Proceedings of the 2016 Winter Simulation Conference
T. M. K. Roeder, P. I. Frazier, R. Szechtman, E. Zhou, T. Huschka, and S. E. Chick, eds.

TUTORIAL ON THE ENGINEERING PRINCIPLES OF COMBAT MODELING AND DISTRIBUTED SIMULATION

Andreas Tolk

Simulation Engineering
The MITRE Corporation
903 Enterprise Pkwy Suite 200
Hampton, VA 20666, USA

ABSTRACT

This advanced tutorial introduces the engineering principles of combat modeling and distributed simulation. It starts with the historical context and introduces terms and definitions as well as guidelines of interest in this domain. The combat modeling section introduces the main concepts for modeling of the environment, movement, effects, sensing, communications, and decision making. The distributed simulation section focuses on the challenges of current simulation interoperability standards that support dealing with them. Overall, the tutorial shall introduce the scholar to the operational view (what needs to be modeled), the conceptual view (how to do combat modeling), and the technical view (how to conduct distributed simulation).

1 INTRODUCTION

Combat modeling and distributed simulation are very challenging and interesting topics. I have been teaching a graduate course on this topic for several years at the Old Dominion University in Norfolk, VA. Over the development of the course, more and more students not working in combat modeling related domains joined me, as the complexity of challenges of their domains could often be mapped to the topics of this course, that in many aspects became a course for engineering managers and system engineers in charge of complex simulation-based projects. After the first couple of iterations I decided that I needed a textbook that addresses all the various challenges, which with the help of friends who are experts in their related domains I finally finished some years ago (Tolk 2012). Following the experiences made in teaching this topic for a diverse student body, I structured the topic into four sections.

First, I provide a historical context for the domain of combat modeling and distributed simulation. As discussed in (Page and Smith 1998), the military domain has its own language and is often separated from other simulation experts. The second section therefore explains the general concepts by providing the terms and definitions as well as the universe of discourse for combat models and distributed simulation systems. The next section introduces the referential domain of combat modeling. It looks into concepts and models to cope with the situated environment used to describe the virtual battle space, how to move in it, which effects can occur, and what models are used to represent sensing, communicating, and making decisions. In combat models, these often boil down to move, shoot, look, and communicate. The final section deals with the methodological domain of distributed simulation, including discussions of supporting simulation interoperability standards, such as the Distributed Interactive Simulation (DIS) protocol (IEEE 2012) and the High Level Architecture (HLA) (IEEE 2010).

The objective of this tutorial is that the participant will understand the main principles of combat modeling and distributed simulation. He will know the basic algorithms, constraints, and application areas, and the interplay between the different challenges. Through methods from the fields of operations research,

978-1-5090-4486-3/16/$31.00 ©2016 IEEE

computer science, and engineering, participants are guided through the history, current training practices, and modern methodology related to combat modeling and distributed simulation systems. The tutorial intends to provide a comprehensive overview of the engineering principles and state-of-the-art methods needed to address the many facets of combat modeling and distributed simulation addressing the operational view – what needs to be modeled – the conceptional or referential view – how to model the resulting propertied concepts, activities, and effects — and the technical or methodological view – how to implement and use a distributed simulation solution.

2 HISTORY

To better understand the state of the art regarding combat modeling and distributed simulation it is beneficial to know where we are coming from. The military domain has a long standing history of using games to educate their members in strategic thinking. Games like the Indian Chaturanga, the Chinese game Go, and Chess were often played by nobility to prepare the future decision makers for their tasks. It may be of interest that the idea of maneuvers was not really known before the Roman armies trained their soldiers in ”bloodless battles” first documented around 100 B.C. Most soldiers before were ”trained on the job.” The idea of using maneuvers gave the Romans a huge advantage and became a building block for all future military organizations.

Another significant step was conducted by the Prussians. Baron von Reisswitz introduced the ”Kriegsspiel” in 1811. As the war counselor in Prussia he used a representation of the terrain, different blocks representing the different army branches – like infantry, cavalry, and artillery – guided by a rulebook on movement and attrition to educate his officers. His son introduced the idea of paper maps, standardized figures, and better rules in 1824 and was so successful that the Prussian Chief of Staff von Muffling ordered the use of wargames throughout the Prussian Army. The Prussian successes in battle did lead many other nations to adopt wargaming. Major Livermore improved the attrition models in 1883 by incorporating historical data to validate his numbers and tried to introduce the idea of wargaming to General W. T. Sherman, the Commanding General of the U.S. Army. However, still under the impression of the brutal encounters of the Civil War that required many more human factors than could be captured in wargaming, he discouraged the use of this approach by stating: ”Men are not wooden blocks!” This hindered the use of wargames in the USA until the modern days, when nations like Germany and Japan proved the value of this approach on the battlegrounds of two world wars. From the 1930s, through WWII, and on into the Cold War, the armies of the world (including the United States) developed and employed many different forms of tabletop wargaming. The complexity of the games required, often, large staffs of referees, many complicated charts and tables governing the actions units in the game could be ordered to undertake, and complex calculations concerning the adjudication of not only combat, but logistics operations and integrated movement functions (airlifts, littoral landings, etc).

The first modern simulators were flight simulators, starting with the Link simulator of the 1930s (first released by Ed Link in 1929 as a prototype). This was a simple device that was intended to give training pilots a feel of what it was like to handle the controls in a moving platform, before actually attempting to fly a plane. The company is still in existence as L3, and is still in the Flight Simulator business. The first fully instrumented Link unit was sold to US Navy in 1931 for $1,500. Shortly thereafter, the Army took delivery of its Link Trainers in 1934. Not only were they credited with saving vast sums of money and time, they also saved the lives of pilots during training, and after (with the skills taught). According to a report to the US House of Representatives, these trainers were estimated to have saved the Army Air Corp at least 524 lives, $129,613,105 and 30,692,263 man hours in one year.

