FFI-rapport 2015/01579

Simulation of land force operations

– a survey of methods and tools

Per-Idar Evensen and Dan Helge Bentsen

FFI Forsvarets forskningsinstitutt

FFI-rapport 2015/01579

Simulation of land force operations
– a survey of methods and tools

Per-Idar Evensen and Dan Helge Bentsen

Norwegian Defence Research Establishment (FFI)

15 February 2016

FFI-rapport 2015/01579

1214

P: ISBN 978-82-464-2674-7
E: ISBN 978-82-464-2675-4

Keywords

Modellering og simulering

Landoperasjoner

Eksperimentering

Analyse

Simuleringsverktøy

Approved by

Halvor Ajer	Research Manager
Jon E. Skjervold	Director

English summary

Military land force operations are complex in nature, and modelling and simulation (M&S) of such operations with sufficient realism is very challenging. This report is a survey of methods and tools for M&S of land force operations at different levels. Furthermore, it summarizes the experiences and lessons learned from working in this field for the last ten years. To model and simulate military operations, it is important to have a conceptual understanding of combat. This survey therefore looks at modelling and simulation of land force operations in the context of modern military theory, analysis, and doctrine.

This survey addresses only computer-based simulations. Within the M&S taxonomy of *live*, *virtual*, and *constructive* (LVC) simulation, it thus only discusses virtual and constructive simulation. Since we are mainly conducting simulations of land force operations for analysis and experimentation purposes, this report has a corresponding focus. However, most of the content should also be relevant for simulation used for training.

In this report we mostly concentrate on M&S of the actual combat phases of land force operations, with engagements and skirmishes. These phases of combat are the most complex and the most challenging to simulate.

Based on the size of the Norwegian Army, our task has been to simulate combat operations with size ranging from platoon to brigade level. The purpose of the simulations has on the platoon to company level been to experiment with and evaluate the operational benefit of new technologies and new concepts. On the brigade level, the purpose of the simulations has been to experiment with and evaluate the operational performance of current and possible future land force structures, including new operational concepts.

In this report we first provide a general introduction to basic M&S, where we go through the terms and concepts that are frequently used in defence-related M&S. Furthermore, we give an overview of the different methods that are used in combat simulation and the most important challenges in this domain. Specially, we look at different methods for modelling the environment, combat units and their core activities (move, observe, engage, and communicate), and human behaviour. Additionally, we briefly describe some of the combat simulation tools that are most widely used today. The report includes two examples of how simulation of land force operations has been used for analysis and experimentation at the Norwegian Defence Research Establishment (FFI). We also summarize the experiences and lessons learned from our simulation experiments, and we give some recommendations for future simulations of land force operations. Finally, we take a look at some of the latest trends in military M&S and present some speculations on how we think the technologies in this field will evolve in the next five to ten years.

Sammendrag

Militære landoperasjoner er komplekse av natur, og modellering og simulering (M&S) av slike operasjoner med tilstrekkelig realisme er svært utfordrende. Denne rapporten er en gjennomgang av metoder og verktøy for M&S av landoperasjoner på forskjellige nivåer. Videre oppsummerer den erfaringer vi har gjort etter å ha jobbet innenfor dette området de siste ti årene. For å simulere militære operasjoner er det viktig å ha en konseptuell forståelse av strid. Denne rapporten ser derfor modellering og simulering av landoperasjoner i sammenheng med moderne militærteori, militæranalyse og doktrine.

Denne rapporten omhandler kun datamaskinbaserte simuleringer. Innenfor kategoriene *live*, *virtual* og *constructive* (LVC) simulering tar den dermed bare for seg virtual og constructive simulering. Siden vi hovedsakelig utfører simuleringer av landoperasjoner for analyse- og eksperimenteringsformål, er det denne anvendelsen av simuleringer som er fokuset for denne rapporten. Mesteparten av innholdet bør likevel også være relevant for simulering til treningsformål.

I denne rapporten konsentrerer vi oss stort sett om M&S av de faktiske kampfasene i landoperasjoner, med engasjementer og trefninger. Disse fasene av striden er de mest komplekse og mest utfordrende å simulere.

Basert på størrelsen til den norske Hæren har vår oppgave vært å simulere operasjoner med størrelse som strekker seg fra tropp- til brigadenivå. Hensikten med simuleringene har på tropp- til kompaninivå vært å eksperimentere med og evaluere operativ nytte av ny teknologi og nye konsepter. På brigadenivå har hensikten med simuleringene vært å eksperimentere med og evaluere den operative ytelsen av nåværende og mulige framtidige landmaktstrukturer, inkludert nye operasjonskonsepter.

Først i denne rapporten gir vi en generell introduksjon til grunnleggende M&S, hvor vi går gjennom begreper og konsepter som blir mye brukt innen forsvarsrelatert M&S. Videre gir vi en oversikt over forskjellige metoder som brukes for å simulere strid, samt de viktigste utfordringene innenfor dette fagfeltet. Spesielt ser vi på forskjellige metoder for å modellere operasjonsmiljøet, stridende enheter og deres hovedaktiviteter (manøvrere, observere, engasjere og kommunisere), samt menneskelig adferd. I tillegg beskrives noen av de stridssimuleringsverktøyene som er mest brukt i dag. Rapporten inkluderer to eksempler på hvordan simulering av landoperasjoner har blitt brukt til analyse og eksperimentering ved FFI. Vi oppsummerer også erfaringene fra disse simuleringseksperimentene og gir noen anbefalinger for framtidige simuleringer av landoperasjoner. Til slutt i denne rapporten ser vi på noen av de siste trendene innenfor forsvarsrelatert M&S og presenterer noen spekulasjoner rundt hvordan vi ser for oss at teknologiene innenfor dette området vil utvikle seg de neste fem til ti årene.

Contents

1 Introduction

Military land force operations are complex in nature, and modelling and simulation (M&S) of such operations, with sufficient realism, is very challenging. This report is a survey of methods and tools for M&S of land force operations at different levels. Furthermore, it summarizes the experiences and lessons learned from working in this field for the last ten years. To model and simulate military operations, it is important to have a conceptual understanding of combat. This survey therefore looks at modelling and simulation of land force operations in the context of modern military theory, analysis, and doctrine.

Much of the work with this survey has been done under FFI-project 1214, "Combat Effectiveness in Land Operations". The simulation experiments used as examples in this report have been conducted under FFI-project 1156, "Technologies for Military Vehicles", and FFI-project 1143, "Future Land Forces".

This survey addresses only computer-based simulations. Within the M&S taxonomy of *live*, *virtual*, and *constructive* (LVC) simulation (see Chapter 2.4), it thus only discusses virtual and constructive simulation. Since we are mainly conducting simulations of land force operations for analysis and experimentation purposes, this report has a corresponding focus. However, most of the content should also be relevant for simulation used for training.

In this report we mostly concentrate on M&S of the actual combat phases of land force operations, with engagements and skirmishes. These phases of combat are the most complex, and the most challenging to simulate.

Based on the size of the Norwegian Army, our task has been to simulate combat operations with size ranging from platoon to brigade level. The goal of our simulations has on the platoon/company level been to experiment with, and evaluate the operational benefit of, new technologies and new concepts [1][2]. On the brigade level the purpose of the simulations has been to experiment with, and evaluate the operational performance of, current, and possible future, land force structures, including new operational concepts [3][4].

Brigade operations include thousands of entities, and combat between brigades are highly complex and highly dynamic. Combat is also adversarial and competitive. It is important to recognize that "although [military operations] currently [appear] to be dominated by technology, [war] is fundamentally a human issue" [5] that is "waged between complex human organizations" [5]. To realistically simulate military operations, the human aspects must therefore be included in the simulation. Modelling human behaviour and cognition, including decision-making and creativity, is very challenging, but these aspects can also be included by having real humans participating in the simulation (human-in-the-loop simulation).

The goal of this report is to give an overview of the state of the art of land combat modelling, including an understanding of the underlying concepts of the methods that are most frequently applied.

We have sometimes been met with an expectation that our combat simulations will be perfect representations of reality. Other times we have been met with an attitude that combat simulations are totally unrealistic. For most simulations the truth lies somewhere in between. This report will hopefully make the reader better able to understand the possibilities and limitations of the methods and tools that are currently being used for simulating combat.

Firstly, Chapter 2 in this report gives a general introduction to basic M&S. In this chapter we go through basic terms and concepts that are frequently used in defence-related M&S. Next, in Chapter 3, we capture the state of the art of M&S of land force operations by giving an overview of the different methods that are used, and the most important challenges in this domain. Specially, we look at different methods for modelling the environment, combat units and their core activities (move, observe, engage, and communicate), and human behaviour. In Chapter 4 we briefly describe some of the combat simulation tools that are most widely used today. After this, in Chapter 5, we provide two examples of how simulation of land force operations has been used for analysis at FFI. In the first example we have used virtual simulations to experiment with, and evaluate the operational benefit of, an augmented reality (AR) system for combat vehicles. In the second example we have used constructive simulations to support land force structure analysis by experimenting with, and comparing the performance of, a set of fundamentally different land force structure concepts. In Chapter 6 we summarize the experiences and lessons learned from our simulation experiments, and give some recommendations for future simulations of land force operations. Finally, in Chapter 7, we take a look at some of the latest trends in military M&S, and present some speculations on how we think the technologies in this field will evolve in the next five to ten years.

2 Basic modelling and simulation

Modelling and simulation (M&S) is "[t]he discipline that comprises the development and/or use of models and simulations" [6]. It includes "[t]he use of models, including emulators, prototypes, simulators, and stimulators, either statically or over time, to develop data as a basis for making managerial or technical decisions" [6]. "M&S in general is often used in situations where exercising or experimenting with the real-world subject of the simulation would be too difficult, too expensive, or too dangerous, and military applications in particular include some of the most extreme examples of difficult, expensive, and dangerous situations" [7]. In this chapter we go through some basic terms and concepts that are frequently used in defence-related M&S. A more comprehensive introduction to modelling and simulation can be found in the book "Modeling and Simulation Fundamentals: Theoretical Underpinnings and Practical Domains" edited by John A. Sokolowski and Catherine M. Banks [8].

Five purposes have been generally established for M&S [9]:

1. An aid for thought
2. An aid to communication
3. An aid for training and instruction
4. An aid to experimentation
5. A tool of prediction

Typical applications of M&S in the military domain are: training, course-of-action (COA) analysis, procurement support, doctrine development, and capability analysis. It should be noted that "[n]o other domain-applied M&S is as successful as the military domain" [10].

2.1 Ways to study a system

A *system* can be defined as "[a] collection of components organized to accomplish a specific function or set of functions" [6]. "A system may be *physical*, something that already exists, or *notional*, a plan or concept for something physical that does not exist" [11]. Often it is not feasible or possible to study the actual system because of cost, availability, safety, or existence. To be able to study a system it is therefore often necessary to build a model as a representation of the system. Figure 2.1 outlines different ways in which a system can be studied [12].

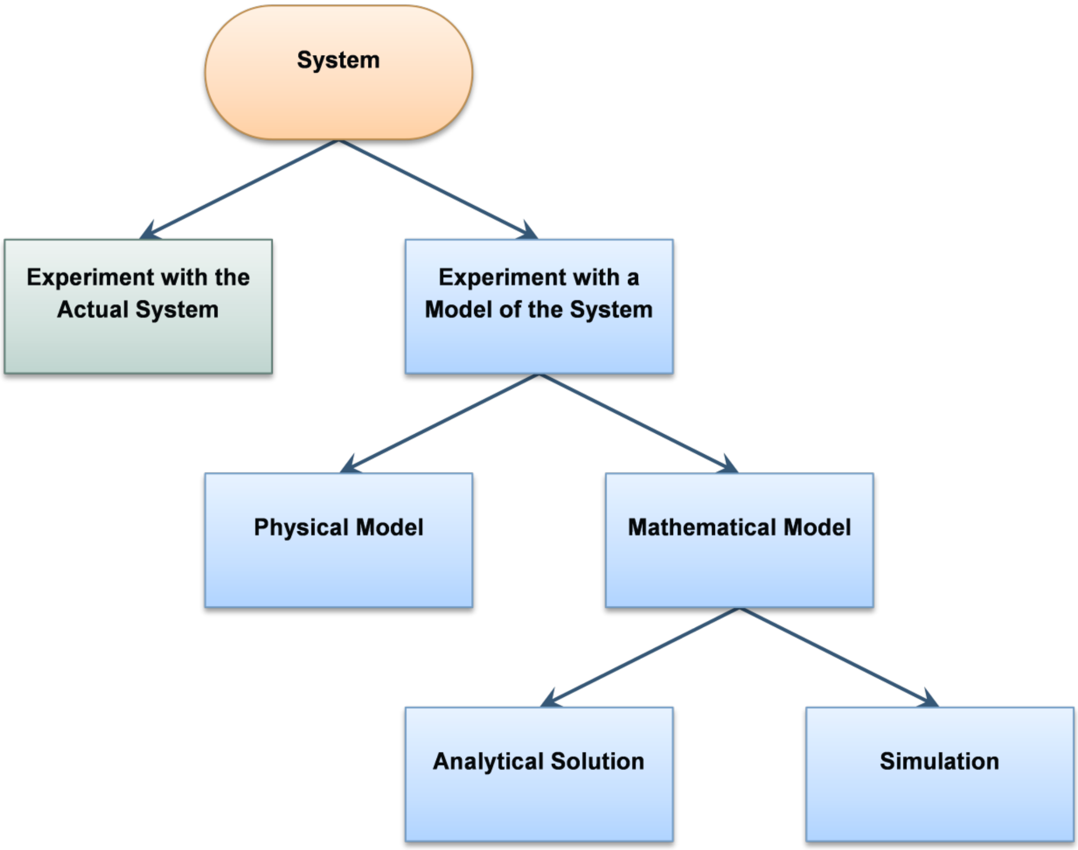


Figure 2.1 Ways to study a system ([12]).

