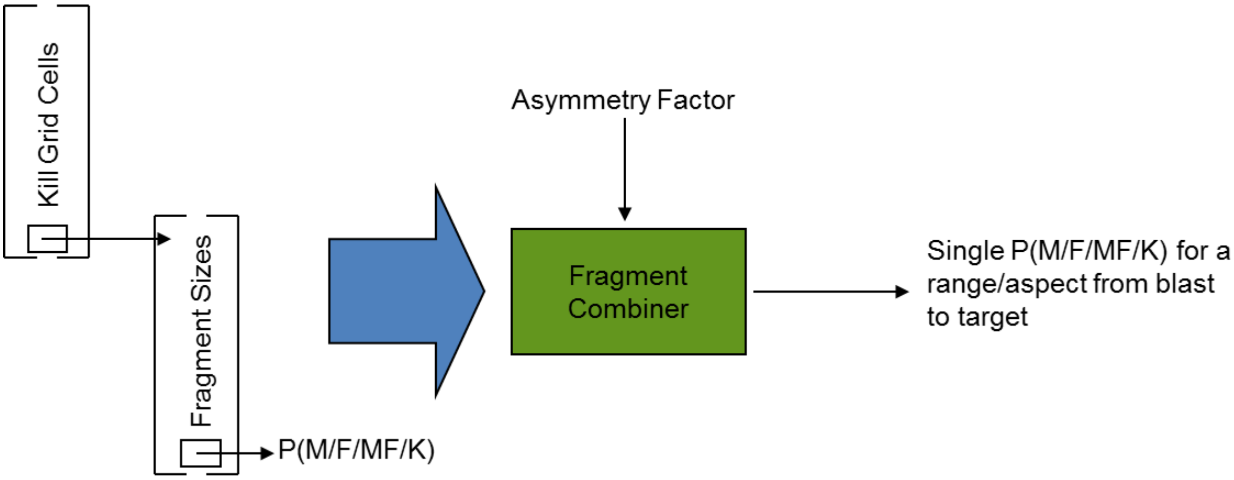


Figure 6 - Fragment Combination Flowchart

6.0 VERIFICATION AND VALIDATION

Our data generation process is necessarily detailed and provides a challenge for Verification and Validation (V&V). We see two types of V&V being especially important in this process. Firstly, we must analyse generated data for internal consistency; for example, that a larger calibre weapon should have greater armour penetration, or that a larger high explosive round has a larger lethal area. Secondly, where possible we compare our data with empirical sources.

There is a temptation to use empirical data directly as a data source. In many cases this makes perfect sense, such as for basic physical properties such as muzzle velocity. However, in the case where empirical data is instead a statistical sample, we do not use it directly, but rather use it to verify the correctness of our models. An example of this is trajectory tables, which are often reasonably available from sources such as field trials or manufacturer's specifications. We could take this data and directly use it in a simulation database, since combat simulations do require this type of information. However, we instead would use such information to verify our own trajectory model; producing a consistent model of data generation, rather than using a combination of empirical data and models in cases where such data is unavailable. We apply this V&V process across the spectrum of our process, starting with information stored in SimR, through intermediate stages of the data generation models and into the simulation database itself. Figure 7 displays a visual representation of this process.

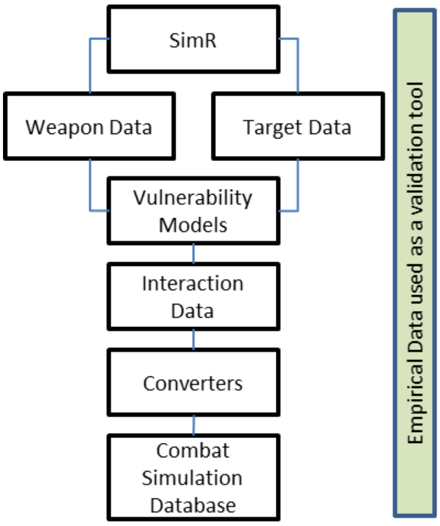


Figure 7 - Verification and Validation Model

7.0 CONCULSION

We have described a detailed, integrated approach to generating, managing and validating input data for use within combat simulations. This allows us to produce fit-for-purpose combat simulation input data which retains an element of traceability and accountability for the sources of this data. Therefore our relatively small team is able to conduct studies using multiple simulations while retaining confidence that these simulations are being used appropriately.

8.0 REFERENCES

[1] Bowden, F. & Williams, P. (2013, in review). A Framework for Determining the Validation of Analytical Campaigns in Defence Experimentation, MODSIM 2013, Adelaide, Australia.

[2] The Technical Cooperation Program (TTCP) (2006). Guide for Understanding and Implementing Defence Experimentation (GUIDEx).

[3] Angel, C., Eigenraam, J., Hemming, D., & Stone, B. (2011). SimR - Centralising Data Storage for Modelling and Simulation. SimTect 2011, Melbourne, Australia.

[4] Mazonka, O. & Shine, D. (2012). Simple Physical Models in Support of Vulnerability and Lethality Data for Wargaming and Simulation Environments, SimTect 2012, Adelaide, Australia.

[5] Mazonka, O. & Shine, D. (2013, in review). Methods and Models in Preparing Weapon-Target Interaction Data for Combat Simulations, MODSIM 2013, Adelaide, Australia.

[6] Roy, R. L. & Cazzolato, F. (2007). An Investigation of Hit Probability Calculations for JCATS (TM-2007-31): Centre for Operational Research and Analysis.

[7] Cazzolato, F., & Roy, R. L. (2006). Calculation of Ballistic Coefficients, Range Tables and Armour Penetration for use by JCATS (TM 2006-39): Centre for Operational Research and Analysis.

[8] Cazzolato, F., Roy, R. L., Levesque, J., & Pond, G. (2007). Probabilities of Kill in JCATS (TM 2007-33): Centre for Operational Research and Analysis.

[9] Australian Army (2005). Land Warfare Procedures – Combat Arms (Engineers) – Blast and Ballistics Effects (LWP-CA (ENGRS) 4-3-2).

[10] Mott, N. F. (1943). Fragmentation of H.E. Shells: a Theoretical Formula for the Distribution of Weights of Fragments. Ministry of Supply.

[11] Cooper, P. W. (1996). Acceleration, Formation, and Flight of Fragments. Explosives Engineering. Wiley-VCH. pp. 385–394.

[12] Mazonka, O. (2013). Cumulative Probability of Blast Fragmentation Effect, arXiv:1305.2285, Cornell University Library.