In the 1960s as computers became more powerful, and the explosion of ideas concerning input and output devices began, the idea of visualizing data came back around, and this time it was taken seriously. All along this time, vehicle simulators became more and more complex and realistic. Finally, in the 1970s and 1980s the first ground vehicle simulators became available.

In parallel to the simulators, simulations became more powerful as well (definitions for both will be given in section 3.1). They represented more and more complex combat situations in more and more challenging terrains to decision makers in the headquarters in so-called computer assisted exercises (CAX) (Cayirci and Marincic 2009). Constructive simulation systems stimulated common operational pictures and received orders from the command and control systems. The supporting de facto standard was the Aggregate Level Simulation Protocol (ALSP). This protocol supported a world-wide federation of systems in the USA, Germany, and Korea and was very successful. To close the gap between the simulator and the simulation world, the High Level Architecture (HLA) was introduced and internationally standardized as IEEE1516.

Today, the use of simulators and simulations in common federations is the rule for military training and education. The biggest and most expensive exercise may be Millennium Challenge 2002. It brought simulators and simulations from all over the nation together for a three week long event (July 24–Aug. 15, 2002), was sponsored by U.S. Joint Forces Command, and has been estimated to have cost approximately $250 million.

3 GENERAL CONCEPTS

The concepts, terms, and scenario elements of combat modeling were introduced to the Winter Simulation Conference in (Page and Smith 1998). One of the particular challenges in this domain are the military terms and abbreviations, but also the special terms used often uniquely in the simulation descriptions. We can only deal with a very limited subset here and have to refer to additional literature for the interested reader.

3.1 Terms and Definitions

We already used several terms in the last section that may not be familiar to a scholar or researcher of M&S in other application domains than defense, but some concepts are also shared. In this section, the main terms and definitions of concepts are introduced. This is not an easy endeavor. Tuncer Ören compiled a lexicon of thousands of M&S related terms (Ören 2011). In the context of this tutorial, the Glossary of Military Terms (U.S. D.O.D. Joint Staff 2010) as well as the Department of Defense (DoD) Modeling and Simulation Glossary are of particular interest.

Models are target driven, purposeful abstractions and simplifications of a perception of reality. The perception will be shaped by cognitive, physical, and legal constraints. *Simulations* are the execution of models over time, in many cases using computers to execute a programmed version of the model to do so. If the resulting device is used to provide stimuli and feedback to an individual or a group of trainees, this device is a *simulator*. Typical examples are driving simulators or battle simulators that provide a realistic virtual environment in which individuals or crews can train. If a system explicitly provides stimuli for a predefined target system in a predefined structure via predefined interfaces, we talk about a *stimulator*. They are often used to generate test cases for new system, e.g., to check if a new battle command system can handle the required number of incoming messages as specified.

In the same context, *live, virtual, constructive (LVC) simulation* is defined. The easiest way to understand the three concepts is to look at people, systems, and the operation. If real people use the real systems to participate in a simulated operation, then we are talking about live simulations. If real people use simulated systems or simulators to participate in a simulated operation, then we are talking about virtual simulations. If simulated people use simulated systems to participate in a simulated operation, then we are talking about constructive simulations. For military training events it is often advantageous to include all three types to support the needs of the trainees.

The model hierarchy of military simulations is often depicted as a pyramid. On the top, theater/campaign level models allow to analyze force structures or force designs and provide training for high–level decision makers and their staff. The next level are mission or battle level models that are used for doctrine, mission planning, force employment, and force modernization. Many CAX events are also supported by this level of models and simulation systems. The difference between theater and mission level is the scope of the simulated engagement. The activities to defend Western Europe during the Cold War or the military activities conducted within the Operation Iraqi Freedom are theater level, while main events like the 1942–1943 Battle for Stalingrad or the 2003 Battle for Baghdad are examples for the mission level. Tactical improvements as well as weapon system level engagements are covered by engagement level models representing one on one or many to many duels. Finally, the engineering level provides high-resolution support for Research & Development of weapon system components. The effect of new ammunition or new types of armor are simulated on this level.

Another set of terms used to describe the different resolution levels are *entity level simulation* versus *aggregate level simulation*. On the entity level, high resolution models are used representing individual weapon systems, often simulating all military processes individually. If several systems are combined into a unit that becomes the simulated entity, we talk about aggregated simulation. If only weapon systems of the same type are aggregated, we call this a homogeneous unit, otherwise it is a heterogeneous unit.

One of the main challenges in decision making is coping with uncertainties and minimizing the associated risks. If a simulation does not contain any random parameters and always produces the same output for a given input, it is *deterministic*. If probabilistic components are used to represent not only point estimates but to generate variations in the simulation following the laws of statistics, the simulation becomes *stochastic.*

When addressing the universe of discourse for a simulation system, we have to address *scope, resolution, and structure.* Resolution answers the question of how detailed something is modeled in the simulation system. The more detail is added, the higher the resolution. Scope answers the question about what is represented in the simulation system. What has been recognized as important in the viewpoint of one simulation may have been seen to be neglectable in another simulation system. Structure describes how observed details are grouped into concepts. The same attributes can be used to describe different concepts in two simulation models, resulting in different structures that are used to categorize the observations of the real system. These terms are often subsumed under the challenge of multi–resolution modeling.

The next three terms are often confused as well: *fidelity, resolution,* and *credibility.* Fidelity of a simulation is the accuracy of the representation when compared to the real world system represented. A simulation is said to have fidelity if it accurately corresponds to or represents the item or experience it was created to emulate. As discussed above, resolution of a model or a simulation is the degree of detail and precision used in the representation of real world aspects in a model or simulation. Credibility is the level of trust of the user of the model. This level can vary and is user dependent as well as application domain dependent. Credibility is the quality or power of inspiring belief, or the capacity for belief.