2.2 Models and simulations

The British mathematician and statistician George E. P. Box is known for the phrase: "Essentially, all models are wrong, but some are useful" [13]. A *model* can be defined as "[a] physical, mathematical, or otherwise logical representation of a system, entity, phenomenon, or process" [6]. In more general terms, a model can simply be said to be "a representation of something else" [14].

Models are simplifications, and will never be exact representations of reality. All models of reality are therefore imperfect and incomplete, and by definition (at least partially) "wrong". Some models are, of course, better representations of reality than others, and thus less wrong and hopefully more useful. The important question is whether the model is good enough to serve its specific purpose, and thus good enough to be useful. "On the one hand, a model should be a close approximation to the real system and incorporate most of its salient features. On the other hand, it should not be so complex that it is impossible to understand and experiment with it. A good model is a judicious trade-off between realism and simplicity" [15].

A *simulation* can be defined as "[a] method for implementing a model over *time*" [6]. We refer to the underlying model of a simulation as the *simulation model*. In addition, a simulation typically includes a set of input data, which defines the initial conditions, or the initial state, for the simulation. During the execution of the simulation, the simulation model typically produces some form of output data, which can be considered as the result of the simulation. Figure 2.2 illustrates the basic components of a typical simulation. The terms model and simulation are often used interchangeably, but the important difference is that a simulation is a model implemented over time.

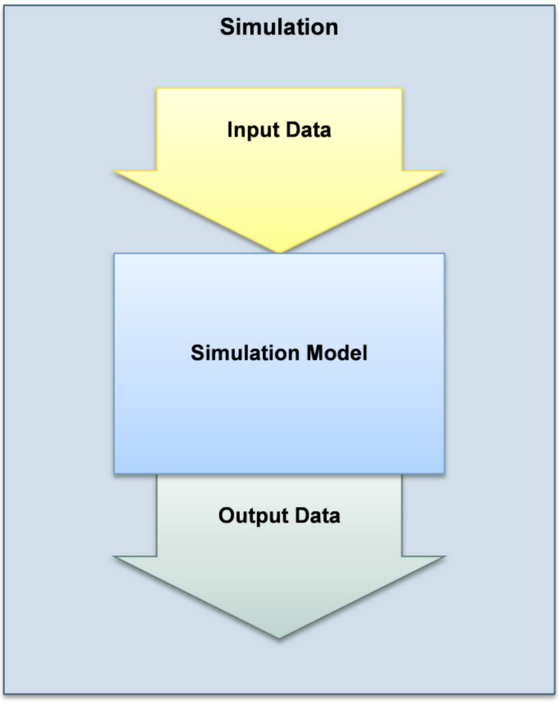


Figure 2.2 The basic components of a typical simulation.

2.3 The military simulation spectrum

Military simulations cover a wide spectrum of activities, ranging from large-scale field exercises, with potentially thousands of people involved, through computer-based simulations with various degree of human involvement, to fully computerized simulations. Figure 2.3 shows the traditional outline of the spectrum of military simulations, together with considered associated operational realism and cost on one side, and considered associated abstraction and convenience and accessibility on the other [16].

As a general scientific principle, the most reliable data come from actual observation of the real world. This also holds true in military analysis, but observing real battles is very often neither convenient nor possible. Military analysts therefore often look towards live field exercises and trials for providing data that are likely to be realistic and verifiable. This data can then generate norms for expected performance under similar conditions in the future. The problem is that "[c]ollective field training is usually constrained, with a controlled enemy, so that the 'right' lessons are learnt from the exercise. In part this is sensible, [since] in the early stages of collective training it is important that a manoeuvre is completed so that the troops can see what should happen. Unfortunately that makes it a very unrealistic simulation of what happens in [real] combat which *is* adversarial" [5]. Moreover, large-scale military exercises, or even smaller-scale ones, are not always feasible or even desirable. Availability of resources, including economic, is a significant factor.

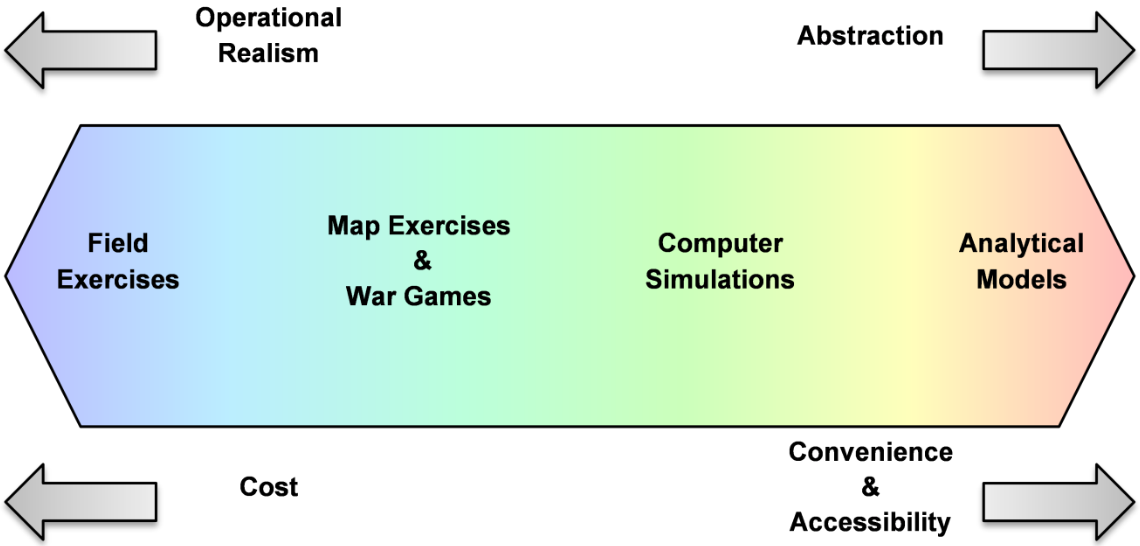


Figure 2.3 The traditional spectrum of military simulations, together with considered associated operational realism and cost on one side, and considered associated abstraction and convenience and accessibility on the other ([16]).

Map exercises and war games involve senior officers and planners, but without the need to move around any real troops, and thus with the advantage of reduced cost and increased accessibility. Map exercises and war games can be manual or computer-assisted. Computer-assisted map exercises and war games are an evolvement of traditional manual map exercises and war games. The computer assistance can vary from just keeping track of unit positions to involve more advanced agent-based semi-automated forces (SAF).

Simulations can also be fully computerized, with automated computer-generated forces (CGF) on both sides. The main advantages of this type of simulation are the accessibility, and the ability to perform thousands of runs in the time it would take a manual or human-in-the-loop simulation to run once. This means statistical information can be collected, and outcomes can be quoted in terms of probabilities.

Since the human aspects are so important in combat, and human behaviour is very difficult to model (see Chapter 3.7), there is obviously a danger in removing the human elements entirely from the simulation. Fully computerized simulations mean that the results are only as good as the simulation model and associated input data themselves (see Figure 2.2). Verification and validation (V&V) thus becomes very important (see Chapter 2.9).

Military simulation is applied on all levels of military operations, from the strategic level, through the operational and tactical levels, to the technical level of individual platforms. Figure 2.4 illustrates this hierarchy of different simulation models on different levels of military operations, together with typical applications.

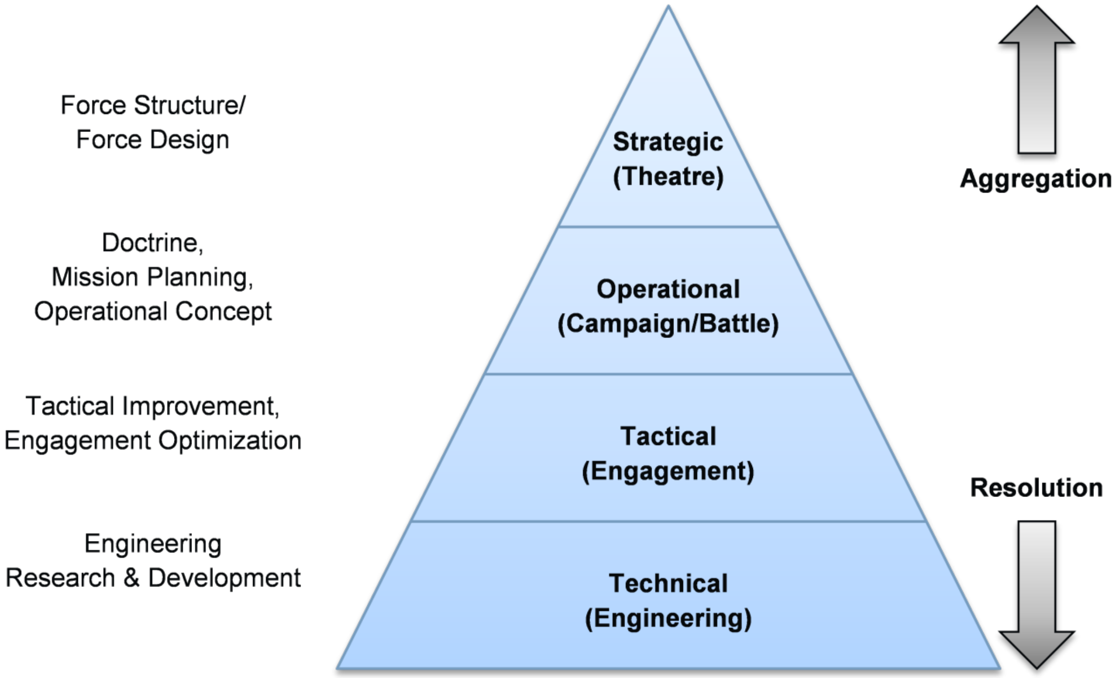


Figure 2.4 Hierarchy of different levels of military simulation.

2.4 Live, virtual, and constructive (LVC) simulation

Live, *virtual*, and *constructive* (LVC) simulation is a broadly used taxonomy for classifying simulation, especially in defence-related M&S:

- *Live* simulation "involves real people operating real systems. Military training events using real equipment, [for example field exercises], are live simulations. They are considered simulations because they are not conducted against a live enemy" [6].

- *Virtual* simulation involves "real people operating simulated systems. Virtual simulations inject human-in-the-loop in a central role by exercising motor control skills (i.e., flying an airplane), decision skills (i.e., committing fire control resources to action), or communication skills (i.e., as members of a C4I team)" [6].

- *Constructive* simulation "includes simulated people operating simulated systems. Real people [can] stimulate (make inputs to) such simulations, but are not involved in determining the outcomes. A constructive simulation is a computer program. For example, a military user may input data instructing a unit to move and to engage an enemy target. The constructive simulation determines the speed of movement, the effect of the engagement with the enemy, and any battle damage that may occur" [6].

It should be noted that "live, virtual, and constructive simulations always include a real or [simulated] person in the simulation, as contrasted with a science-based simulation which typically models a phenomenon or process only" [6]. Table 2.1 summarizes the nature of the people and systems involved in live, virtual, and constructive simulation. Figure 2.5 shows images with examples of live, virtual, and constructive simulation.

The term *LVC simulation* refers to a combination of live, virtual, and constructive simulation. LVC simulation is starting to become an emerging concept for training land force operations [17] and joint operations. In this report we will, as mentioned earlier, focus on virtual and constructive simulation used for experimentation and analysis.

Simulation type	**People**	**Systems**
Live	*Real*	*Real*
Virtual	*Real*	*Simulated*
Constructive	*Simulated*	*Simulated*

Table 2.1 The nature of the people and systems involved in live, virtual, and constructive simulation.

Figure 2.5 Examples of live (Torbjørn Kjosvold/Norwegian Armed Forces), virtual (FFI), and constructive simulation (VT MÄK).

