Modeling and Enforcing Realism Requirements for Simulations

Dr. Suzanne Barber-Simons
The University of Texas at Austin
Phil Hallenbeck
The MITRE Corporation

Abstract

We present a methodology and implementation by which the realism of a simulation can be defined and quantified, and efficiently enforced through the use of standard requirements-engineering techniques using computational models of requirements in a commercial tool. These techniques made highly efficient by a collaborative engineering approach that directly links requirements to test cases and makes all artifacts visible to all stakeholders in near real time.

Prior work has resulted in the conclusion that simulation realism can be defined as minimization of the **risk** of the **simulated environment** incorrectly portraying the real-world environment **factors** that are most influential in the performance of selected **tasks**, [and for a test] using the **system(s)** under consideration.

These environment factors are in turn derived from the application of the military's Mission Variables (Mission, Enemy, Terrain, Troops, Time, and Civil—METT-TC), plus the Immersive Environment, to the missions and tasks to be performed in the simulation.

Once the influential environment factors have been identified, we require a means by which they can be specified and enforced as requirements for a simulation or a federation. This in turns means that the requirements must be (among other qualities) correct, unambiguous, consistent, and verifiable IAW IEEE 830-98. This is done by modeling these factors as functional and non-functional requirements in a commercial systems engineering tool. (Functional requirements are those that state a requirement to perform a function, such as to simulate the terminal ballistics of a certain weapon against a specified target. Non-functional requirements are those that specify the qualities achieved by the function performance, such as performance or reliability; or the conditions or constraints under which the function must be performed.)

We show that while realism is a highly important concept in simulation requirements, it is not the only important facet; therefore, defining a set of well-formed functional and non-functional requirements with the input of all stakeholders is crucial including users, management and system developers, integrators, and testers. Defining requirements in a computational model that explicitly and directly models the association between requirement and test case makes the accomplishment of these requirements efficiently

verifiable. Further, the requirements can be directly validated against the original domain model.

This collaborative engineering and development methodology and implementation is shown to be highly efficient through the use of a commonly accessible, semantically rich and rigorous computational model of requirements. These models therefore replace textual or other representations and offer the potential of saving significant time and cost compared to them.

The problem of simulation realism

> Commanders are responsible for training units and developing leaders…through the development and execution of progressive, challenging, and realistic training.
> -Army Doctrine Publication 7-0, page 1

In its 2012 Annual Report, the US Department of Defense Director of Operational Testing and Evaluation (DOT&E) stated its concern with the realism of Army tests, saying that "…force-on-force battles must contain enough realism to cause Soldiers and their units to make tactical decisions and react to the real-time conditions on the battlefield" (DOT&E, 2012). This concern has since been repeated in DOT&E's 2013 and 2014 reports (DOT&E, 2013 and DOT&E, 2014).

These statements naturally lead to questions: How can a commander be sure his training is realistic? How much realism is enough "to cause Soldiers and their units to make tactical decisions and react to the real-time conditions on the battlefield"? How will we know if our planned investments will achieve this goal? How, indeed, is realism quantified in the first place?

Numerous mentions have been made over many years of the need for "realism" or a "realistic combat environment." Dr. Ernest Seglie, then the Science Advisor of DOT&E, wrote a lengthy editorial for the International Test and Evaluation Association (ITEA) Journal in 2008 entitled "Enhancing Operational Realism in Test & Evaluation" (Seglie, 2008). More recently, Mr. Steve Daly, DOT&E Chief of Land and Expeditionary Warfare, gave a presentation on operational testing whose 35 slides contained no fewer than 29 uses of the words "realistic" or "realistically" (Daly 2014).

Unfortunately, though numerous examples have been cited of poor realism, and many sources place responsibility for a "realistic" test or a "realistic" training event on one person or another, none of these sources—indeed, no publication or reference that we have seen to date—defines just what is meant by this term, let alone puts it into usable form for a given test or training event.

Realism is not an abstract or immeasurable concept

Douglas W. Hubbard, in his book *How to Measure Anything: Finding the Value of Intangibles In Business* (Hubbard, 2014), makes the following claim:

> Anything can be measured. If something can be observed in any way at all, it lends itself to some type of measurement method. …you've heard of "intangibles" in your own organization—things that presumably defy measurement of any type… The intangible could even be the single most important determinant of success or failure…

Hubbard goes on to cite examples of successfully measuring—even measuring the costs and benefits of—a wide variety of apparently-vague things such as environmental improvements, disease reduction, and computer security. Given the attention paid to it and examples given by senior leaders such as Seglie, it should be apparent to the reader that realism is indeed observable; therefore, according to Hubbard, it is measurable. The following example will illustrate that this is indeed the case.

Consider the example of two opposing main battle tanks engaging each other.