The engineering methods of *verification, validation, and accreditation (VV&A)* help determining if a simulation is correct and usable to solve a given problem. Validation is defined as the processes of determining the degree to which a model or simulation is an accurate representation of the real world from the perspective of the intended uses. The scope is therefore the behavioral or representational accuracy. It answers the question: Did we build the correct simulation? Verification is the process of determining that a model or simulation implementation accurately represents the conceptual description and specifications. The scope is transformational accuracy. It answers the question: Did we build the simulation correctly? Accreditation is the official determination that a model or simulation is acceptable to use for a specific purpose. While this is required for military simulations in the USA, other nations are often talking about acceptance and do not apply the same formal process. A good overview of the state of the art for general V&V is given in (Sargent 2013).

3.2 Scenario Elements

In this subsection, some military terms are introduced to describe the elements that most likely will make up a scenario. In order to support the military, the simulation engineer has to understand what the mission is (the big picture), what capabilities are required to conduct the mission successfully, what relations are needed to conduct the tasks in an orchestrated manner, what system can be modeled that provide the needed capabilities as well as the communication, and what the time constraints are. In other word, what is needed for an effective and efficient conducting of the mission?

It is pivotal for the simulation engineer to be aware of the harmonization and alignment principle that addresses the triangle of represented concepts, internal decision logic, and external evaluation logic. It is obvious that if we want to evaluate something, it needs to be modeled. But how much detail is enough? The question can only be answered in collaboration with the sponsor, but every detail needs to play a role in the internal decision rules and must be considered in the external evaluation for success, the measures of effectiveness – how well are needed function performed – and measures of performance – how much do they contribute to the success of the mission. If the internal decision rules are not aligned with what is evaluated as a success externally, we are wasting resources. If the focus lies on different attributes of the concepts, discontinuities will be observed. It is the task of the simulation engineer to avoid this for all scenario elements.

3.2.1 Land Components

Land based operations, such as conducted by the army, are characterized by the distribution and range of the weapon systems, sensors, and communication means, or aggregations thereof. Typical weapon systems comprise

- Infantry is made up of soldiers, sometimes modeled as squads, with handheld firearms, like rifle, machine guns, or even anti-tank missiles. They may be protected by body armor, but, in general, are soft targets that should avoid direct fire without protection.
- Infantry transportation is provided by off-road capable vehicles, like Jeeps or High Mobility Multipurpose Wheeled Vehicle. They are normally not armored and have only light weapons, like mounted machine guns.
- Armored Fighting Vehicles, sometimes referred to as Infantry Fighting Vehicles or Mechanized Infantry Combat Vehicles, are light tanks designed to carry infantry into battle and provide fire support for them. They carry several soldiers that may be able to engage in battles while being in the tank and have light to medium weapon systems for direct fire.
- Main Battle Tanks carry the main part of direct fire battles. They have strong armor and heavy weapons.
- Mortars are high-angle-of-fire weapons that fire ammunition in a high angle so that it falls onto the enemy. Mortars come in several sizes, from small mortars that can be used and carried by infantries to bigger mortars that are part of the artillery.
- Main artillery systems are howitzers and rocket launchers. Howitzers can be towed or self-propelled. As a rule, Howitzers fire ammunition while rocket launchers, often MLRS, launch self-propelled rockets, but there are exceptions for modern howitzers.
- Army aviation focuses most often on helicopters, often referred to as rotary wing air craft, but also uses fixed wing air craft. These are used mainly for transportation and air based fire support.

These systems are aggregated into combat or maneuver units, fire support units, combat engineers, air defense units, and aviation units. Headquarters, communications and networks, and logistics and supply are additional challenges to cope with.

3.2.2 Air Components

Air components are as complex as land operations, but due to the high speed of their operations, the focus lies more often on the individual events connected with aircraft in the sky, so called sorties, than the overall number of entities available. Many models of air operations are therefore more activity driven than there land-based cousins focusing on the entities and objects instead. However, their typical weapon systems include fighters, bombers, and many special task platforms.

- Fighters are highly maneuverable, but often short range aircraft. They engage hostile fighters and escort own bombers to protect them from air attacks.
- Bombers are less maneuverable, but usually have a long range. They are mainly designed to attack ground targets or sea targets dropping bombs or launching shorter range missiles.
- Transportation aircraft in various sizes provide the means for air lift operations.
- Drones are unmanned air vehicles that are controlled remotely for surveillance as well as combat operations.
- Command and Control and Intelligence aircraft provide all kind of means for command, control, communications, and sensing to air and ground forces. Long range surveillance and Airborne Warning systems belong to this group as well.

Many modern aircraft, in particular strategic bombers and intelligence platforms, are stealth platforms that are nearly invisible to radar observers. Due to the technical nature of air warfare, the available capabilities on the ground – ensuring a quick turn-around maximizing the number of sorties – are also important and often make up a significant part of the model.

If space-based entities, such as satellites, are modeled, they often fall under the lead of the air forces, but with increasing importance, more and more simulation systems are introduced with focus on this new element of warfare.

Another topic that traditionally was covered under air operations but deserves its own group of models by now is the ballistic missile defense. In particular nuclear components and intercontinental missile defense are topics covered in models focusing on these topics.

3.2.3 Naval Components

Navy warships are complex and expensive, and are rarely built in large numbers. For models, this provides a special challenge, as even if two ships belong to the same category, they may still have clearly distinguishable capabilities reflecting the technical state of the art when they were built.