2.5 Interactive simulation

Interactive simulation is defined as a simulation model that "requires human interaction during runtime" [6]. Figure 2.6 illustrates the basic components of a typical interactive simulation.

More specially, in a *human-in-the-loop* (HITL) simulation a human is always part of the simulation, and influences the outcome in such a way that it is difficult, if not impossible, to reproduce exactly. This type of simulation allows for the identification of problems and requirements that may not be easily identified without including humans in the simulation. Since humans are real-world systems, including humans in a simulation also has the benefit of automatically including the mental properties of the real world [18].

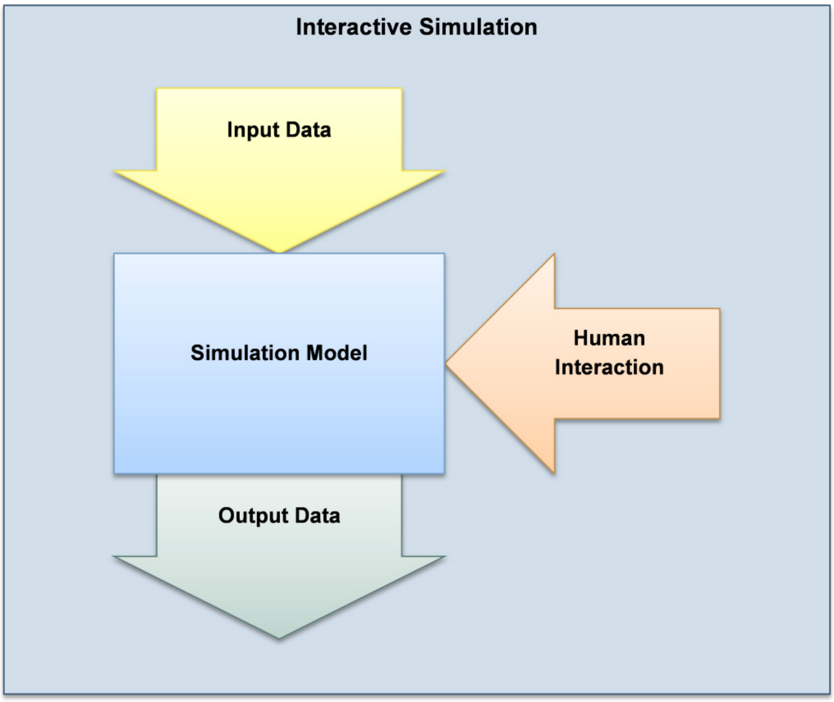


Figure 2.6 The basic components of a typical interactive simulation.

HITL simulation is mainly associated with virtual simulation in the LVC taxonomy, but constructive simulation may also require various degrees of human interaction, for example to control semi-automated forces (SAF). Interactive simulations with more than one participant typically require more than one computer, and are referred to as distributed interactive simulations.

2.6 Distributed simulation

A *distributed* simulation is a simulation that consists of multiple individual simulation components, which can be run on multiple computers connected through a network. The computers can be located in the same room or in geographically dispersed sites.

In a distributed combat simulation there is typically a need for sending state information, such as position and orientation, for the entities controlled by one computer, to all the other computers participating in the simulation. To ensure *interoperability* and data exchange between the different simulation components running on different computers in a distributed simulation, a set of standards has been created. Examples of such standards are Distributed Interactive Simulation (DIS) and High Level Architecture (HLA).

2.7 Interoperability and interoperability standards

Interoperability describes the ability of systems and devices to work together by exchanging data, and interpreting that shared data. Within M&S, interoperability can be described as "[t]he ability of a model or simulation to provide services to, and accept services from, other models and simulations, and to use the services so exchanged to enable them to operate effectively together" [6]. To achieve interoperability between models and simulations *interoperability standards* are needed.

Interoperability standards are standards that enable components to be plugged together for rapid assembly of systems. Examples of interoperability standards used in defence-related M&S are Distributed Interactive Simulation (DIS), High Level Architecture (HLA), Data Distribution Service (DDS), Military Scenario Definition Language (MSDL), and Coalition Battle Management Language (C-BML). Each of these standards is briefly described in the chapters 2.7.1 to 2.7.5.

The *Simulation Interoperability Standards Organization* (SISO) is an international organization dedicated to promotion of M&S interoperability and reuse, for the benefit of a broad range of M&S communities, including developers, procurers, and users world-wide. SISO's Standard Activity Committee develops and supports modelling and simulation standards, both independently and in conjunction with other organizations. SISO is recognized as a Standards Development Organization by NATO (North Atlantic Treaty Organization) and as a Standards Sponsor by IEEE (Institute of Electrical and Electronics Engineers) [19]. The NATO Modelling and Simulation Group (NMSG) is the Delegated Tasking Authority for M&S standards in NATO.

NMSG has been tasked with creating and maintaining the NATO Modelling and Simulation Standards Profile [20].

2.7.1 Distributed Interactive Simulation (DIS)

Distributed Interactive Simulation (DIS) [21][22] is an IEEE standard (IEEE Standard 1278) [23][24][25][26] for conducting real-time platform-level simulations across multiple computers. It was developed over a series of DIS Workshops at the Interactive Networked Simulation for Training Symposium, held by University of Central Florida's Institute for Simulation and Training. The standard itself is very closely patterned after the original Simulator Network (SIMNET) distributed interactive simulation protocol, developed in the 1980's.

In DIS, the simulation data are encoded in formatted messages, known as protocol data units (PDUs), and exchanged between hosts using existing transport layer protocols. The standardized part in DIS is the format of the messages. The DIS standard does not dictate how to send or receive those messages.

The latest version of DIS, DIS version 7 (DIS 7), was published in 2012. DIS version 7 is the first new version of DIS since 1998, and is a major upgrade which enhances extensibility and flexibility. DIS is considered as one of the predecessors of HLA, but is still widely used.

2.7.2 High Level Architecture (HLA)

High Level Architecture (HLA) [27] is a general purpose architecture, an IEEE standard (IEEE Standard 1516) [28][29][30], and a NATO STANAG (Standardization Agreement) (STANAG 4603) [31], for distributed simulations. The first versions of HLA were developed by the U.S. Department of Defense (DoD) in the 1990's, to provide a common framework to integrate and facilitate the interoperability and reuse of distributed simulations that run on a variety of different platforms.

An HLA compliant simulation component is called a *federate*, and a collection of interconnected federates is called a *federation*. The interaction between the federates is managed by a *Run-Time Infrastructure* (RTI). The RTI provides a set of general-purpose services for carrying out federate-to-federate interactions and functions for federation management. All interactions among the federates go through the RTI and are publish/subscribe based. The software and internal algorithms of the RTI however, are not defined by the HLA standard. The HLA runtime interface specification provides a standard interface for federates to interact with the RTI, to invoke the RTI services to support interactions among the federates, and to respond to requests from the RTI [7][12][27]. Figure 2.7 shows an illustration of the components of an HLA federation.

The types of data (objects, attributes, and interactions) that can be exchanged in a federation are defined in a *Federation Object Model* (FOM). A reference FOM is a standardized FOM for a specific purpose or domain. The Real-time Platform Reference (RPR) FOM is used as reference FOM in defence-related M&S. The latest version of the RPR FOM is version 2.0 [32].

Today there are three major versions of HLA, which are all widely used: HLA 1.3, HLA 1516-2000, and HLA 1516-2010 (HLA Evolved). The next update is scheduled for 2016 (HLA 2016/HLA Evolved Plus). This will be a smaller update that will include some new parameters and attributes on existing objects.

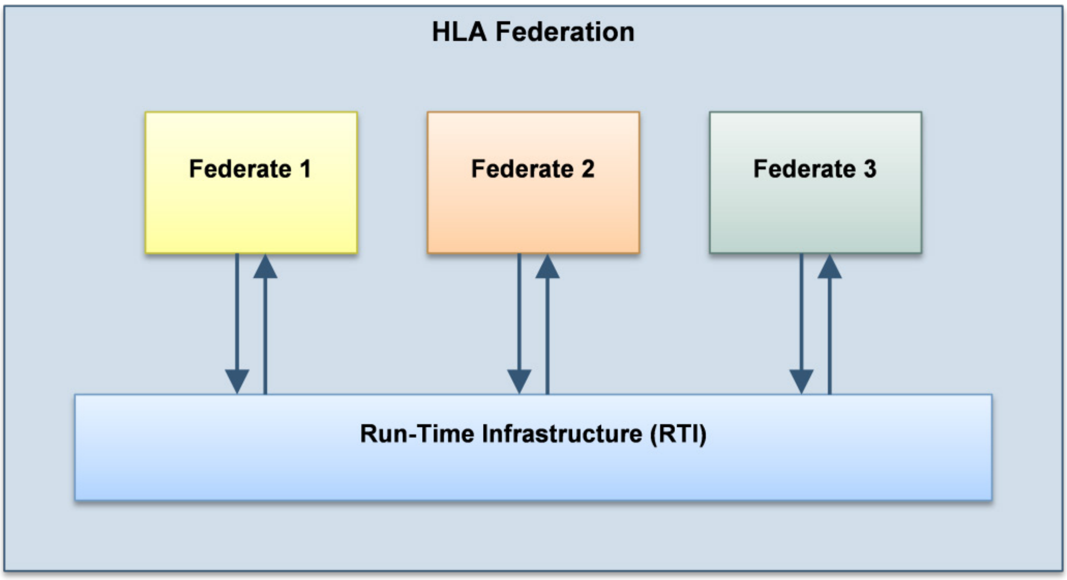


Figure 2.7 The components of an HLA federation (simulation).

2.7.3 Data Distribution Service (DDS)

The *Data Distribution Service* (DDS) [33] for real-time systems is an open middleware standard that aims to enable scalable, real-time, dependable, high-performance, and interoperable data exchanges in a publish/subscribe model for sending and receiving data, events, and commands among the nodes. The DDS specification was created by Real-Time Innovations and Thales Group, and is maintained by the Object Management Group (OMG).

DDS introduces a virtual *global data space* where applications can share information by simply reading and writing data-objects addressed by means of an application-defined name (topic) and a key. DDS features control of *quality of service* (QoS) parameters, including reliability, bandwidth, delivery deadlines, and resource limits. DDS also supports the construction of local object models on top of the global data space [34].

The latest version of DDS, DDS version 1.2, was released in 2007. A beta version of the next version, which will be version 1.4, was released in 2014. DDS and HLA shares many of the same characteristics, and a comparison of the two standards can be found in [35].

2.7.4 Military Scenario Definition Language (MSDL)

The *Military Scenario Definition Language* (MSDL) is a SISO standard (SISO-STD-007) [36] intended to provide a mechanism for loading military scenarios independent of the application generating or using the scenario. MSDL is defined utilizing an Extensible Markup Language

(XML) schema, thus enabling exchange of all or part of scenarios between command and control (C2) planning applications, simulations, and scenario development applications [19].

2.7.5 Coalition Battle Management Language (C-BML)

Coalition Battle Management Language (C-BML) is a SISO standard (SISO-STD-011) [37] for expressing and exchanging plans, orders, requests, and reports across C2 systems, live, virtual, and constructive M&S systems, and autonomous systems participating in coalition operations [19].

The development groups for MSDL and C-BML have been working closely together, to ensure that the two standards achieve full compatibility [38]. Recent activity has also begun towards merging MSDL and C-BML into a unified C2-to-Simulation (C2SIM) standard [39].

2.8 Fidelity, resolution, and scale

Fidelity, *resolution*, and *scale* are three primary descriptors applied to a model or simulation that serve as defining properties/characteristics of the model or simulation [11].

Fidelity is a term used to describe how closely the model or simulation matches the reality. A model or simulation that closely matches or behaves like the real system it is representing has a high fidelity [11]. Different applications might require different levels of fidelity [40].

Resolution is a term used to describe "[t]he degree of detail and precision used in the representation of real-world aspects in a model or simulation" [6]. The more details included in the simulation, the higher the resolution [11].

Scale or *level* are terms used to describe the size of the overall system the model or simulation represents. Logically, this means that the larger the system, the larger the scale or level of the simulation [11].

Due to the fact that the computer system running a simulation has a finite limit on the computing capacity, and each simulated entity requires a specific amount of computational power for a given level of resolution, a trade-off must be made between increased scale on one side and increased resolution on the other. If the number of entities is increased, the resolution must be decreased and vice versa. Figure 2.8 illustrates this issue. However, since the computational power is expected to increase every year, the maximum scale and the maximum resolution that can be simulated is also expected to increase.

Increasing the resolution of a model or simulation does not automatically increase the fidelity, but it always increases the complexity and thus makes the simulation computationally more expensive. Whether increasing the resolution of a model or simulation leads to increased fidelity depends on the validity of the additional details. Generally, it is desirable to maximize the fidelity, but at the same time try to keep the resolution as low as possible, by eliminating details that do not enhance the fidelity.