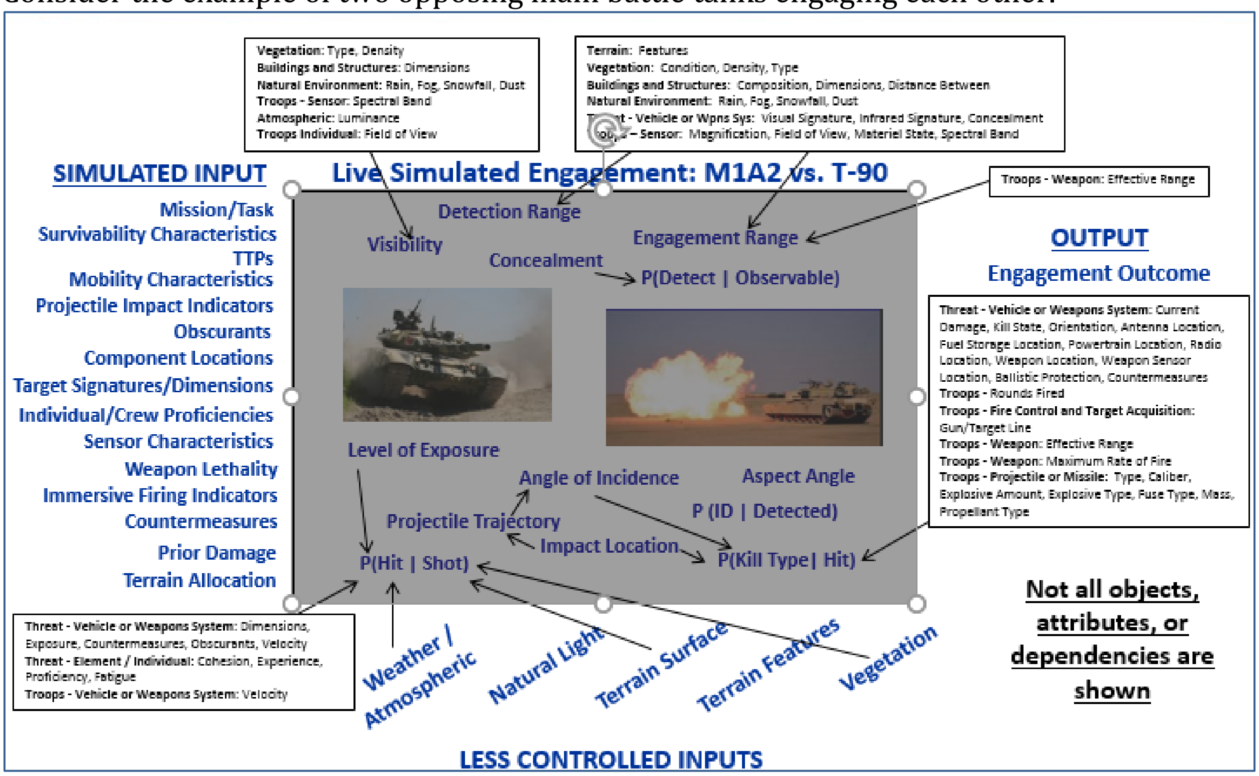

Figure 1. Main battle tank engagement.

Some of the environmental and other factors that might influence the outcome are listed in Figure 1, above.

While it is clear that the list of factors possibly influencing even this simple engagement is rather large, it is equally clear that *all of them are observable* in some form, and many if not most *already have measurements associated with them*. We can, for instance, describe weapon lethality in terms of the probability of killing a certain target given a shot at a certain range using a given weapon and ammunition—indeed, it is measurements like this one that appear to form the basis of DOT&E's concerns with simulation realism. We see from this example, however, that the list of things likely to influence the engagement outcome is much longer than simply a citation of the qualities of weapons and targets. What, for instance, of the proficiency of the respective crews? What if one crew is operating under rules of engagement (ROE) that are far more restrictive than the other? What is the influence of differing terrain or visibility conditions, and what if one side is attacking and the other defending?
It is probably apparent to the reader that some of these factors will very probably have greater influence than others on the engagement outcome, so the consequences of "doing it wrong" will be greater than for others. In a related vein, some (such as the missions and tasks given to each side) will be much easier to simulate correctly than others (such as the lethality of advanced weapons and munitions): The probability of "doing it wrong" will vary.

The definition of realism

The above considerations lead us to a straightforward definition of realism:

> Realism is the minimization of **risk** of the **simulated environment** incorrectly portraying the real-world environment **factors** that are most influential in the performance of selected **tasks (**and for a test) using the **system(s)** under consideration.

In turn, we define this risk in accordance with the Army's Field Manual (FM) 5-19, Composite Risk Management (US Army, 2006): The probability and severity of adverse impacts—but in this case to realism rather than to safety or to mission accomplishment. This will be further explained below.

Realism in terms of risk

As briefly discussed above, it is apparent that some of the items in Figure 1 may have a large impact on outcomes, but are relatively easily simulated—there is little chance of "getting it wrong." The choice of missions and tasks for each side, for instance, can probably be done with little chance of a problem. Other items, however, such as the simulation of lethality characteristics, not only are likely to have large impact but are often (or even always) poorly simulated. Consider, for instance, the capability of current engagement simulations like the Multiple Integrated Laser Engagement system (MILES) to simulate the lethality of kinetic-energy munitions firing through light vegetation: While a few leaves can block a MILES laser signal, the real munition can often perforate many meters of solid material such as earth or wood while still retaining enormous lethality.

These two considerations—the likelihood of a realism problem, and its severity—lead us to consider realism in terms of risk, as shown in Figure 2, below.

Readers with military risk-assessment experience will recognize the similarity of Figure 2 to the risk assessment matrix found in Army FM 5-19, Composite Risk Management (US Army, 2006). This is deliberate, making it easy for military users of the model to consider realism risk in the same way they consider risks to safety or to mission accomplishment. The only differences between the two methodologies are the specific definitions of severity and likelihood, which are described in Table 1, below. (The likelihood definitions are very similar to those in FM 5-19 but have been reworded for clarity in the context of a single event such as a test or training event.)

Impact Severity	Probability of Inadequately Simulating: Very likely	Likely	Occasional	Seldom	Unlikely	
Catastrophic	E	E	H	H	M	E - Extremely High
Critical	E	H	H	M	L	H - High
Marginal	H	M	M	L	L	M - Moderate
Negligible	M	L	L	L	L	L - Low

Figure 2. Realism Risk Assessment Matrix.

Table 1. Realism Severity and Likelihood Definitions.