Naval forces conduct surface operations, underwater operations, and littoral operations. They can provide massive fire power by naval artillery, including missiles, as well as air power by naval air forces. They engage in sea mine warfare against surface and underwater vessels, actively as well as passively.

There are many vessel types - giving the caveat already mentioned – such as

- Aircraft carriers that are deployable air bases on the sea.
- Battle cruisers and battle ships provide the artillery firepower and missile launching capability of naval force.
- Frigates and corvettes are used to protect battle ships and aircraft carriers, in particular against opposing submarines. Special submarine hunters focus exclusively on battles against enemy submarines.
- Destroyers and cruisers fulfill a similar role as frigates and corvettes, but their main weapon system is the torpedo.
- Tenders provide logistic and maintenance for the navy and the systems.
- Submarines are used for underwater warfare.

Coast guard operations usually fall under another jurisdiction than navy operations. As their tasks – in particular in peace times – are very different from navy tasks, they have to extend navy models to cover tasks like drug interdiction, alien migration interdiction, fishery violation, and search and rescue operations.

3.3 Supporting Guidelines

The North Atlantic Treaty Organization (NATO) Code of Best Practice for Command and Control (C2) Assessment (Alberts et al. 2002) and the Technical Cooperation Program (TTCP) Guide to Experimentation (Labbé et al. 2006) are both guidelines helping the simulation engineer as well as the operations research expert to conduct better simulation-based experiments and analysis.

3.3.1 NATO Code of Best Practice for C2 Assessment

The NATO Code of Best Practice for Command and Control Assessment (COBP) was produced to facilitate high quality assessment in the area of C2. The COBP offers broad guidance on the assessment of C2 for the purposes of supporting a wide variety of decision makers and C2 researchers. The COBP presents a variety of operations and operational research methods related to combat modeling that can be applied and orchestrated in support of analysis and evaluation of C2 related research questions. As such, it is a best practice guide on how to apply various means of operations research within a combat modeling related study. Furthermore, the COBP is the product of international collaboration that has drawn together the operational and analytical experience of leading military and civilian defense experts from across the NATO nations. It represents the common understanding on how to conduct good C2 research within a coalition. In summary, the COBP enhances the understanding of best practice and outlines a structured process for the conduct of operational analysis for C2. It shows how to structure a study that utilizes combat modeling and distributed simulation. It can be downloaded without cost.

3.3.2 TTCP Guide to Experimentation GUIDEx

TTCP is an international organization that collaborates in defense scientific and technical information exchange, program harmonization and alignment, and shared research activities for the five nations: Australia, Canada, New Zealand, United Kingdom, and the USA. The TTCP Guide for Understanding and Implementing Defense Experimentation (GUIDEx) provides critical guidance to support successful defense experimentation. It has been produced by defense experimentation expert representatives from the defense science and technology (S&T) organizations of these nations. Like the NATO COBP, it is distributed without cost.

The main objective of the GUIDEx is the application of scientific principles to conducting defense experiments. It emphasizes the need for frequent communication with stakeholders and utilizing integrated teams to conduct the work under observance of ethical, environmental, political, multinational, and security issues.

4 COMBAT MODELING

Modeling is the task-driven, purposeful simplification and abstraction of a perception of reality that is shaped by cognitive, physical, and legal constraints. The following subsections will cope with the combat modeling challenges that have to be addressed in every defense related model. In addition to (Tolk 2012), these section utilizes (Deitz and Edwards 2009) and (Strickland 2011).

4.1 Modeling the Environment

The environment is often assumed to be implicitly given, but it deserves as much attention as every other simulated entity. The reason is that the common battle space or battle sphere actually is situated, i.e., it builds the common foundation for how elements move, sense, act, and communicate. If the resulting virtual battle space differs significantly in two combat models that are linked or federated, the results will likely differ significantly as well. If the resulting simulation systems are composed, the result will be an unfair fight: a systemic bias that may be rooted in the different representation of the environment.

The environment comprises everything required by the simulated entities. If these are land entities, the modeling of terrain, cover, surface, etc. is of particular interest. For air force entities, clouds and wind and different temperatures in different air layers are important. For a sea based entities, current and salinity can be as important as the sea state – i.e. height of the waves. The ocean itself has a climate that is for submarines as important as the climate of the atmosphere is for the aircraft. So-called space weather influences satellites and need to be modeled when these entities are needed. In summary, everything that is perceived to be important for the defense domain should be included and not be simplified or abstracted away. If a certain attribute of the environment is needed, it has to be captured and modeled somewhere.

To give an example, modeling the environment for land-based entities is more than just using the terrain elevation. Many other factors are often needed, such as terrain roughness, the degree of urbanization and/or forestation, the vegetation and soil type, rivers, roads, and bridges, natural and man-made obstacles and barriers, precipitation, weight bearing capacity and many more details. Questions like "Does the season make a difference for the model, as trees and bushes may have leaves or not?" have to be asked and answered.

For modeling the terrain, as a rule cells are used that capture the various properties, such as height or cover, as attributes. To compute the influence of the terrain on the current status and activities of a simulated entity, the attributes of the cell the entity is in as well as the attributes affected by the activity have to be considered in the computation. There are only three regular shapes that can be used to cover a plane without leaving gaps (excluding Escher–like irregular shapes): triangles, parallelograms, and hexagons. Triangles have the advantage that three points define a plane so that an area approximated by triangles can easily be visualized without any gaps in the representation of elevation, as long as the elevation is stored in the corners of the triangles. Parallelograms - with rectangles being a subgroup - allow the approximation of the terrain via a chessboard like structure of cells, allowing in particular the use of Cartesian grids, such as they are used in military maps. Hexagons allow the use of the advantages of triangles (by storing the elevation in each of the six corners as well as in the center, thus using de facto six triangles to represent the elevation), plus they provide for computation friendly definitions of distances: when numbering the hexagons accordingly the distance of two hexagons can be derived by simple additions and subtractions, avoiding the computationally intensive multiplications and square-root operations needed in Cartesian grid systems.