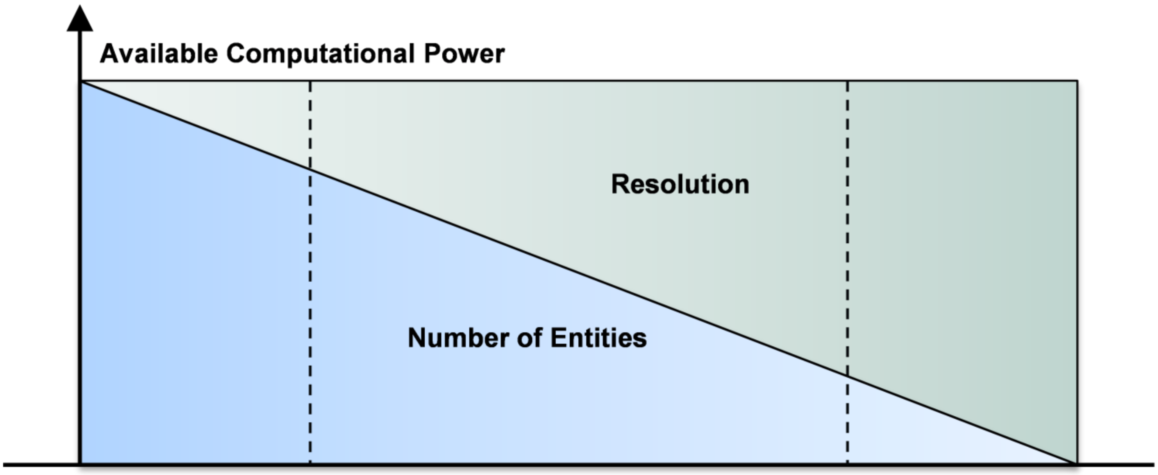


Figure 2.8 Since a computer system running a simulation has finite capacity, a trade-off must be made between increased number of entities (scale) on one side and increased resolution on the other.

2.9 Verification and validation (V&V)

For models or simulations to be useful, we must have confidence in their predictive ability and in the results. Such confidence can be obtained through a *verification* and *validation* (V&V) process.

Verification is "[t]he process of determining that a model or simulation implementation accurately represents the developer's conceptual description and specification" [6].

Validation is "[t]he process of determining the degree to which a model or simulation and its associated data are an accurate representation of the real world, from the perspective of the intended uses of the model" [6]. Figure 2.9 illustrates how V&V processes are applied in the modelling process [41].

More specially, *face validation* is "the process of determining whether a model or simulation based on performance seems reasonable to people knowledgeable about the system under study. The process does not review software code or logic, but rather reviews the inputs and outputs to assure that they appear realistic or representative" [6]. "Subject-matter experts (SMEs) are a hallmark of face validation, since they compare the simulation structure and output to their area of expertise in the real world" [42]. A step-by-step guide for conducting a face validation can be found in [42]. "While moving beyond face validation to more objective and quantitative methods should always be a goal, face validation is clearly preferable to no validation at all" [14].

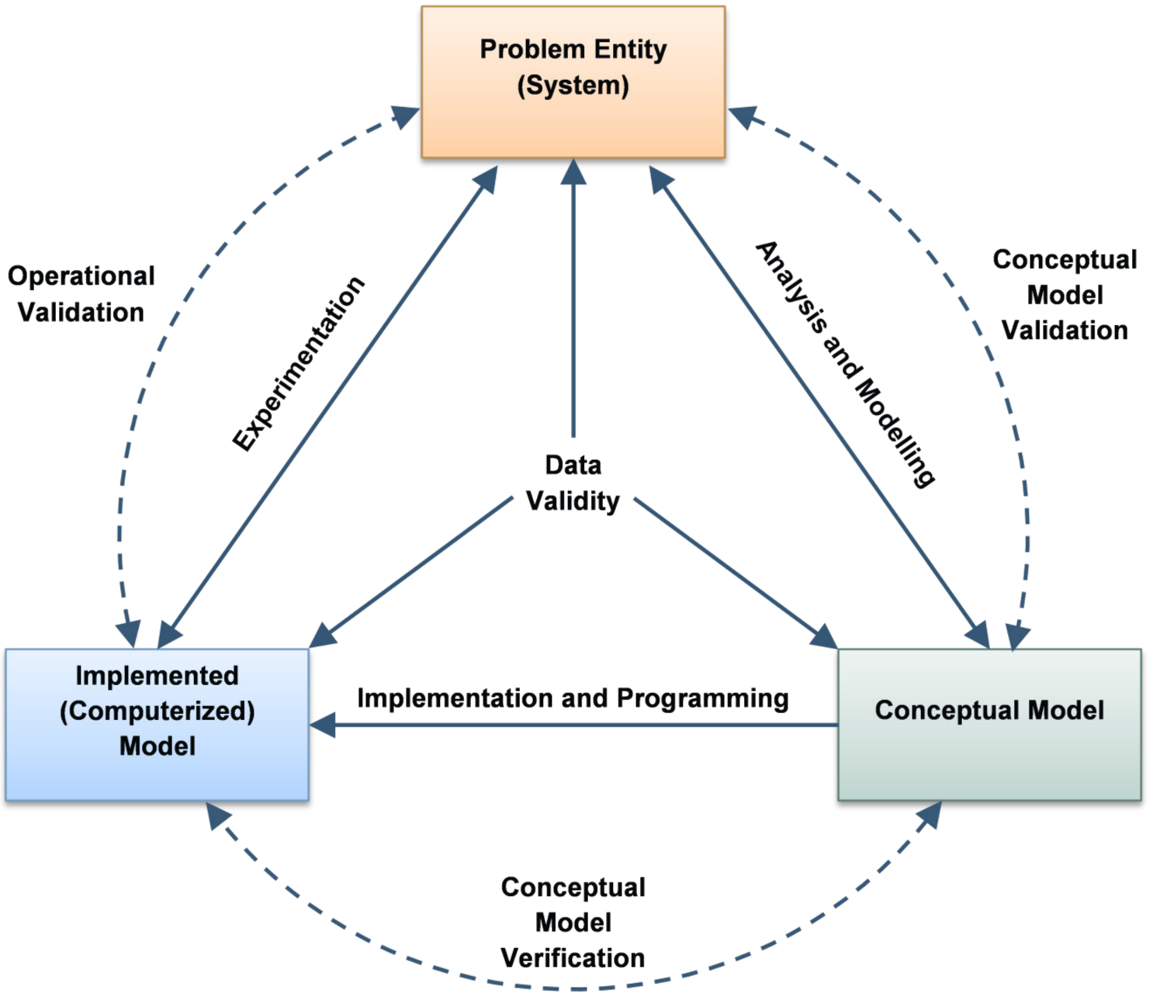


Figure 2.9 Verification and validation (V&V) processes applied in the modelling process ([41]).

The terms *accreditation* and *acceptance* are often used in conjunction with V&V. *Accreditation* is "the official certification that a model, simulation, or federation of models and simulations and its associated data are acceptable for use for a specific purpose" [6]. *Acceptance* is "the decision to use a simulation for a specific purpose" [6].

The *credibility* of a model or simulation can be understood as "a measure of how likely its results are to be considered acceptable for an application" [14].

V&V processes are conducted to avoid the following two main error categories regarding the use of M&S [10][14]:

1. *Valid* simulation results are *not accepted*.
2. *Non-valid* simulation results are *accepted*.

The NATO Modelling and Simulation Group (NMSG) and SISO have developed a generic methodology for V&V (GM-VV) to support acceptance of models, simulations, and data [43]. GM-VV has recently been accepted as a SISO standard [44][45][46].

3 Simulation of land force operations

Land force operations are very complex, and realistic simulation of brigade-size operations are very challenging. In this chapter we capture the state of the art in this domain, and summarize the different methods used and the most important challenges. More specially, we will capture the state of the art within virtual and constructive combat simulation. For a more comprehensive overview of combat modelling, the authors recommend the book "Engineering Principles of Combat Modeling and Distributed Simulation" edited by Professor Andreas Tolk [10].

3.1 Combat modelling

Combat has been modelled at a wide range of different scales and resolutions, and using almost every modelling paradigm and architecture available [7]. Based on observations of the nature of combat, the British defence analyst and former officer in the British Army, Jim Storr summarizes a model of combat like this (in his book "The Human Face of War" [5]):

- *Overall, the numbers of elements involved, and the interactions between them, make any attempt at detailed prediction meaningless.*
- *Each side consists of a nested set of systems. Each has a command and control (C2) node and a number of subordinates. The C2 node makes decisions and, simplistically, orders subordinates to move and to fight.*
- *At each level, information, orders and logistics are inputs. Outputs include casualties inflicted and sustained; ground gained and lost, and enemy positions captured. Additionally, firepower can be injected from elsewhere.*
- *Combat is adversarial: the outputs from one element become inputs to an enemy element. Casualties inflicted by one side are sustained by the other. Positions captured by one side are lost by the other, and so on. This process is a many-to-many relationship. A defending platoon may inflict casualties on a number of attacking platoons, and an attacking platoon may capture positions belonging to a number of enemy platoons in one attack. Thus the interrelationships between opposing forces are intensely complex. They will seem to be utterly confusing to an observer on the battlefield.*
- *The behaviour of each element at each level is moderately predictable. However, the overall performance of two forces in a battle or engagement is not. Nonetheless, the fact that at each level the behaviour is at least to some extent predictable does allow some control* [5].

Combat modelling can be described as the activity of purposefully abstracting and simplifying combat entities, their behaviour, activities, and interrelations to answer defence-related research questions [10]. The core activities of the combat units or fighting elements on every battlefield, which need to be modelled in a combat simulation, are: *moving*, *observing/sensing*, *shooting/engaging*, and *communicating* [10]. Depending on the resolution of the simulation, the combat units can be either single *entities* or *aggregated units* (see Chapter 3.3). These core activities are performed in the combat units' *situated environment*, which also needs to be modelled. Figure 3.1 illustrates the core components of combat modelling.

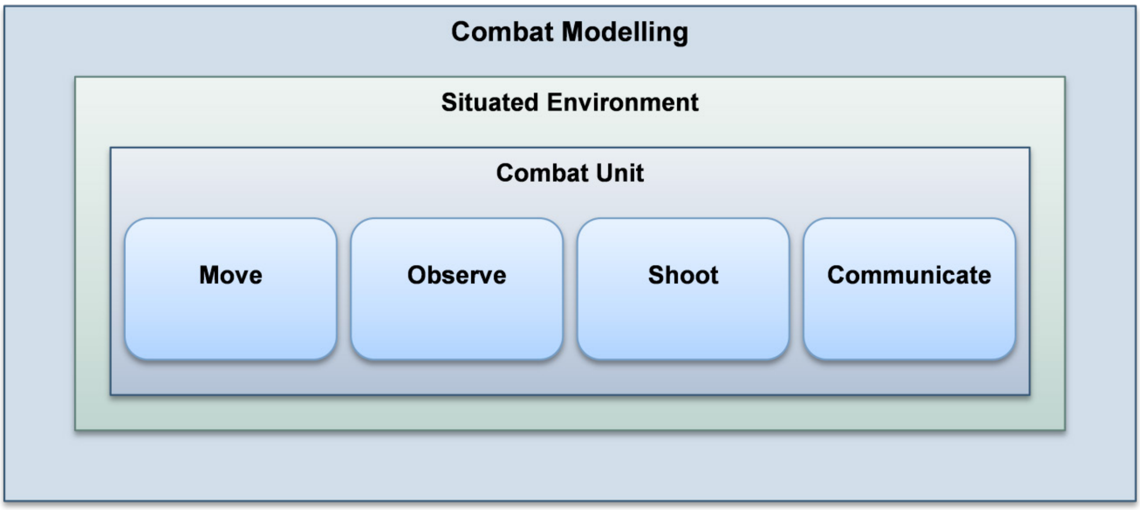


Figure 3.1 The core components of combat modelling ([10]).

The *synthetic natural environment* (SNE) *conceptual reference model* [47] outlines the interactions between the military system representations and the representation of the environment (often referred to as the *synthetic natural environment*). Figure 3.2 shows the SNE conceptual reference model.