Severity				Likelihood	
Level	**Change to combat outcomes**	**Impact on event validity**		**Level**	**Definition**
Catastrophic	Greatly change	Invalidate		Very likely	Will happen regularly
Critical	Significantly change	Severely degrade		Likely	Will happen at some point
Marginal	Marginally change	Minor degradation		Occasional	Not certain, but would not be uncommon
Negligible	Almost imperceptibly chang	No significant degradation		Seldom	Remotely possible, but could happen
				Unlikely	Almost certainly won't happen, but not impossible

In each case, we state the risk proposition in terms of failure to adequately simulate a phenomenon, *or uncontrolled changes to it*. The addition of control accounts for the knowledge that some factors—such as terrain and weather—are often easily simulated realistically, especially in live environments; but their impact is such that they must be controlled, especially when comparison or other analysis is desired. It would be inappropriate, for instance, to require one platoon to conduct an attack in daylight and good weather while its sister unit conducts the same mission at night and in heavy rain.

Example Assessment and Results

As we developed the model and began presenting it to interested stakeholders in the US test and evaluation community, some expressed interest in a “trial run” of the model on a near-term event so that its utility could be assessed, and suggested the upcoming Joint Light Tactical Vehicle (JLTV) Limited User Test (LUT). At the time of the request, development of our realism software application was incomplete, so we used a manual assessment of the 29 factors cited above in Figure 4.

Manual Assessment Methodology

To perform the manual assessment at the best possible level of rigor, we obtained a copy of the test scenario (containing the set of missions to be performed in the test), and a copy of the Army Training Circular (TC) 3-21.12, Weapons and Antiarmor Company Collective Task Publication (US Army, 2012). (TC 3-21.12 contains a detailed description of all the missions performed by this type of unit, including their decomposition into collective and leader tasks, and the standard for accomplishment for each.) We also discussed the test plan with the test officer and test operations officer to learn the specific plans for the combat simulation, such as the simulations used for direct-fire engagements and for artillery and mortar support.

Using the missions from the test scenario, an analyst (a retired combat-arms officer with long experience in operational testing) proceeded step by step through the tasks in the Training Circular, assessing the realism risk of each factor in Figure 3 with respect to the accomplishment of each task, using an Excel spreadsheet to record the risk assessments. This resulted in a 464-row spreadsheet, with one line for each task step performed in the test scenario.

The initial analyst’s risk assessment was then quality-checked by three additional military-experienced analysts, and finally presented to the test operations officer (an infantry officer with combat experience in Iraq as a company commander) for review. This process took about 40 manhours to complete—20 hours by the producing analyst, and another 20 manhours for quality-checking. A partial display of the result of this assessment is shown in Figure 3, below.

	Enemy										
	OPFOR TTPs	OPFOR proficiency, including MILES	Comms/Network Characteristics	Mobility Characteristics	Engagement Simulation	Unit Type, Strength, and Task Org	Survivability Characteristics	Kinetic Weapon Lethality	Cyber and EW Weapon Lethality	Supporting Arms and Services	Sensor Performance
TASK: CONDUCT AN ATTACK											
10. The unit executes the attack. It takes the following actions:											
a. Moves to the line of departure (LD) using a technique and formation based on the factors of METT- TC (may be executed by other unit leaders while the unit leader is forward conducting a leader's reconnaissance).		M			M			M		M	L
b. Navigates from checkpoint to checkpoint or phase line by using basic land navigation skills supplemented by precision navigation.									M	M	
c. Moves from the LD through the assault position to support positions, assault positions, or breach or bypass sites. Pauses in the assault position, if absolutely necessary, to ensure synchronization of all friendly forces. Takes the following actions:		H	M		E	M	E	E	H	H	H

Figure 3. Manual realism risk assessment of JLTV LUT.

After the initial risk assessment was complete, the number of spreadsheet cells at each risk level (Extremely High, High, Moderate, Low) was totaled for each Realism Factor. Using a weighting scheme (5 for Extremely High, 4 for High, 2 for Moderate, and 1 for Low risks), and aggregating the risks for Enemy and Troops factors (which have the same structures), we arrived at a ranking of realism risks for the test, from greatest to least.

This pre-event ranking indicated that the highest risks to the realism of the test would be, in order:

- Friendly and enemy proficiency, including with the engagement simulation
- The simulation of weapon lethality
- The engagement simulation, such as the simulation of projectile trajectory and interactions of projectiles with the environment
- The simulation of target survivability
- Visual effects in the immersive environment, such as of tracers and round impacts
- The presence and effectiveness of obscurants
- The characteristics of networks and communications capabilities
- The specific use of terrain among trials
- Supporting arms and services such as artillery or mortars, and
- Weather and light conditions differing among trials.

Post-event feedback from the test and evaluation team was quite positive. The test operations officer said that his experience at the test suggested no changes were needed to the model for combat systems, and in fact highlighted the importance of two of the highest risks. He said that the proficiency of friendly and enemy troops, especially with the engagement simulation, had a very strong impact on outcomes, and that the visual and haptic components of the immersive environment—especially the simulation of improvised explosive devices (IEDs) had a strong influence on soldier behavior. The team lead called the model results "spot-on."

Benefits of measuring realism

We believe that defining and measuring realism has at several significant benefits to practitioners: Reasoning about and communicating realism; assessing realism of an event or plan; assessing realism of a solution; optimizing the realism of a portfolio of solutions; and making (and defending) event or solution design decisions.

First, a model of realism allows for coherent reasoning about realism, and clear communications among stakeholders with respect to the subject. No longer is realism only "in the eye of the beholder:" The model facilitates identification, prioritization, and communication of specific aspects of the simulation environment and their impacts to realism—in terms of both the severity of realism problems and their probability of occurrence. Hence, stakeholders across a community—or even in different communities—can reason about and discuss realism of an event, or that provided by a solution, in a common language and have full understanding of realism concerns and their probable validity.