For all these aspects, we also have to understand if we are adding detail in support of the simulation or of the visualization. In the ideal case, every detail should be considered for simulation as well as visualization, but this practice is not always followed in real-world applications. It is good practice to understand visualization as a special form of external evaluation and apply the *Harmonization and Alignment Principle:* only if an attribute is used for the internal decision process it should also be visualized. Otherwise, we can easily become guilty to work with "smoke and mirrors" as our simulation visualization presents a different picture than the one used by the simulation itself.

This approach of using grid cells to capture the main characteristics is comparable to creating a "game board" on which the simulated entities are acting. Moving, sensing, and acting is guided by rules, similar to those being created for war games, just that they are captured as algorithms and are parametrized for more accuracy. However, the mental picture of the game board with the simulated entities being the figures influences many of the following algorithms, and even standard development.

4.2 Modeling Movement

Modeling movement is strongly connected to the environment. Movement can be modeled implicitly or explicitly. Implicit mobility models outsource the computation of all mobility factors to produce mobility maps that are used to look-up the possible speed at runtime. Explicit mobility models use the mobility relevant properties of the simulated entity as well as of the relevant parts of the environment to compute the speed on the fly. The list of attributes can be rather impressive. The Army mobility model (AMM) utilizes the following vehicle attributes: vehicle weight, vehicle geometry (in particular ground contact geometry), vehicle power characteristics, dynamic reaction to obstacle impact, vehicle braking characteristics, front end strength, dynamic reaction to rough terrain, and the driver's tolerance to longitudinal shock. In addition, the following environmental characteristics are use to compute the speed: surface type, surface strength, surface roughness, slope, season, precipitation form (rain, snow), precipitation amount, obstacle geometry, obstacle spacing, vegetation size, vegetation density, and visibility characteristics.

Another important factor is the current task conducted by the simulated entity. If simply in transfer mode, cloud cover is no big issue for modern aircraft, but if visual contact is required to fulfill a mission, this can be a major slow-down factor. Similar observations are true for land forces as well: the same system in the same terrain will move differently when it simply moves into a new assembly area or when it is looking for enemies hiding in the terrain.

When modeling movement explicitly, point movement is often used to take these different aspects into account. Attributes of the system are used to compute a number of points that it can use to move in the current simulated time step. The attributes of the environment are also used to compute resistance points. How far a system can move is then defined by the point values. This simple approach allows to add tactical resistance values to terrain cells: if land mines are laid, it increases the value; if hostile systems can shot at you in a certain cell, it increases the value; if artillery shoots into a cell, it increases the value; etc. The total resistance value is then the sum of environment and tactical resistance. Optimization algorithms can now be used to compute the path of least resistance, providing simulated entities with the ability to move using artificial intelligence to behave tactically appropriately.

When using models that aggregate several weapon systems into a unit, or even several units into a higher unit, these point algorithms have to be modified. Usually, the unit schema are used that prescribe the distribution of systems – or units – within this schema. Typical arrangements on the tactical level are lines, columns, or wedges. It is common practice to select a reference system within this schema, often the leader of the formation, that is used to compute the movement. All other systems follow accordingly. The schema is often used to present a tactical standard schema that needs to be adopted to the current terrain constraints. For higher aggregates, shapes like circles or rectangulars are often used. To avoid model artificialities, the tactical schema or often adapted to the terrain.

Finally, the results of a combat situation may influence the speed and movement as well. In particular casualty numbers that usually slow the speed down. The casualty rate is another factor. The way these factors influence the speed may be highly dependent on other states of the unit: a veteran unit may slow the operation down to get a better idea of what is going on while a new unit may panic and rush. In an open terrain, the unit may run for cover, etc. These are decisions the simulation engineer needs to make with the user of the model. In any case, all these factors and their effect need to be captured and documented for validation.

4.3 Modeling Sensing

The easiest way to model sensing is to give simulated entities full access to all the information available in the model: the ground truth. In reality, weapon systems and units do not see and know everything. Their decision is based on a perception of their situation that is incomplete and inaccurate. The more information is provided via communication with other units and the better the results are that are observed by its own sensors, the better is the perception of the unit.

There are many types of sensors: acoustic sensors, like microphones or hydrophones, that listen for sounds in the environment. Chemical sensors that identify chemical and biological substances. Electromagnetic sensors observing changes in the electrical and magnetic field.Thermal sensors, such as infrared sensors, utilize changes in heat. Optical sensors observe the visible spectrum of the electromagnetic spectrum. For all these sensor types, the target–background–ratio is pivotal: if the signal of interest is overshadowed by the same or very similar signals from the background, it can hardly be detected: a weak sound in a noisy environment cannot be heard, a chemical agent that smells just like the environment does cannot be detected. The reason for using camouflage is to blend optically into the environment, etc. Therefore, in order for a sensor to detect a target, three requirements have generally to be fulfilled:

- The sensor has to be able to detect a certain property or a combination of properties (like an infrared spectrum)
- The target exposes at least one of the observable properties (like giving out heat in the detectable infrared spectrum).
- The background does not expose the same observable property or at least is significantly different (the environment is colder than the target).

Not fulfilling one requirement prevents detection of the target. This requires, however, that important attributes observable for weapon systems or units must not only be modeled for the targets, but also for the environment. If we don't know how hot the environment is, we cannot determine if an infrared sensor is effective, etc.

The steps of creating a perception are normally observing the assigned area, detecting that something is present, tracking the movement of this object, classifying the type of the detected object, recognizing whose side the object is on, and identifying the details. In combat, this usually leads to target acquisition.