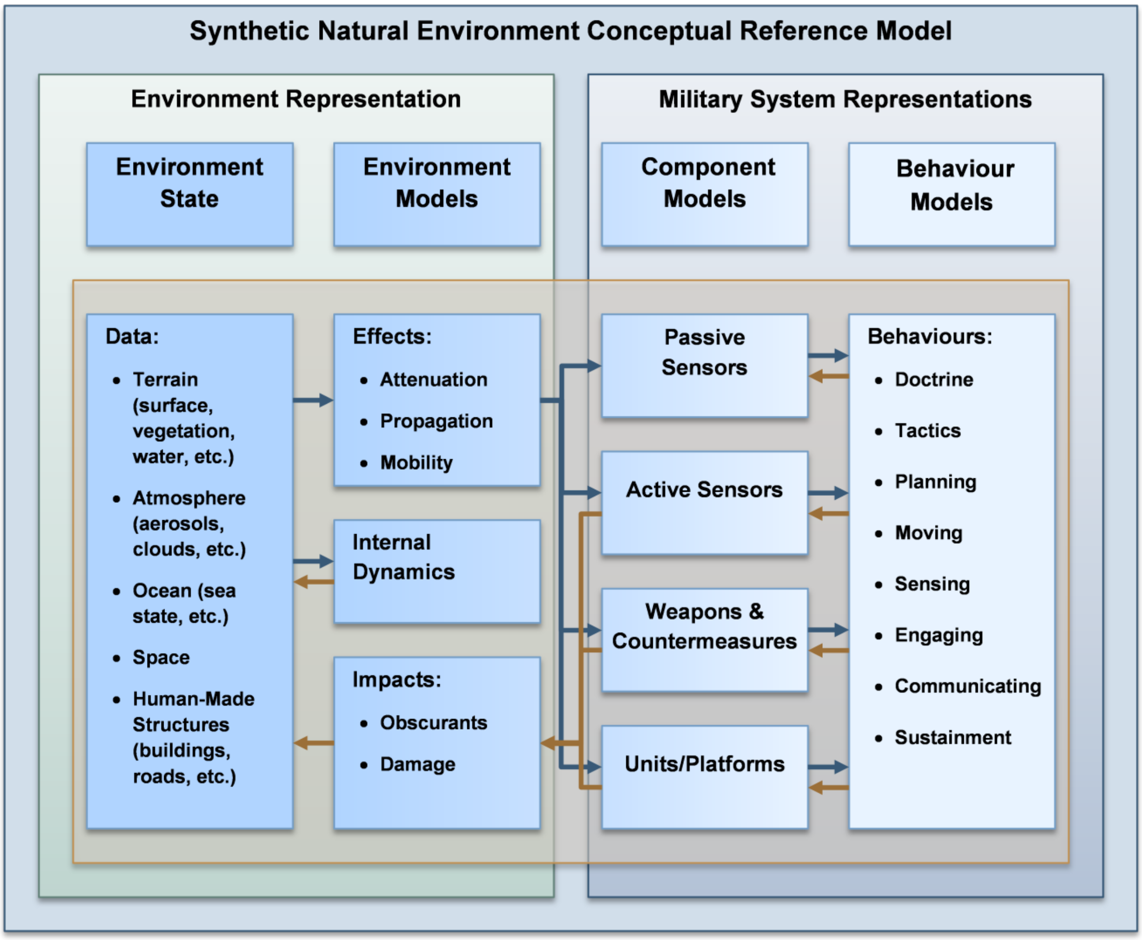


Figure 3.2 The synthetic natural environment (SNE) conceptual reference model ([47]).

What needs to be included in a combat model depends on the purpose of the simulation. Generally, all factors that can potentially affect the result of the military operation being simulated should be included. On the other hand, each element in a military operation can be described to an almost non-ending degree of detail. However, introducing more and more factors, and the relations between them, will cause the complexity of the model to increase exponentially [10]. In practice, time and resource constraints often limits what can be modelled. It is therefore always essential that the model's limitations and shortcomings are clearly described as a part of the simulation results. An important principle is that the level of fidelity should be balanced throughout the model. This will reduce the risk of introducing systematic biases in the combat model.

3.1.1 Land force components

This survey focuses on modelling and simulation of land force operations. Land force operations include a wide range of vehicles, weapon systems, and sensors. The command and control structure, and hierarchical organisation of the units, is captured in the order of battle (OOB) of the force. Examples of types of operations that may be included in a land force combat scenario are: manoeuvring, attacking, defending, delaying, withdrawing, receiving, and disengaging [10].

The units that may need to be represented in a combat model for land force operations usually fall into one of the following categories [10]:

- Combat or manoeuvre units (main battle tanks (MBTs), infantry fighting vehicles (IFVs), armoured personnel carriers (APCs), unmanned ground vehicles (UGVs), infantry, etc.)
- Fire support units (artillery, missile units, close air support (CAS), etc.)
- Combat engineering units (obstacles and mine warfare)
- Air defence units (air defence launchers, radars, etc.)
- Aviation units (fixed wing aircrafts, helicopters, unmanned aerial vehicles (UAVs), etc.)
- Command and control (C2) units (headquarters)
- Intelligence, surveillance, target acquisition, and reconnaissance (ISTAR) units (sensors and facilities)
- Communications and networks (infrastructure and systems)
- Logistics and supply units (transportation units and facilities)
- Medical units (in-field support and facilities)
- Maintenance units (in-field support and facilities)
- Electronic warfare (EW) units
- Chemical, biological, radiological, and nuclear (CBRN) defence units

For the combat model to be balanced, all units should be represented with about the same level of fidelity. It is, of course, also important that the units are represented with the same level of fidelity on both the *blue* (friendly) and the *red* (opposing) side. In addition it is often necessary to include paramilitary forces, insurgents, and civilian units in the combat model.

3.2 Virtual and constructive simulation

Virtual simulation involves real people operating simulated systems (see Chapter 2.4). The spectrum of virtual simulation systems range from expensive, high-end simulators to low-cost desktop simulators based on game technology.

The high-end combat simulators are designed to replicate the operating environment of the warfighters as closely as possible, and are most commonly used to simulate aircraft, helicopters, and combat vehicles. These simulator systems include replicas of cockpits or vehicle cabins (which may be mounted on hydraulic platforms for movement), image generators, and advanced display systems (often forming a dome for 360-degree field-of-view). They are mainly used to train operating skills of pilots and combat vehicle crews. Figure 3.3 shows examples of high-end virtual combat vehicle simulators.

It is naturally more problematic to fully replicate the operating environment of dismounted soldiers in a virtual simulator, since dismounted soldiers are using their own body to move around. Virtual simulators for dismounted soldiers have consequently not yet reached the same level of fidelity as the state-of-the-art aircraft and vehicle simulators [48][49][50][51][52]. Typically the current solutions for virtual dismounted soldier simulators use virtual reality (VR) head-mounted displays (HMDs) and full-body motion tracking systems [52][53][54][55]. Some solutions also use different types of omnidirectional treadmills (ODTs) [50] or human-sized hamster balls. Figure 3.4 shows an example of a body tracking system (to the left), and a virtual dismounted soldier simulator using an omnidirectional treadmill (to the right).

In the other end of the virtual simulation spectrum we find the low-cost desktop simulators based on first-person shooter (FPS) game technology. These simulation systems do not replicate the operational environment of the warfighters down to the details needed for training vehicle or weapon operating skills. Instead they are focused on training tactics, communication skills, decision-making, and how to think (cognitive training). They can be controlled by mouse and keyboard, or more advanced game controllers like gamepads, joysticks, or steering wheels and pedals.

Figure 3.3 Examples of high-end virtual combat vehicle simulators (U.S. Army).

Figure 3.4 Example of a body tracking system (to the left), and a virtual dismounted soldier simulator using an omnidirectional treadmill (to the right) (Interservice/Industry Training, Simulation and Education Conference).

In the recent years we have also seen an increase in the availability of low-cost virtual reality (VR) HMD devices, for example Oculus Rift from Oculus VR. These display devices combine head-tracking with full stereoscopic view to increase the sense of immersion in the three-dimensional virtual environments [56].

Examples of low-cost desktop virtual simulation systems are Steel Beasts from eSim Games (see Chapter 4.6) and Virtual Battlespace (VBS) from Bohemia Interactive Simulations (BISim) (see Chapter 4.7). Figure 3.5 shows examples of this type of simulation system. At FFI we have been using VBS for analysis and experimentation since 2008.

The clear distinction between high-end legacy simulators and low-cost simulators are now starting to fade, and we are seeing an increased use of low-cost simulation systems based on game technology as components in high-end legacy simulators [57]. For example, the newly introduced VBS IG (Image Generator) makes it possible to build advanced simulators, with integrated physical mock-ups, based on VBS.

Figure 3.5 Examples of low-cost desktop virtual simulation systems (FFI, Bohemia Interactive Simulations).

Virtual simulations can have very high fidelity, but since all simulated systems are operated by real humans, and therefore require at least one computer, there are limitations on how large operations that can be simulated [58]. Generally, virtual simulations are limited to a few hundred participants, and can be used to simulate squad, platoon, and company size operations. For simulating larger operations, constructive simulations are used.

Simulations can also include both virtual and constructive units. However, when virtual and constructive simulation models are combined, difference in resolution between them can lead to *fair-fight* issues (see Chapter 3.5).

Constructive simulation involves simulated people operating simulated systems (see Chapter 2.4), and includes both simulation systems with fully automated computer-generated forces (CGF) and simulation systems with semi-automated forces (SAF). In the latter case real people give input to the simulation in the form of orders to the simulated units. This means that one person can control several units. In constructive simulation the units must have some sort of smart behaviour or artificial intelligence (AI) (see Chapter 3.7).

Constructive simulation can be used to simulate operations of all sizes, including theatre level, and large constructive simulations can include millions of entities [59]. Application areas are mainly command and staff training, computer-assisted exercises (CAX), and wargaming for analysis and experimentation. Generally, the fidelity in constructive simulations will be lower than in virtual simulations.

Examples of constructive simulation systems are Joint Theater Level Simulation (JTLS) from ROLANDS & ASSOCIATES Corporation (see Chapter 4.3), One Semi-Automated Forces (OneSAF) from U.S. Army (see Chapter 4.5), and VR-Forces from VT MÄK (see Chapter 4.8). Figure 3.6 shows examples of this type of simulation systems. Constructive simulations can use either *entity-level* or *aggregate-level* simulation models (see Chapter 3.3). Virtual simulations are, of course, inherently based on entity-level simulation models.

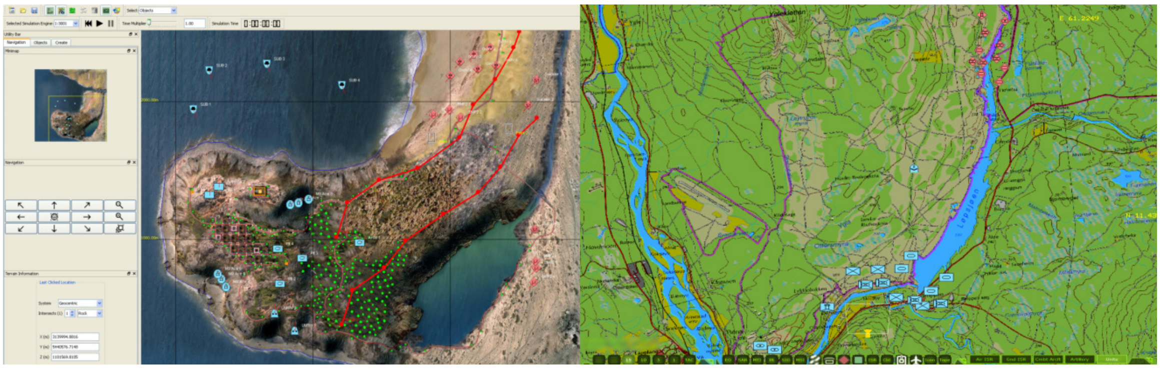


Figure 3.6 Examples of constructive simulation systems (VT MÄK, FFI).

In simulation for training, the choice between virtual and constructive simulation depends on the training audience and the training goals. In simulation for analysis and experimentation, the desired fidelity, the resources available, and the size of the scenario that needs to be simulated are important factors that must be considered when making this choice. Virtual simulations generally have a higher fidelity than constructive simulations, but virtual simulations are also much more people-intensive, and thus much more expensive. Current technology also limits the number of entities in a virtual simulation to a few hundreds at most.

We have generally used virtual simulation in experiments where human system operators are essential, for example when experimenting with technology that directly affects human performance or how the humans operate (low-level tactics). The size of these experiments has been limited to a few platoons (reduced company level). An example of this type of simulation is provided in Chapter 5.2.

To simulate operations at the battalion and brigade level we have used constructive simulation for conducting interactive war games with SAF. An example of this type of simulation is provided in Chapter 5.3.

3.3 Entity-level and aggregate-level models

Entity-level combat models have high resolution, and represent individual vehicles, platforms, and personnel as distinct entities. State information is maintained separately for each entity, and the four core activities of combat (moving, observing/sensing, shooting/engaging, and communicating) are modelled at the level of the individual entities. These models are usually based on performance data for the specific entity types [60].

Aggregate[1]*-level* combat models (often also referred to as *unit-level* combat models) have lower resolution, but provide the capability to simulate larger operations. In these models the military objects are represented as aggregated units with a given size (e.g. a company, a battalion, or a brigade). The individual entities however, are not represented. State information is maintained for the unit as a whole, and is often computed based on statistical analysis and attrition models like the Lanchester models [61] (see Chapter 3.6.2) [60].