Closely related to this is the ability to explicitly assess realism of an event—or a planned event—and take action early to identify and mitigate risks to realism. This in turn suggests that the traditional scope of simulation verification and validation (V&V) should be broadened from the V&V of one or more simulation solutions to V&V of the simulation environment.

In a related manner, users of the model can assess the improvements to realism provided by a solution—or the prospective improvements provided by a future system—and explicitly assign costs and benefits to the solution. These results can be aggregated to assess and optimize the realism provided by a portfolio of simulation solutions.

Perhaps most importantly, design decisions for an event or a solution can be made—and communicated, and defended—with respect to a coherent model. While disagreements among stakeholders may remain, they will be explicit, and limited to a certain enumeration of concerns that all stakeholders can see and understand. We believe that this is a particularly powerful benefit, as it provides evidence by which investments in simulations can be defended based on their specific, quantifiable benefits to testing or training. A simple example from the training domain will illustrate this point: If a simulation solution can provide visual signatures of muzzle flashes, tracers or missile flight, munition impacts, and target damage—all instrumental to the task of controlling and distributing fires—this task can be trained using this simulation, whereas it could not be trained in a simulation that cannot provide such signatures.

This last point is also, however, the most difficult, especially with respect to a solution: Design decisions of course depend critically on requirements—but how do we specify and enforce realism in software?

Requirements refresher: IEEE 830-98

IEEE Standard 830-1998, IEEE Recommended Practice for Software Requirements Specifications (IEEE 1998), says that a Software Requirements Specification or SRS should be—

- Correct (every requirement is one that the software shall meet);
- Unambiguous (every requirement has only one interpretation);
- Complete (all significant requirements are included; all responses of the software to all classes of input in all situations are included—for both valid and invalid input values);
- Consistent (no subset of requirements conflict);
- Ranked for importance or stability;
- Verifiable (some finite cost-effective process exists by which we can check that the software meets the requirement);
- Modifiable, and
- Traceable (the origin of requirements is clear, and the SRS facilitates the referencing of the requirement in development or documentation.

In turn, it is clear by the inspection of this list that these qualities are in many cases attributes of the requirements themselves: Unambiguously modeling a single requirement, for instance, is obviously a necessary predicate to having an unambiguous requirements model overall.

It is also probably clear that meeting these qualities can be extremely difficult even for a few requirements. We argue that being able to confidently meet them in a large requirements set, especially with a large stakeholder community, requires the use of a tool. We introduce the use of one such tool below.

Functional and Non-functional Requirements

One highly useful construct in requirements engineering that is only briefly addressed by IEEE 830-1998 is the concept of functional and non-functional requirements. As their name implies, functional requirements "define the fundamental actions that must take place in the software in accepting and processing the inputs and in processing and generating the outputs" (IEEE, 1998).

Non-functional requirements, on the other hand, describe attributes of performance, usability, scalability, reliability, and similar qualities (Leffingwell and Widrig, 2000). Leffingwell and Widrig point out that many of non-functional requirements are in fact attributes of the system or software environment—effectively, constraints on the solution in one form or another. This may include operating in certain environmental conditions—which we shall see is a key concept for enforcing simulation realism.

The concept of non-functional requirements is not only important in terms of capturing all user requirements. It becomes highly useful in efficiently modeling requirements and architecting the software when we realize that a non-functional requirement can have a *scope* that includes (or excludes) specified parts of the software or system. For example, a (non-functional) requirement for reliability may have in its scope the entire system or only some subset of it; or different reliability requirements may be enforced for different parts of the system. The same is of course true for the other non-functional requirements mentioned above; and we will demonstrate that this an especially useful construct for modeling realism requirements.

Functional and Non-functional Requirements in Simulation Realism

Consider the real-world engagement of a target using a given weapon and ammunition. Parameters influencing the outcome of this engagement will include the following:

- Type of weapon and ammunition
- Type of target
- Target exposure (e.g., fully exposed, hull down, turret down)
- Atmospheric visibility conditions, including the use of obscurants such as smoke
- Range from firer to target
- Aim point
- Ballistic solution inputs and outputs, such as rangefinder data
- Biases and errors in the weapon system, munition, and crew—and therefore the actual impact point of the munition which will probably be different than an ideal aim point.

Additionally, participants in the engagement, and bystanders, will experience various visual and audible phenomena associated with the engagement such as muzzle flash and blast; and the signature of tracers, of rounds impacting on target, and of target damage. Further, they will probably experience numerous other battlefield phenomena not related to the engagement in question, such as smoke, dust, and the sight of damaged or destroyed equipment.

Considering a process-based requirements approach in accordance with IEEE 830-1998, we can simulate this engagement using three functions—

- Simulate target (by specific type, e.g., "Simulate T-90S main battle tank as target")
- Simulate engagement (by specific weapon and ammunition, e.g., "Simulate target engagement with Abrams M1A2 SEP v2 firing M829A3")
- Calculate engagement result.

In this example, the third function (Calculate engagement result) receives inputs from the first two functions and then provides the result, taking into account factors such as those listed above.

It is at this point that we can now begin to assess the realism of a simulation, or to impose requirements on a solution. Considering the environmental factors that influence the performance of the task "Engage targets with the tank main gun," we require that our solution fulfills the following requirements—

- Simulate the lethality of the 120mm main gun firing M829A3 ammunition
- Simulate the survivability of the T-90S main battle tank against this weapon-ammunition combination
- Account for target exposure (e.g., fully exposed, hull down, turret down)
- Account for range from firer to target
- Correctly simulate lethality and survivability in the presence of obscurants such as smoke or fog
- Account for the use of the fire control system and the generation of a ballistic solution
- Simulate common biases and errors, and therefore calculate impact point and munition angle of incidence.
- Simulate the visual signature of muzzle flash, tracer, munition impact, and target damage, to firer, target, and bystanders.