Line–of–sight algorithms play a special role, as they define if two systems can see each other or whether an obstacle is in the way. To save computing time, they are often used in advance to produce visibility maps that provide the information which environmental cells can be observed from the current one.

Radar and sonar models are more complex than line–of-sight applications and take many additional factors into account, like transmitted power, the gain factor of the antenna, cross–section of the target radar, noise and temperature of the radar system, and more. High resolution models also take the earth curvature into account and compute reflection characteristics of the observed waveform in the observed environment.

There are many options utilized for modeling sensing, from simple cookie–cutter function to computationally expensive high–resolution models of ray tracing and wave distribution. For the simulation engineer it is therefore important to capture and document all these aspects to avoid unfair combinations of applied sensor models that create a systemic bias of this composition, e.g., if one model includes a sensor that can penetrate an environmental obstacle while the other model simple uses line–of–sight based perceptions. It is also important to understand which attributes are used to create a perception and what values for them have what effect on the modeled sensors.

4.4 Modeling Effects

Although there are many effects on the battlefield, the main effect looked for is attrition of the opponent. Most combat models on the entity level are looking at the probability of hitting the target, i.e., how accurate is the shot, and the probability that the hit kills the target, i.e., how efficient is the ammunition used against the armor of the target. To compute the effects, models distinguish between direct fire weapons, that require a line–of–site between shooter and target, and indirect fire weapons.

The standard formula used for direct fire weapons is:

$$P_k = P_{k|h} * P_h \tag{1}$$

P_h is the probability to hit the target with the current shot. The conditional likelihood to destroy a target when it is hit is $P_{k|h}$. The resulting probability to kill a target with a given shot is computed to P_k. If you shot more than one shot at the target, the overall probability to destroy combines with n shots to $P_k^n = 1-(1-P_k)^n$. These n shots can result from one shooter shooting n times in a short period of time or from n shooters shooting at the same target. However, for salvos, like machine gun fire, another formula is used.

Indirect fire weapons compute the effect by the lethal area of one shell A_l compared to the overall target area A_T. One shell destroys a target in the target area with the likelihood of $P_k = A_l/A_T$, A salve of

n indirect fire shells targeted at the same area computes the likelihood to destroy a target in the target area to $P_k^n = 1-(1-P_k)^n$.

Not every shot destroys the target completely. In many combat models, the following damage classes are defined: fire power kill, movement kill, communication kill, and catastrophic kill. The catastrophic kill is a total loss of the target, the other categories are self explaining. The probability computations for these events are equivalent to those described above.

Some models alternatively use a game-based point system to compute if a system is destroyed or not. Every system receives a certain point level in the beginning, and every duel reduces the points while maintenance can increase the points. If the points fall under a certain threshold, the system receives the related damage.

For aggregated combat models, the so-called Lanchester equations are still used to compute attrition of forces. Frederick W, Lanchester formulated them in 1916 to show the usefulness of force accumulation in modern warfare. He looked at units as force collections that mainly decrease the number of opponents within duels while simultaneously being decimated by them as well, both based on attrition coefficients depending on the duel situation. This view results in differential equations describing the battle and the number of forces to be expected on both sides over time.

In direct fire, the amount of destroyed targets on the blue site dB depends only on the number of red shooters at the given time $R(t)$ times the red Lanchester coefficient l_r. The same is true on the opposing site as well. To solve the differential equation for $(B(t)$ and $R(t)$ at any given time, we need to know the initial force numbers B_0 and R_0. The result is the so–called square law for direct fire attrition:

$$l_b[B_0^2 - B^2(t)] = l_r[R_0^2 - R^2(t)] \tag{2}$$

In indirect fire, the number of destroyed targets is proportional to the amount of targets in the target area as well as the amount of shooters shooting into this area. Therefore, the amount of red losses depends on the number of blue shooters, the number of red targets, and the attrition coefficient: $dR = l_b B(t)R(t)$. The blue losses are computed equivalently. Resolving these differential equations results in the linear law for indirect fire attrition:

$$l_b[B_0 - B(t)] = l_r[R_0 - R(t)] \tag{3}$$

The military operations research community derived many additional Lanchastrian equation to support the analyses of attrition. Coefficients were derived analytically as well as empirically. Although often criticized for the many assumptions and constraints, these equations are still in use. So far, no alternative with a similar solid mathematical foundation has been agreed upon.

4.5 Modeling Communications and Decision Making

We already learned about the importance of communication in the creation of a perception, which can be highly improved if information from trusted sources regarding the current situation are received by the unit or weapon system that needs a better situational awareness. Generally, communication between systems and units is pivotal to exchange information and orders between superior and subordinates. Information is also often exchanged between neighbored units. The command and control structure between units is the main guide when setting up theses communication channels. In particular for distributed planning, the communication of operational orders became increasingly important. While many CAX system still outsource the decision making and planning to the training audience, constructive simulations become more and more sophisticated in modeling command, control, and communication of the related pieces of information.

When modeling the communications explicitly, line–of–sight models coupled with range models are still an often used option. If two communication device can share information, like radios working on the same frequency, and they are within range and connected via line–of–sight, they can communicate. More detailed simulation models capture for each information exchange requirement the necessary communication means, the required or usable channels, the required bandwidth, and capacity and time constraints. If more than one option exists, optimization algorithms can be used to compute the best use of all communications means.

Some aggregate models assume perfect connectivity, but allow for time delays. Other models use the connection probability to compute if a message makes it through or not. More and more models explicitly model network communication models, such as the Optimized Network Engineering Tools (OPNET) model group. Newer concepts, like airborne networks, or digital radio based tactical Internet options, require new models that are more and more shared with industry, as they are used for cellphone coverage, etc., as well.