Entity-level combat models are much more complex, and computationally much more expensive, than aggregate-level models. With today's computing capabilities it should be possible to simulate operations of brigade-level size and below using entity-level models. However, the size of the operations that can be simulated with entity-level models is expected to the increase, as computational power is expected to increase in the future.

Entity-level models have higher resolution and thus the potential to achieve higher fidelity than aggregate-level models. It is also easier to see what is going on in an entity-level simulation, and this also makes them more accessible for face validation. Nevertheless, current entity-level

---

[1] An aggregate is a collection of items that are gathered together to form a total quantity.

models tend to produce attrition levels that are higher than those observed historically [60][62]. "Possible phenomena present in actual combat, and accounted for in [the parameters of aggregate-level attrition models (such as the Lanchester models)] but not [in the] entity-level combat models, that could explain this, include target duplication, shooter non-participation, suppression effects, self-preservation, and suboptimal use of weapons and targeting systems" [62]. In other words, current constructive entity-level combat models lack good representations of the human aspects of combat and *combat friction* (see Chapter 3.9), resulting in that the simulated operations tend to run smoother than they would in the real world.

To simulate large operations, and at the same time have high resolution in particular areas of interest in the battlefield, it is possible to create multi-resolution combat models by linking entity-level and aggregate-level combat models. There are however a lot of challenges regarding interaction between entities and aggregated units [60][63]. One solution that avoids such *inter-level* or *cross-level* interaction is to have designated areas in the battlefield where all the units are either aggregated units or individual entities. When the units move between these areas they have to be *aggregated* or *disaggregated,* and transferred from one resolution level to another at runtime. Figure 3.7 illustrates this concept. To further increase the fidelity in parts of the simulation, virtual entities can be included in the disaggregated area in the battlefield.

An example of a multi-resolution combat model is the NATO Training Federation (NTF), which (at some point) has included JTLS (which is a constructive, aggregate-level simulation system), Joint Conflict and Tactical Simulation (JCATS) (which is a constructive, entity-level simulation system), and VBS2 (which is a virtual simulation system) [64]. Figure 3.8 illustrates NTF.

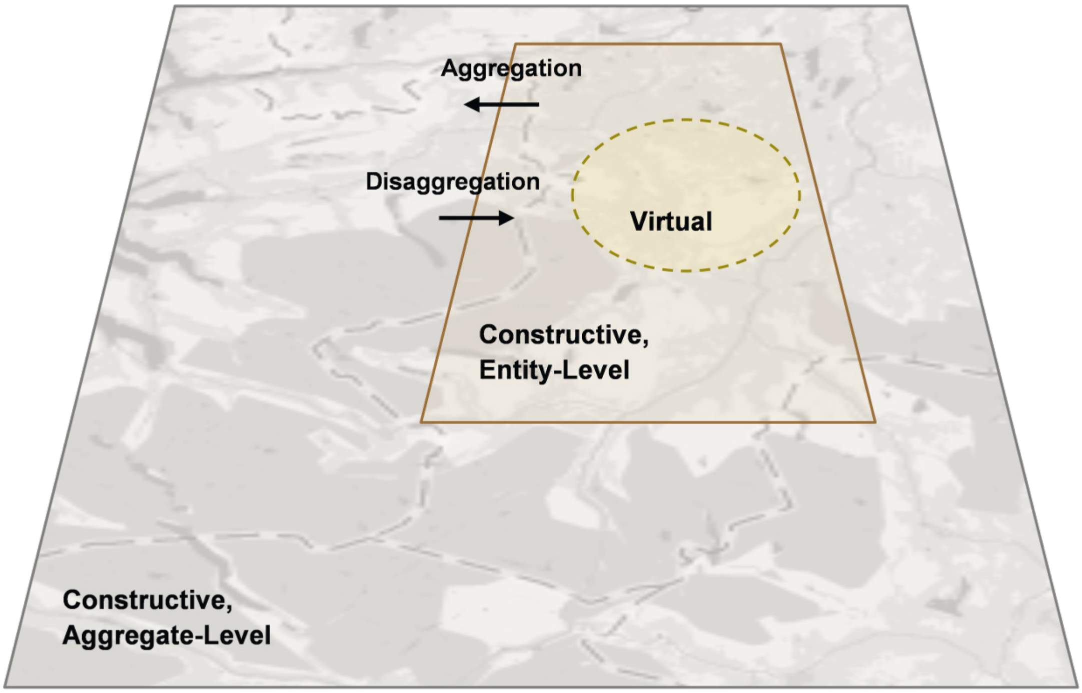


Figure 3.7 Example of multi-resolution combat model with designated areas for aggregate-level and entity-level models.

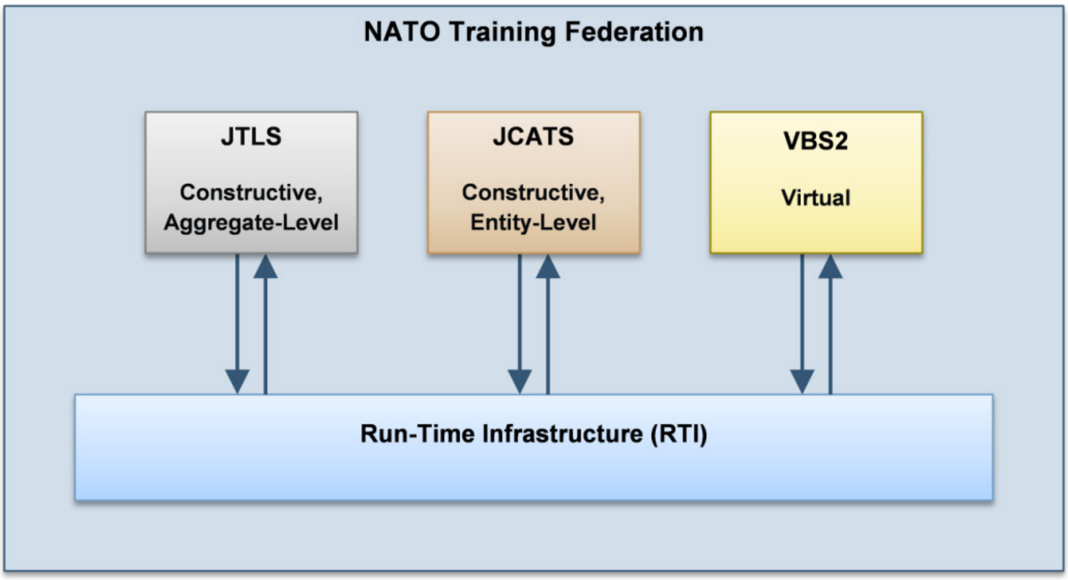


Figure 3.8 The NATO Training Federation (NTF).

3.4 Modelling the environment

Combat scenarios take place in a *situated environment*. A model of this environment (often referred to as a *synthetic natural environment* (SNE) or *virtual environment*) includes the terrain with lakes, seas, rivers, and vegetation. It also includes static human-built structures like buildings, roads, and bridges. Another important component of the environment is the weather. Modern military simulation systems allows for agile and dynamic representations of the environment, where the terrain characteristics may change due to weather and explosions, and varying light conditions may influence sensors. Often the simulation systems include a *physics engine* [65][66][67], providing realistic simulation of physical systems in the environment model. Figure 3.9 shows examples of components of the environment. Many of these components directly affect how combat units move, observe, engage, and communicate. The collection of data forming an environment model is often referred to as an *environment database* or *terrain database*. SEDRIS[2] is a non-proprietary infrastructure technology, and a STANAG (STANAG 4662, 4663, and 4664) [68][69][70], for representation and interchange of environmental data [71].

The resolution in which the environment needs to be modelled depends on the resolution of the combat units. Virtual simulations, where the system operator looks directly into the virtual environment, have the highest resolution requirements. Today environments for virtual simulations can be modelled with resolutions approaching photorealistic quality. However, creating such detailed environment models is labour intensive, and they consist of substantial amounts of data. This limits the size of the areas that can be modelled with very high resolution. Figure 3.10 shows examples of environment models with high resolution.

---

[2] SEDRIS used to be an acronym for *Synthetic Environment Data Representation and Interchange Specification*, but it is now used as a noun.

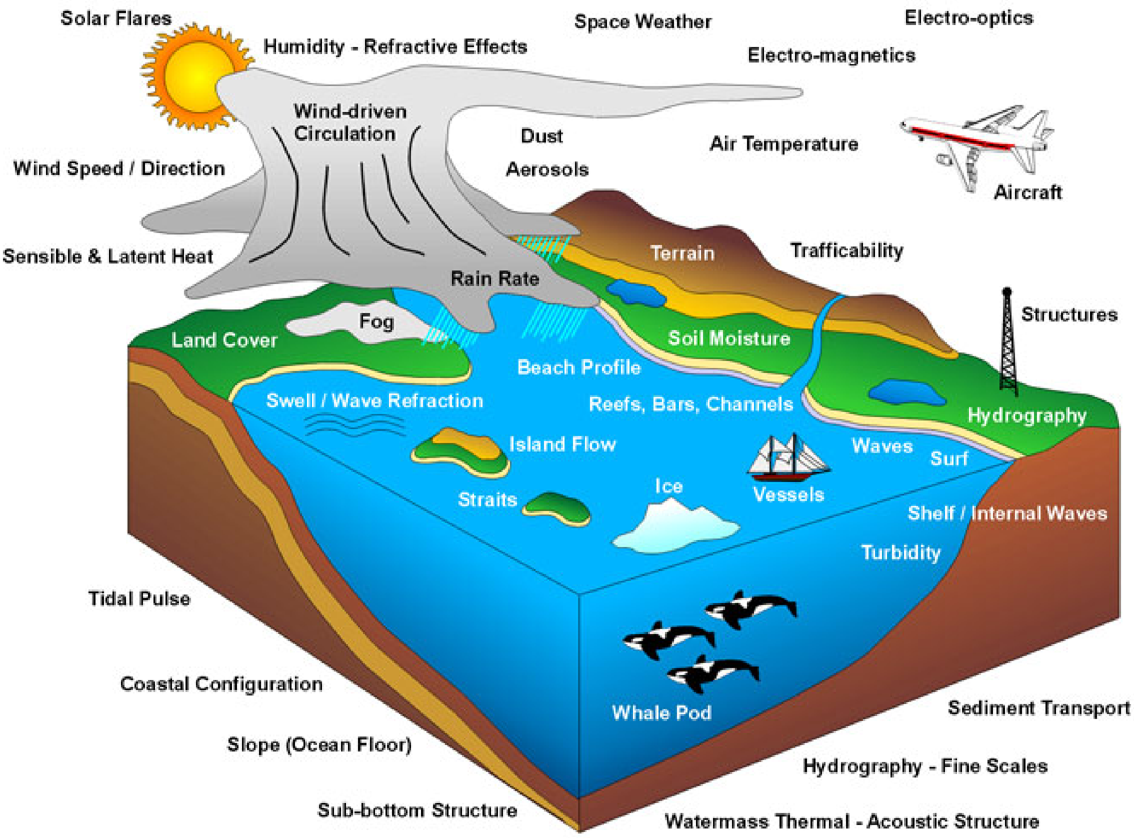


Figure 3.9 Examples of components of the environment (SEDRIS [71]).

Figure 3.10 Examples of environment models with high resolution used in virtual simulations (Bohemia Interactive Simulations).

Constructive simulations are usually monitored, and interacted with, through two- or three-dimensional map views, so they generally have lower requirements for visual resolution. When different simulation systems are plugged together, their representations of the environment should be correlated to avoid strange effects (e.g. ground vehicles floating in the air due to elevation mismatch) and *fair-fight* issues (see Chapter 3.5).

The required size of the modelled environment depends on the size of the simulated scenario. A virtual simulation of an operation performed by a dismounted infantry squad may require only an environment model of a few square kilometres. On the other hand, theatre-level scenarios may require whole continents to be modelled. When modelling environments, a trade-off usually has to be made between resolution on one side and size on the other. To simulate large operations, and at the same time have high-resolution environment models in particular areas of interest in the battlefield, it is possible to create multi-resolution environment models. For example if multi-resolution combat models are used, it is appropriate to use corresponding multi-resolution environment models.

A computer has a limited amount of memory, and environment models can be very large and memory intensive. Environment models are therefore often divided into smaller *tiles*. Only the tiles that are needed by the simulation are loaded into the computer's memory, and the tiles that are no longer needed are released [72]. This functionality is often referred to as *terrain paging*. As we will come back to in Chapter 3.4.5, the tiles can also be streamed from a server.

An environment model can either be *geospecific*, which means that it is representing an actual real-world location, or *geotypical*, which means that it is generated from fictitious environmental data representing what is typical in an area.