An efficient way to use non-functional requirements to model realism in this case is to model the specific "engagement functions" (Simulate T-90S main battle tank as target; Simulate target engagement with Abrams M1A2 SEP v2 firing M829A3; and Calculate Engagement Result) as the functional requirements they appear to be; but to model the rest as a series of non-functional requirements (whose scope includes the functional requirements listed above):

- Engagement Simulation Fidelity – Target, including the following criteria:
    - Account for target exposure: Fully exposed, hull down, turret down (vehicles)
    - Account for target movement: 0-10 km/h, >10 km/h

    - Account for target ballistic survivability by impact point
    - Account for munition lethality by type and range
    - Account for impact location
    - Account for impact angle of incidence
    - Account for target structure survivability with respect to blast
    - Account for target structure survivability with respect to ballistic shock.
    - Account for internal structure and component survivability with respect to shot lines of penetrating projectiles
    - Account for internal component survivability with respect to blast
    - Account for occupant survivability with respect to shot lines
    - Account for occupant survivability with respect to blast
    - Account for occupant survivability with respect to spalling and shattering
- Engagement Simulation Fidelity – Firer, including the following criteria:
    - Simulate delivery accuracy given aim point
    - Kinetic Energy Munitions: Account for munition lethality by type, range, impact point, impact obliquity, and target composition
    - Explosive Munitions: Account for munition lethality by type, impact location, impact obliquity, and target composition
    - Perforate concealment or light cover in accordance with behavior of real-world munition
- Immersive Environment – Visual – Muzzle Flash
    - Simulate muzzle flash of real-world weapon: Luminosity IAW weapon simulated
    - Simulate muzzle flash of real-world weapon: Size IAW weapon simulated
- Immersive Environment – Visual – Tracers
    - Simulate tracer of real-world weapon: Intensity IAW munition simulated
    - Simulate tracer of real-world weapon: Ballistic trajectory of round IAW munition simulated
    - Simulate tracer of real-world weapon: Terminate at impact point IAW munition simulated
- Immersive Environment – Visual – Target Impact
    - Simulate target impacts - projectile impact – luminosity IAW munition simulated
    - Simulate target impacts - projectile impact – size IAW munition simulated
    - Simulate target impacts - projectile impact – location at impact point IAW munition simulated
    - Simulate target impacts – munition detonation – luminosity IAW munition simulated
    - Simulate target impacts - munition detonation – size IAW munition simulated
    - Simulate target impacts - munition detonation – location at impact point IAW munition simulated

- Immersive Environment – Visual – Target Damage
    - Flame in accordance with (IAW) engagement result- visual spectrum
    - Smoke - IAW engagement result visual spectrum
    - Structural damage IAW engagement result - visual spectrum
    - Flame IAW engagement result - Near IR spectrum
    - Flame IAW engagement result - Far IR spectrum
    - Smoke IAW engagement result - Near IR spectrum
    - Smoke IAW engagement result - Far IR spectrum
    - Structural damage IAW engagement result - Near IR spectrum
    - Structural damage IAW engagement result - Far IR spectrum
    - Heated surfaces IAW engagement result - Near IR spectrum
    - Heated surfaces IAW engagement result - Far IR spectrum

Note the reusability and therefore economy of this set of non-functional requirements: We need not model a realism requirement for every target, nor for every weapon-munition combination. Rather, we specify the realism requirement *in terms of the functional requirement* (e.g., simulate target engagement with a specific weapon and ammunition). This results in a necessarily-large number of functional requirements (one for each target and one for each weapon-ammunition combination in the simulation), but a relatively small number of non-functional requirements to express realism (one for each realism attribute of concern).

The complexity problem, 1: Requirements complexity

We now confront one of the truly vexing problem of requirements engineering—complexity. As many engineers have found to their regret, the complete and satisfactory specification of a system—even a small one—may result in dozens or hundreds of requirements. An example set of functional requirements for our engagement simulation might look something like an expansion of the list in Figure 4.

0. Simulate Personnel as Target - Live
0. Simulate Personnel as Target - Live - High Fidelity
0. Simulate Structure or Fortification as Target - Live
0. Simulate Watercraft as Target - Live
0. Update Target State
01. Simulate Target Engagements from AH-64 Series Helicopter
02. Simulate Target Engagements from Abrams MBT
03. Simulate Target Engagements from Bradley IFV or CFV
04. Simulate Target Engagements from M1128 Stryker MGS
04a. Simulate Target Engagements from M1134 Stryker ATGM Vehicle
05. Simulate Target Engagements from M151 PROTECTOR Remote Weapon Station
06 Assess Target Engagement - Live
06. Simulate Target Engagements from T-90 Main Battle Tank
06a. Simulate Target Engagements from T-80 Main Battle Tank
07. Simulate Target Engagements from T-72 Main Battle Tank
07b. Simulate Target Engagements from T-62 Main Battle Tank
07c. Simulate Target Engagements from T-55 Main Battle Tank
08. Simulate Target Engagement from BMP-1 IFV
08. Simulate Target Engagements from BMP-2 IFV
09. Simulate Target Engagements from BMP-3 IFV
1. Maintain Entity State Based on Prior Engagements
1CV-01a: Simulate M1A1 - Abrams as Target - Live
1CV-01b: Simulate M1A1 AIM - Abrams as Target - Live
1CV-01c: Simulate M1A1 AIM SA - Abrams as Target - Live
1CV-01d: Simulate M1A1 FEP as Target - Live
1CV-01e: Simulate M1A2 SEP V1 as Target - Live
1CV-01f: Simulate M1A2 SEP V2 as Target - Live
…
1CV-24: Simulate T-80 Main Battle Tank as Target - Live
1CV-26: Simulate BMP-1 as Target - Live
1CV-27: Simulate BMP-2 as Target - Live
1CV-27a: Simulate BMP-2M Kurganmashzavod as Target - Live
1CV-27b: Simulate BMP-2M Berezhok as Target - Live
…
Simulate 125mm HE Engagement from T-90 MBT - Live
Simulate 125mm HE Engagement from T72 - Live
Simulate 125mm HE Engagement from T80
Simulate 125mm KE Engagement from T-90 MBT - Live
Simulate 125mm KE Engagement from T72 - Live
Simulate 125mm KE Engagement from T80 MBT - Live
Simulate 14.5mm Machinegun Engagement from BTR-70
Simulate 25mm AP Engagement from Bradley IFV or CFV
Simulate Firing Smoke Grenades
…