5 DISTRIBUTED SIMULATION

The last sections gave an idea about the multitude of options to model the environment and the entities, and how they move, look, shoot, and communicate in their virtual battle space. It is already a challenge to ensure consistency in a single model, but this challenge increases when several independent simulation systems shall be federated to support a common training event or some analytic activity. The two subsections of this section will first address some general challenges of distributed simulation and then have a short look at supporting interoperability standards.

5.1 Challenges of Distributed Simulation

This subsection focuses on what tasks a simulation engineer will face when executing distributed simulations where independently developed systems are performed on autonomous networked computers supported by information exchange models and protocols that govern the exchange of information between these simulation systems. The tasks of a simulation engineer in this context in general can be summarized as follows:

- Selecting the best simulation systems in support of the task,
- Composing the simulation systems into a federation,
- Exposing the information needed by other simulation systems conform with the selected interoperability protocol,
- Integrating the information provided by other simulation systems via the interoperability protocol into the respective receiving simulation systems,
- Avoiding inconsistencies, anomalies, and unfair fight situations,
- Addressing additional issues regarding multiple interoperability protocols that are used within the federation,
- Ensuring that all simulation systems and information exchange models are initialized consistently,
- Ensuring that all information needed can be exchanged via the supported information exchange models and interoperability protocols during execution.

We will first look at the various roles that simulation systems and the runtime infrastructures have to play before we look into commonalities and differences of interoperability and composability.

5.1.1 Simulation Systems and Runtime Infrastructures

The main reason for building a federation is the coupling of functionality of contributing systems to provide a new capability. To allow this, the common entities, events, and state changes represented in participating simulation systems must be represented in both systems consistently and synchronized, that means the challenges of temporal and mapping inconsistencies must be addressed, as discussed later in this chapter. To this end, the infrastructure that supports the interoperability protocol and the information exchange model must support three requirements: (1) All information exchange elements must be delivered to the correct simulation systems (effectiveness); (2) Only the required information exchange elements must be delivered to the simulation systems (efficiency); and (3) The delivery must happen at the right time (correctness).

Synchronizing time and avoiding time anomalies is one of the most challenging tasks. It is not surprising that many solutions focus on real–time solutions that do not require a complex time-algorithm ensuring consistencies of temporal cause–effect chains in multiple time representations.

Just adapting an interface to a protocol is generally not sufficient to prepare an interoperable solution. No matter how we create the federation, the individual simulation systems must be able to fulfill a set of tasks as well. This needs to be supported by the design of the simulation system from the beginning:

- All information that needs to be provided from the system to the federation needs to be retrieved and mapped to the protocol used.
- All information provided by the federation to the simulation system needs to be read from the protocol and mapped to internal representations.
- The simulation system must consider in its algorithms which information it can change and it needs to update, and which information is owned by another system and only represented for awareness.
- The simulation system must be able to set its time in accordance with the supported protocol, actively and passively.

5.1.2 Interoperability and Composability

The M&S community understands interoperability quite well as the ability to exchange information and to use the data exchanged in the receiving system. *Interoperability* can be engineered into a system or a service after definition and implementation. Alternative data representations can be mediated into each other as long as the constraints are understood. Only when data have to be disaggregated (which requires that the information that got lost in the aggregation process be reinserted) the engineer has the problem from where to extract this needed information, but often heuristics can be applied that lead to satisfactory results.

Composability is different from interoperability. Composability is the consistent representation of truth in all participating systems. It extends the ideas of interoperability by adding the pragmatic level to cover what happens within the receiving system based on the received information. In contrast to interoperability, composability cannot be engineered into a system after the fact. Composability requires often significant changes to the simulation to ensure that a research question is either answered equivalently in all participating simulation systems, or it is not answered at all. Inconsistent versions of truth are not allowed.

5.2 Interoperability Standards

Two standards have been developed for simulation interoperability that will be described here. Many alternative options are possible, like using not standardized, but internationally successfully applied solutions as described in (Powell and Noseworthy 2012), or using more general solutions like semantic web methods, but describing these options goes beyond the context of this tutorial.

5.2.1 IEEE 1278: Distributed Interactive Simulation (DIS)

The IEEE 1278 Standard for Distributed Interactive Simulation (DIS) evolved from the SIMNET project of DARPA. There are five volumes: IEEE 1278.1 – Application Protocols; IEEE 1278.1A – Supplement to Application Protocols: Enumeration and Bit-encoded Values; IEEE 1278.2 – Communication Services and Profiles; IEEE 1278.3 – Exercise Management & Feedback (EMF): Recommended Practice; and IEEE 1278.4 – Verification Validation & Accreditation. Of particular interest for this tutorial are the enumerations that standardize the so–called Protocol Data Units (PDU) used to exchange information.

DIS was mainly developed to support simulators. They are connected via a network supporting the application protocol, such as an Ethernet token ring. The PDUs are broad–casted from the sending simulator to all other simulator. If they can use the information, they do so, otherwise they ignore the data package. The PDUs are standardized to the bit level. Each PDU comprises of a header and the pay load. The header allows the receiving simulator to decide if this data is of interest. They payload comprises the information describing details on the originating and receiving entity and the type of event. There are 50 types defined, such as fire, detonation, and collision events, but also transmitter, designator, and signal events. Some PDUs allow to create new objects or delete objects no longer needed.

The general characteristics of DIS are the absence of any central management; all simulations remain autonomous and are just interconnected by information exchange via PDUs; each simulator has an autonomous perception of the situation; cause-effect responsibilities are distributed for the PDUs to minimize data traffic. There is no time management or data distribution management. The PDUs are transmitted in a ring or on a bus and each simulator uses PDUs that are directed at one of his entities.