The NATO Modelling and Simulation Group (NMSG) has created a whole new virtual continent called *Missionland*, that can be used for simulation exercises. Missionland is located in the middle of the North Atlantic Ocean, and has a size of about 2,000 x 2,000 kilometres. The following climate zones have been defined in Missionland: arctic, temperate, arid, and tropical. Furthermore, the following elevation profiles have been defined: flat, hilly, mountainous, and cliff/fjord [73]. Figure 3.11 shows the location of Missionland.

There are several professional software packages available for generation of environment models. Some examples are ArcGIS from Esri, Global Mapper from Blue Marble Geographics, TerraTools from TerraSim, and Terra Vista form Presagis.

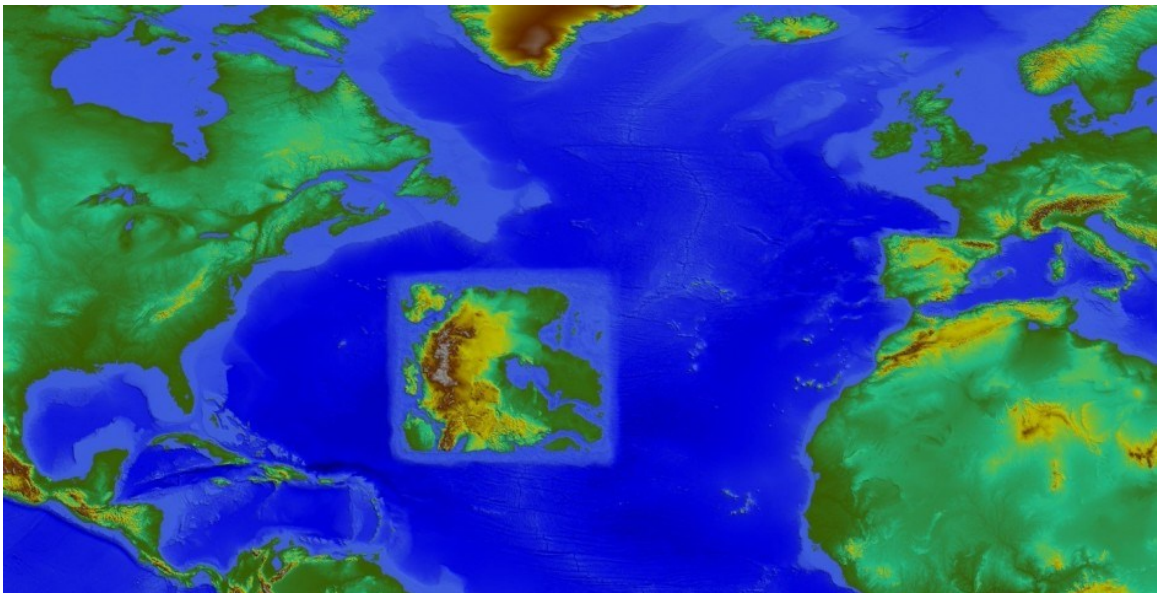


Figure 3.11 The virtual continent Missionland is located in the middle of the North Atlantic Ocean.

3.4.1 Terrain

In entity-level combat models the terrain is usually represented by a large three-dimensional surface tessellated into a polygonal mesh. The polygonal mesh can form a regular grid, or be an unstructured grid. The most advanced terrain models also support subterranean features such as overhangs, caves, and tunnels.

Aggregate-level combat models, especially older ones, often instead use terrain models where the battlefield is tessellated into a regular grid of quadratic or hexagonal areas, where the characteristics, such as elevation and surface type, are homogenous.

A *digital elevation model* (DEM) is commonly stored as a matrix of regularly spaced elevation data points (raster data). A limitation to two-dimensional raster data is that they lack the possibility to represent subterranean features. It is also difficult to capture steep vertical features with raster data [72].

One of the elevation data standards that traditionally often has been used in military simulation systems is *Digital Terrain Elevation Data* (DTED) [10]. The DTED standard defines three resolution levels, Level 0, Level 1, and Level 2, which respectively have a resolution of approximately 900, 90, and 30 meters between the elevation points [74]. Virtual simulation systems however, often require a higher resolution than this. For constructive simulation systems the required terrain resolution depends on if the CGF has a tactical AI that is advanced enough to be able to exploit the terrain or not. Other commonly used DEM standards are the *Geographic Tagged Image File Format* (GeoTIFF) and the *American Standard Code for Information Interchange Grid* (ASCII Grid).

Elevation data can also be stored in a *triangular irregular network* (TIN) format. A TIN is a series of triangles that forms a minimal set of polygons that accurately represents the shape of the terrain. TINs can also be used to represent steep vertical features and subterranean features [72].

To look realistic, terrain models are covered by a texture layer representing the land-cover materials. The texture layer can be generated from satellite imagery and aerial photos, or material maps and generic land-cover material images.

The photo-specific approach is considered to create the most realistic imagery, and the best correlation with the real world. However, one problem with this approach is that the imagery contains artefacts such as cloud cover, shadows, and potentially the presence of dynamic things like vehicles. Minimizing, or if possible removing, these artefacts require significant image processing time. In addition, high-resolution photo-specific terrain textures are very memory intensive [75].

The material-map approach uses a texture library of imagery of different land-cover materials like soil, gravel, rock, grass, and asphalt. The most significant problem with this approach is that land-cover material data are often not available at the desired resolution [75]. It is also possible to combine these two approaches [76]. Figure 3.12 shows the components of a terrain model.

In the real world the terrain may be significantly changed during a combat scenario, due to craters formed by explosions, or engineering units creating ditches, defilades, or trenches. A terrain model which can be modified at run-time during a simulation is referred to as a *dynamic terrain* model. (Dynamic terrain models are sometimes also referred to as *destructible terrain* models or *deformable terrain* models.) Several of the current simulation systems support dynamic terrain, and the most advanced models use detailed physics-based approaches [67][77]. Furthermore, much research is focused on how to get correlated representations of dynamic terrain in federated simulation systems [78].

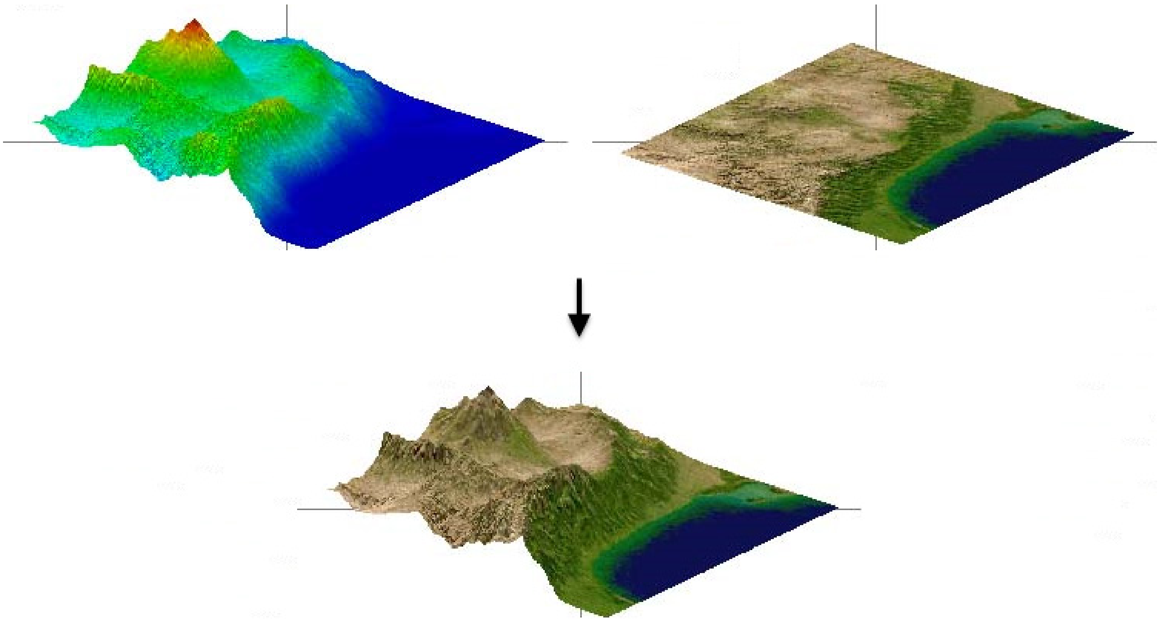


Figure 3.12 A terrain model consists of an elevation model and a land-cover material texture.

3.4.2 Vegetation, rivers, lakes, and seas

To further increase realism, environment models are populated with vegetation, and rivers, lakes, and seas are filled with simulated water. Vegetation has an impact on the movement of units, and the intervisibility between units. As shown in Figure 3.10, foliage in virtual simulations can be modelled at a very high level of detail, with individual plants, bushes, and trees.

In most simulation systems flowing and non-flowing water bodies restrict the movement of ground combat units. Furthermore, in high-fidelity virtual simulations creeks and rivers, and ponds, lakes, and seas, can be simulated using detailed physics-based models of water [79].

The positions of trees and bushes, geographical features such as rivers and lakes, and human-made structures such as road networks and building footprints, are usually stored as *vector data*. Vector data can be points, lines, or polygons. Points can represent trees, cell phone towers, or other points of interest. Lines can represent road networks, fences, streams, tunnels, bridges, power lines, or dams. Polygons can represent lakes, rivers, forests, soil types, populated areas, or building footprints. A commonly used vector data format is the Environmental Systems Research Institute (ESRI) *Shapefile* format [80]. Positions of geographical features can also be extracted from satellite imagery and aerial photos [81][82].

In federated simulation systems the lack of correlated representations of vegetation and water bodies may lead to fair-fight issues.

A forest can typically have tree densities of several thousand trees per hectare, which means several hundred thousands of trees per square kilometre. An environment model can therefore include millions of trees. For large environment models it is therefore not feasible to store the individual positions for each of the trees as part of the environment model. Instead vegetation can be procedurally generated (see Chapter 7.1.4) by the simulation system at run-time using a set of rules based upon terrain shape and land-cover material data. Using this approach the placement of the vegetation will be completely deterministic [83]. If different simulation systems in a federation use the same rule set for generation of vegetation, and their terrain models are correlated, fair-fight issues can be avoided.

An example of procedural generation of vegetation is the technology for procedural *biotopes*[3] in VBS [83]. Figure 3.13 shows examples of procedural generation of vegetation in VBS by defining two different biotopes ("Light Jungle" and "Heavy Temperate").

Biotopes can be loaded in real-time and switched between on-the-fly, allowing for sweeping environment changes without the need to generate an entirely new terrain

Biotopes can be loaded in real-time and switched between on-the-fly, allowing for sweeping environment changes without the need to generate an entirely new terrain

Biotopes are configurable through a vast range of options, allowing them to be tweaked and tailored to give the precise outcome desired

Biotopes are configurable through a vast range of options, allowing them to be tweaked and tailored to give the precise outcome desired

Figure 3.13 Examples of procedural generation of vegetation in VBS by defining procedural biotopes for "Light Jungle" (to the left) and "Heavy Temperate" (to the right) (Bohemia Interactive Simulations).

---

[3] Biotope is an area that is uniform in environmental conditions and in its distribution of animal and plant life.

3.4.3 Human-made structures

Human-made structures like roads, runways, bridges, and buildings are important parts of environment models. Roads and bridges are, of course, critical for simulating movement and transportation, and buildings are crucial for simulating urban combat operations. When simulating urban combat operations it is also important to include models of ambient civilian life [84][85] and civilian traffic [86].

In aggregate-level combat models using hexagon-based environment models, roads are usually aligned at the boundaries between the hexagons, and urban environments are represented by defining entire hexagons as urban areas.

For detailed environment models, modern environment-generation tools can automatically generate roads [87] and buildings [88] based on vector data for road networks and building footprints. For high-fidelity virtual simulations with dismounted soldiers, buildings with multiple floors and interior can also be automatically generated [89]. Additionally, state-of-the-art simulation systems may include physics-based models for destruction of human-made structures [90][91]. Figure 3.14 shows examples of physic-based simulation of building damage in VBS.

Figure 3.14 Examples of physics-based simulation of building damage in VBS (Bohemia Interactive Simulations).

3.4.4 Weather

Another environmental factor that can significantly affect military operations is the weather [92]. Simulated weather conditions and weather effects should therefore be included in combat simulation. Weather conditions such as fog, clouds, and precipitation can, of course, greatly reduce visibility, but some weather conditions can also cause the terrain characteristics to change. For example, rain can cause slippery roads, and snow can reduce the speed of movement. In

Norway, where we have snow in the winter most places, simulated snow is of particular interest for simulated combat operations.

BISim has recently implemented support for simulated snow layers in VBS. The snow layers are procedurally generated at run-time, and affect the line of sight (LOS) and movement speed of units [83]. Figure 3.15 shows an environment model with (to the right) and without (to the left) procedurally generated snow layers in VBS.