Figure 4. Example Abbreviated Requirements List.

As the ellipses indicate, the list in Figure 3 is greatly truncated compared to the actual set of requirements for an engagement simulation—and these are the top-level user requirements, *not* the detailed or specification-level requirements.

Consider, however, that to meet the intent of IEEE 830 we must do far more than simply list the requirements—we must add information such as:

- A textual description of the requirement to clarify its context and background.

- References to any supporting documents, such as software or networking standards, or interface controls
- Graphics as required (such as an Integrated Definition (IDEF)0 diagram of inputs and outputs)
- Specifications of inputs and of outputs
- Traceability to—
    - Source documents
    - A source person
    - The requirements elicitation session from which the requirement was gathered
- Unique identifying information for forward traceability
- Contact information for the requirements engineer
- The priority of the requirement
- Start and stop conditions (either formal or informal); or triggers for execution of the requirement
- Conditions or resources required
- Change history of the requirement
- Approval status of the requirement.

It is clear that depicting this information for even one requirement is likely to take several pages of text.

Now, consider attempting to enforce consistency across hundreds of such requirements using text: The auditor is faced with the daunting task of parsing hundreds or even thousands of pages of text in an attempt to find inconsistencies. To say that this task is unappealing and unfruitful would be something of an understatement.

Perhaps even more challenging is maintaining the currency of a text document containing these many (and ever-changing) requirements—to say nothing of the challenge of version control, especially as requirements are approved and therefore "frozen."

The complexity problem 2: Stakeholder complexity

The problem of requirements complexity is exacerbated by the related issue of stakeholder complexity: The project is likely to involve dozens or even hundreds of stakeholders, including—

- Customers
- End users
- Developers
- Testers
- Systems engineers
- Project managers

- Logistics managers
- Budget analysts.

So, the challenge of complexity listed above is effectively multiplied by the large number of stakeholders, each of whom has a reasonable expectation of seeing the most current information with respect to his or her area of concern. Consider further that these stakeholders are likely to be distributed not only nationwide but probably worldwide. Can we really expect to serve this need with a continual series of revisions to thousand-page text documents?

Tool-built requirements models

Our experience indicates that these problems are largely overcome through the use of current systems engineering tools: While the sheer volume of requirements remains challenging, finding and accessing relevant information becomes vastly easier and faster—a matter of a few seconds rather than several minutes. Almost needless to say, over the course of reviewing or modifying many requirements, and ensuring semantic consistency across the entire model, the time and cost savings rapidly add up. The use of one such tool, AWAREness®, is described in this section.

Figure 5, below, shows one view of the tool's requirements engineering main screen, which follows the process-oriented requirements paradigm of IEEE 830-1998. Note the "panes" for Scenarios (business processes), Functions (atomic components of functionality, usually very close to or congruent with system requirements), Non-functional Requirements, Test Events (or deployments), and Data, Events, and Reports (which are the inputs and outputs of Functions).

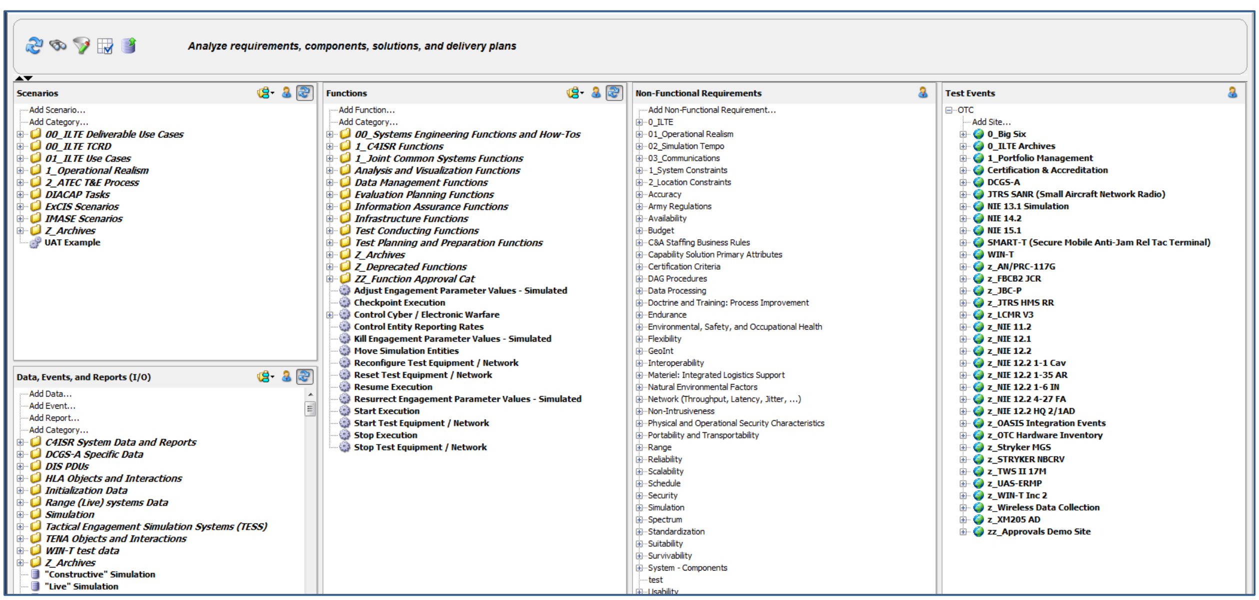

Figure 5. AWAREness® requirements screen.