5.2.2 IEEE 1516: High Level Architecture (HLA)

The IEEE 1516 Standard for Modeling and Simulation High Level Architecture is defined by three core volumes, all updated in 2010: IEEE 1516 – Framework and Rules; IEEE 1516.1 – Federate Interface Specification; and IEEE 1516.2 – Object Model Template (OMT) Specification. In addition, the IEEE 1730-2010 - Recommended Practice for Distributed Simulation Engineering and Execution Process (DSEEP), augmented by the IEEE 1730.1-2013 IEEE Recommended Practice for Distributed Simulation Engineering and Execution Process Multi-Architecture Overlay (DMAO), are of interest, as they define rules and guidelines on the development of standardized simulation federations.

HLA was developed to unify various distributed simulation approaches within the US DoD and has been adopted by NATO as well. The objective of defining the HLA was to define a general purpose architecture for distributed computer simulation systems. It defines a federation made up out of federates, which are the simulation systems, and the connection middleware that allows the information exchange between the simulation systems. To this end, three components are defined by the technical parts of the standards:

- The HLA Rules describes the general principles defining how the federation and the participating federates work together, i.e., how responsibilities for updates are shared, who does what when, etc.
- The Interface Specification between the connection middleware which is called Runtime Infrastructure (RTI) and a federate, which provides an application interface in both direction: what services provided by the RTI the simulation system can call, and what services the RTI will call in order to request something from the simulation system.
- The Object Model Template (OMT) that defines the structure of the information exchange between the federates via the RTI.

In order to make sure that (1) all information required is provided to the right federate, (2) only the information required is provided to the right federate, and (3) the information is provided at the correct time, six management areas are provided for effectiveness, efficiency, and timeliness: federation, declaration, object, data distribution, time, and ownership management.

The Object Model Template (OMT) defines what information can be exchanged. In principle, there are two categories of information that can be exchanged, which are persistent objects and transient interactions. The main difference is that interactions are distributed just once while objects are created, they can be updated, they can change ownership, and they can be destroyed. All interactions and objects including parameters and attributes and other definitions build the Federation Object Model (FOM). The information exchange within the federation is done in orchestration of RTI services with the OMT definitions. The information provided in the OMT defines what information can be exchanged between the participating federates, the services provided by the RTI defines define how the information can be exchanged.

6 CONCLUDING REMARKS

This tutorial could hardly scratch on the surface of all topics. The simulation engineer supporting this domain has to an expert in many domains and support bridging many gaps between important experts. He needs to understand the fundamentals of combat and the related missions and tasks, he has to know the basics about the weapon systems and the tactics and procedures, and he has to understand how to model all aspects accordingly. Once modeled, the simulation system must be implemented allowing to be used in distributed operations and exercises. Therefore, he needs to understand the computational and conceptual challenges of distributed computing, applied to the defense domain. As such, combat modeling and distributed simulation remain one of the most challenging application domains within the M&S discipline and provide many valuable lessons learned for other domains interested to apply M&S in their field on a comparable scale, such as health care and medical simulation are currently aiming at.

REFERENCES

Alberts, D. S. et al. 2002. *NATO Code of Best Practice for Command and Control Assessment*. Washington, DC: CCRP Press.

Cayirci, E., and D. Marincic. 2009. *Computer Assisted Exercises and Training: A Reference Guide*. Hoboken, NJ: John Wiley & Sons.

Deitz, P. H., and E. W. Edwards. 2009. *Fundamentals of Ground Combat System Ballistic Vulnerability/Lethality*. American Institute of Aeronautics and Astronautics.

IEEE 2010. *IEEE Standard for Modeling and Simulation (M&S) High Level Architecture (HLA) – IEEE Std 1516-2010, 1516.1-2010, 1516.2-2010*. Piscataway, NJ: Institute of Electrical and Electronics Engineers, Inc.

IEEE 2012. *IEEE Standard for Distributed Interactive Simulation – IEEE 1278.1-2012*. Piscataway, NJ: Institute of Electrical and Electronics Engineers, Inc.

Labbé, P. et al. 2006. *Guide for Understanding and Implementing Defense Experimentation GUIDEx*. Norfolk, VA: The Technical Cooperation Program, NATO ACT.

Ören, T. 2011. "The many Facets of Simulation through a Collection of about 100 Definitions". *SCS M&S Magazine* 2 (2): 82–92.

Page, E. H., and R. Smith. 1998. "Introduction to Military Training Simulation: a Guide for Discrete Event Simulationists". In *Proceedings of the 1998 Winter Simulation Conference*, edited by J. C. D.J. Medeiros, E.F. Watson and M. Manivannan, 53–60. Piscataway, NJ: Institute of Electrical and Electronics Engineers, Inc.

Powell, E. T., and J. R. Noseworthy. 2012. "The Test and Training Enabling Architecture (TENA)". In *Engineering Principles of Combat Modeling and Distributed Simulation*, edited by A. Tolk, Chapter 20, 449–477. Hoboken, NJ: John Wiley & Sons.

Sargent, R. G. 2013. "Verification and Validation of Simulation Models". *Journal of simulation* 7 (1): 12–24.

Strickland, J. 2011. *Mathematical Modeling of Warfare and Combat Phenomenon*. Raleigh, NC: Lulu Enterprises Inc.

Tolk, A. 2012. *Engineering Principles of Combat Modeling and Distributed Simulation*. Hoboken, NJ: John Wiley & Sons, Inc.

U.S. D.O.D. Joint Staff 2010. *Department of Defense Dictionary of Military and Associated Terms*. Washington, DC: Department of Defense.

AUTHOR BIOGRAPHY

ANDREAS TOLK is Computer Science Principal in the Simulation Engineering Department of The MITRE Corp., Hampton VA USA. He is a Fellow of the Society for Modeling and Simulation (SCS). His email is atolk@mitre.org.