This depiction provides the user with a rapid *coup d'oeil* overview of the requirements model, allowing rapid and easy selection of items of interest; the color-coding of Functions provides an intuitive understanding of our current capability to fulfill them using a particular set of solutions. Further, it offers a Search function, and allows "querying" of any entity on the screen, a facility that highlights all related items of interest: For instance, querying a Function highlights Scenarios that contain the Function, Inputs and Outputs of the Function, Non-functional Requirements whose scope contain the function, and Test Events (or deployments) that require the function. The results of such a query are shown below in Figure 6.

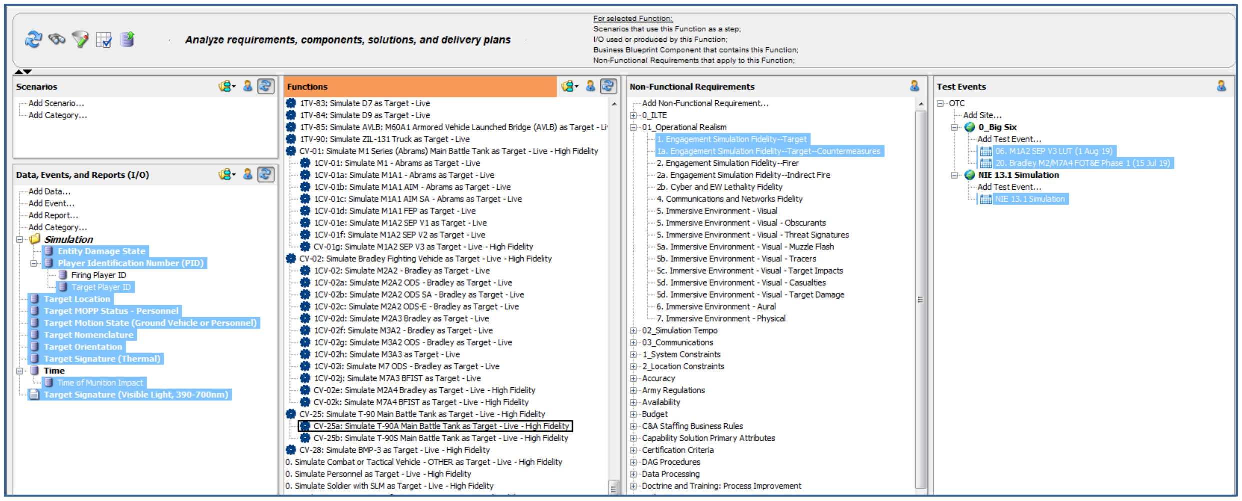

Figure 6. Query of a Function.

The detailed model of a function is shown in Figure 7, below. Using this model, the user can "drill down" to a virtually unlimited level of detail regarding the function using one or two mouse clicks on the relevant buttons. The ability to rapidly find an item of interest and to focus on its relationships and components, down to whatever level of detail is desired, allows the user vastly quicker and more precise information accessibility than textual or purely graphical models (such as IDEF).

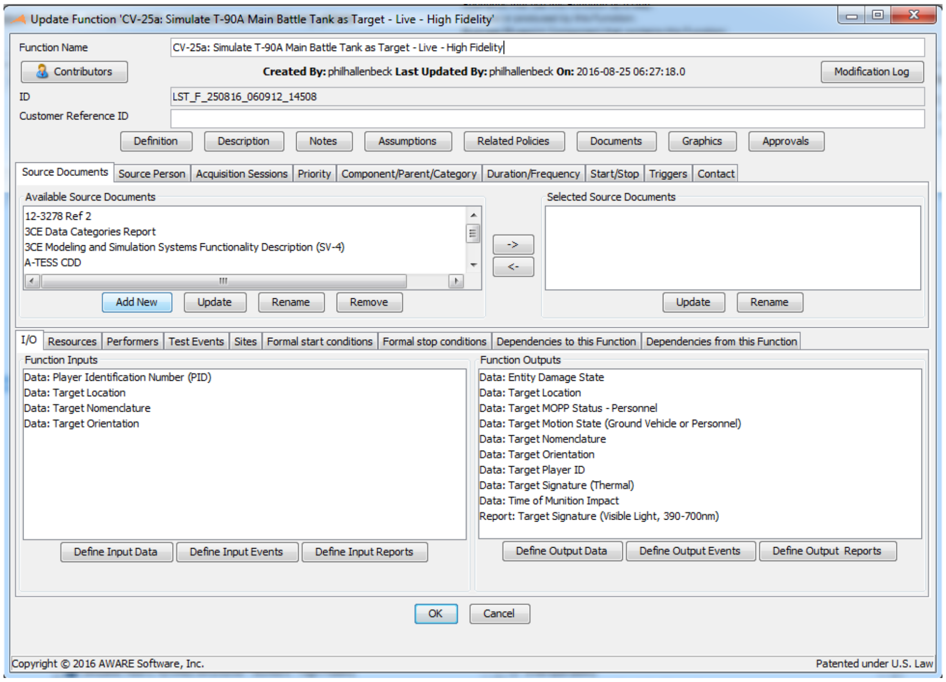

Figure 7. Detailed model of a function.

As the figure indicates, functionality includes the following notable information items:

- Name
- Unique identifier (facilitating forward traceability)
- Definition (graphical and textual definition of what is required)
- Description (a textual explanation of the function)
- Documents (URL references to relevant external documents, such as IETF RFCs)
- The source of the requirement, whether a person, a session, a document, or some combination of these (facilitating backwards traceability)
- The approval status of the requirement (and once approved by the appropriate stakeholder, the requirement is "locked" and cannot be changed)
- Start and Stop condition (both informal and formal)
- Inputs and Outputs of the function.

The model of a non-functional requirement (NFR) is shown in Figure 8, which illustrates the common "look and feel" of the various windows in the tool. Of particular note with respect to the Non-functional Requirement model is the definition of scope (what Functions, Scenarios, or other entities must comply with the NFR); and the acceptability criteria, which describe at any desired level of detail the criteria by which meeting the NFR will be judged.

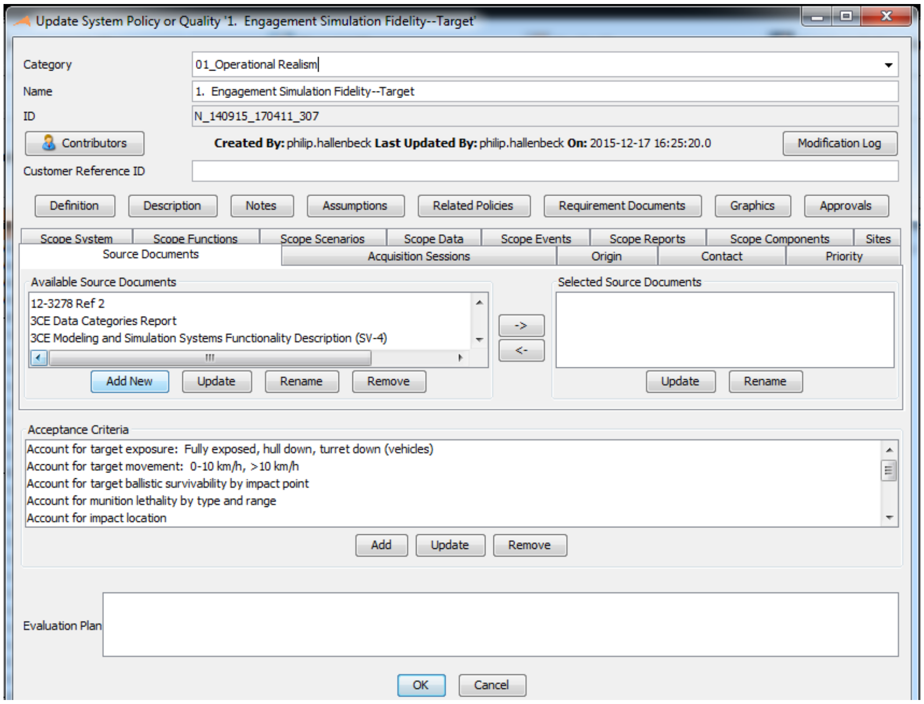

Figure 8. Non-functional requirement model.

Conclusion and Next Steps

We have demonstrated that, far from being a nebulous or "intangible" concept, realism can and should be measured to get the most out of our investments in simulations for testing and training. We believe that the METT-TC + I framework, based on proven doctrine, provides a rigorous yet tractable means of assessing and communicating realism; and the example of the JLTV test—and its practitioner feedback—reinforces this claim.

Hence, we believe we have shown a feasible model and method by which the previously-abstract concept of realism can not only be defined but quantified such that solutions can be developed to specific realism problems. The use of a capable systems engineering tool and process to model and enforce the resulting requirements allows vastly more efficient work—especially for large or distributed teams—than is possible using textual models.

Astute readers of this paper will certainly agree that there are at least two avenues for further work on the topic of realism—

- Development of the relevant environmental factors for tasks not directly related to combat. The examples in this paper, and the majority of our work to date, have focused on the performance of combat tasks such as engaging targets, moving in the presence of enemy fire, and so forth; no less important in the overall picture is the performance of tasks in other domains such as communications, medical services, supply, and transportation. We believe that the basic methodology presented here is applicable to non-military tasks as well.

- Further refinement of the models of relevant environmental factors (modeled as non-functional requirements in the tool). For instance, varying levels of target exposure might be refined from "fully exposed; hull down; hidden" to reflect finer-grained levels of exposure—perhaps as a percentage of the target's visible form.

References

1. DOT&E FY 2012, FY 2013, and FY2014 Annual Reports, retrieved from www.dote.osd.mil/annual-report/index.html.
2. Title 10 United States Code, Section 139, retrieved from https://www.law.cornell.edu/uscode/text/10/139.
3. Seglie, Ernest, Ph.D. (2008). Enhancing Operational Realism in Test &Evaluation. *ITEA Journal* 2008: 29, 121-124.
4. Daly, James (2014). DOT&E 101. Unpublished Powerpoint presentation, Director of Operational Test and Evaluation (DOT&E).
5. Hubbard, Douglas W. (2012). *How to Measure Anything: Finding the Value of Intangibles In Business,* Hoboken: John Wiley and Sons.
6. US Army Field Manual (FM) 5-19, *Composite Risk Management*, 2006.
7. IEEE Standard 830-1998, IEEE Recommended Practice for Software Requirements Specifications (IEEE 1998)
8. Leffingwell, Dean, and Don Widrig (2000). *Managing Software Requirements: A Unified Approach*, Boston: Addison-Wesley.