Wind is another weather effect that should be modelled. Strong wind can cause material damage, limit the opportunity for air support and air transport, and have a negative impact on the precision of direct and indirect fire. Wind can also seriously reduce the effectiveness of smoke screens.

Again, to avoid fair-fight issues, it is important that the weather conditions and weather effects are correlated in federated simulation systems. Consistent weather conditions can be ensured by using central weather services, but the weather effects still need to be consistently modelled in all the participating simulation systems [10].

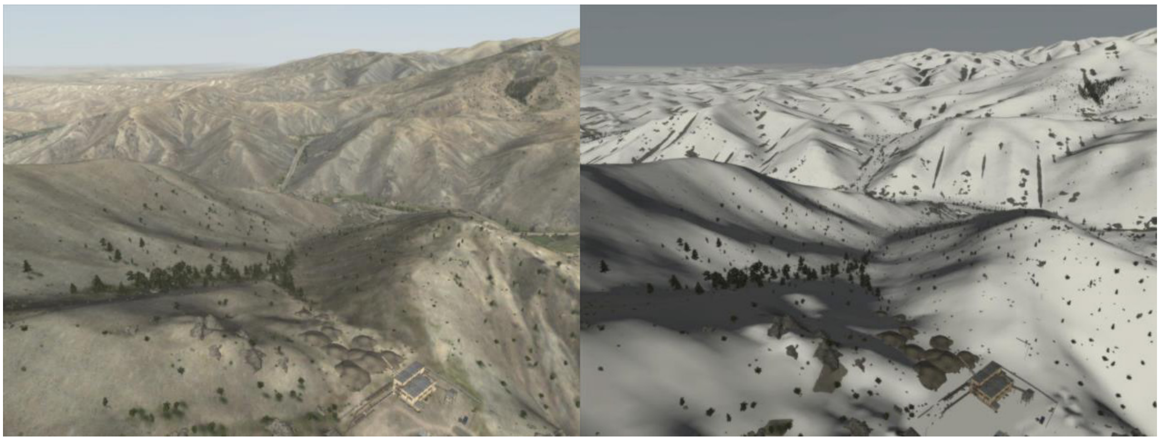


Figure 3.15 Environment model with (to the right) and without (to the left) procedurally generated snow layers in VBS (Bohemia Interactive Simulations [83]).

3.4.5 Environment services

Generating environment models are very time consuming, and large environment databases are very memory intensive. In addition, many simulation systems often use their own format for representing the environment. Depending on the approach taken within the terrain modelling process, different assumptions, simplifications, and abstractions may lead to very different implementations. For example, some simulation systems take the curvature of the earth into account, while others abstract this into a plane [10].

To simulate large joint operations there is an increasing demand for the ability to have environment models that include the whole earth. Environment models of the entire globe require huge amounts of memory, so it would not be feasible for each computer in a distributed simulation to store its own copy. For example, according to an article from 2012 published on the news website Mashable, a spokesperson from Google stated that: “Combining satellite, aerial, and street-level imagery, Google Maps has over 20 petabytes of data, which is equal to approximately 21 million gigabytes, or around 20,500 terabytes” [93].

A solution is to have a service-based approach where a central server streams environment tiles to the individual simulations [94]. Environment streaming also ensures correlated environment representations in different simulation systems. An example of an environment streaming service is the VR-TheWorld Server from VT MÄK. Figure 3.16 illustrates the working principle of an environment streaming service. It remains to be seen if environment streaming will be standardized, and support for environment streaming will be implemented in different simulation systems. A terrain streaming service must also be able to support dynamic terrain.

As we mentioned in Chapter 3.4.4, weather is another environment factor that can be provided as a service from a central server. The concept of providing M&S as a service (MSaaS) is an emerging trend that will be discussed further in Chapter 7.1.2.

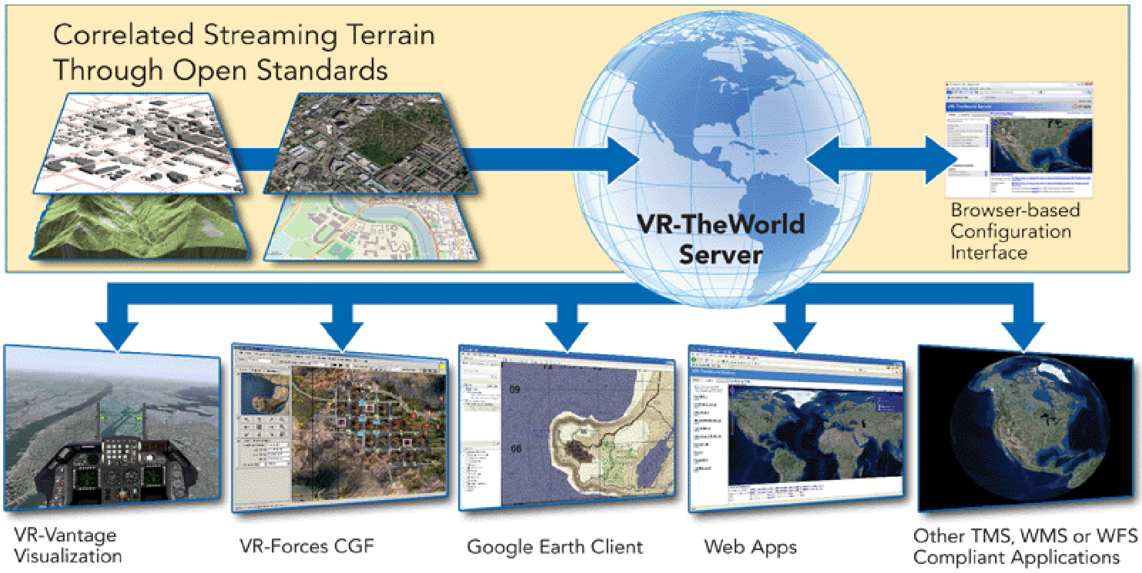


Figure 3.16 The VR-TheWorld Server from VT MÄK is an example of an environment streaming service (VT MÄK).

3.5 Fair fight

When federating two simulation systems it is essential to make sure that the environment is represented in both systems in a way that avoids giving a systematic advantage to one of the simulation systems [10]. In other words, it is important to provide a *fair fight* between units controlled by different simulation systems. Factors that can lead to fair-fight issues are uncorrelated representations of environment components such as terrain elevation, vegetation, buildings, and weather. In addition, different models for representing damage, movement, and sensors can lead to fair-fight issues. Also, when combining virtual and constructive forces fair-fight issues may occur.

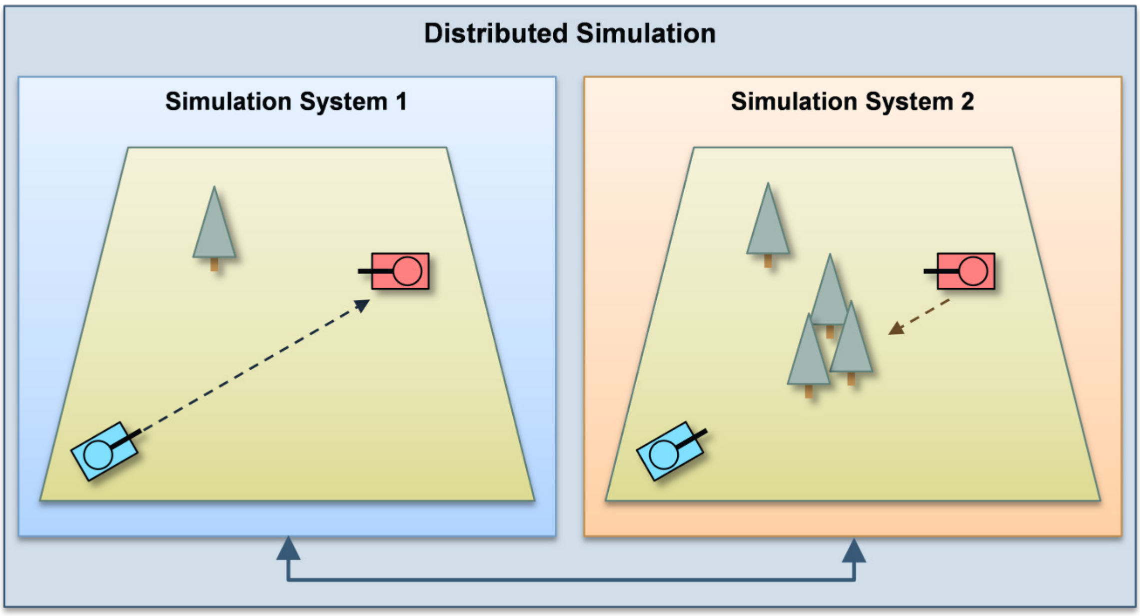


Figure 3.17 Example of a fair-fight problem in a distributed simulation consisting of two simulation systems with uncorrelated environment representations.

Figure 3.17 shows an example of a fair-fight problem in a distributed simulation consisting of two simulation systems with different environment representations. In this simulation the *blue tank* is controlled by *simulation system 1*, while the *red tank* is controlled by *simulation system 2*. The blue tank has line of sight (LOS) to the red tank in simulation system 1. However, in simulation system 2 vegetation is blocking the LOS between the two tanks. In this situation the blue tank obviously has a clear advantage, so this is not a fair fight.

While the fair-fight issue in this example is easy to identify, many real-world fair-fight issues can be far more subtle and complex, and much harder to trace [10]. Several methods and tools for automated environment database correlation testing have been developed over the past few years [95][96].

Using correlated environment representations will reduce fair-fight issues [97]. However, to completely eliminate all fair-fight problems in distributed simulations, every possible difference in the underlying simulation models used by the different simulation systems must be addressed [18][97][98].

3.6 Modelling combat units

As we described in Chapter 3.3, combat units can be modelled using *entity-level* or *aggregate-level* models. Whether units are modelled at entity-level or aggregate-level is decisive for how their core activities (moving, observing/sensing, shooting/engaging, and communicating) are modelled. Another decisive factor is whether the units are directly controlled by humans (i.e. virtual simulation) or not (i.e. constructive simulation). Furthermore, the modelling of the core activities is tightly connected to how the environment is modelled [10]. Generally, there should be a correspondence between the level of fidelity for the unit models and the environment model.

Figure 3.18 Example of a high-resolution, high-fidelity 3D model of a combat vehicle in VBS (to the right) alongside the real-world vehicle (to the left) (Bohemia Interactive Simulations).

The visual representation of the combat units varies from high-resolution, high-fidelity three-dimensional models to simple two-dimensional military map symbols (for example the MIL-STD-2525C standard for military map marking symbols [99]). Figure 3.18 shows an example of a high-resolution, high-fidelity three-dimensional model of a combat vehicle in VBS alongside the real-world vehicle.

3.6.1 Modelling movement and route planning

The movement of land combat units is, of course, largely dependent on the trafficability of the terrain. Mainly, there are two approaches for modelling trafficability.

In the first approach the terrain surface is subdivided into areas based on their degree of trafficability. The trafficability within each area will then be uniform, and the subdivision often form a regular grid. The areas can for example be categorized into: “good trafficability” (roads, etc.), “medium trafficability” (light terrain, fields, etc.), “poor trafficability” (rough terrain, forest, etc.), and “no trafficability” (water, steep terrain, etc.). How fast, if at all, a unit is able to move in a given area will also depend on the unit type (tracked vehicle, wheeled vehicle, dismounted soldier, etc.). This approach is commonly used in aggregate-level simulations systems, as well as some entity-level simulations systems.

The second approach for modelling trafficability is to use detailed physics-based models of movement [100]. Using this approach, vehicles will slide if they lose traction, collide with obstacles, and may flip over in steep terrain. This approach is typically applied in high-fidelity virtual and constructive simulation systems with an integrated physics engine.

Another aspect of movement modelling in constructive simulation is *route planning* (also commonly referred to as *path planning* or *pathfinding*) [101]. In addition to the trafficability of the terrain, route planning should take into account the unit's *task* or *mission*, *rules of engagement* (ROE), *tactics*, and *situational awareness* (SA) (see Chapter 3.6.2). The important aspects of military terrain analysis are often summarized by the memory aid acronym OAKOC (previously known by OCOKA), which stands for: **O**bservation and fields of fire, **A**venues of approach, **K**ey and decisive terrain, **O**bstacles, and **C**over and concealment [102]. Table 3.1 gives short explanations for these terrain factors. All of these aspects should be included in the units' SA. Human aspects like willingness to take risks will also influence route planning. A combat situation can be highly dynamic; therefore route planning can be a highly dynamic process with frequent re-planning based on updated SA and possible change of tactics.

Route planning software often generates a *network* of all possible routes based on the trafficability of the terrain. This network is often generated as a weighted graph, where each edge has an associated weight. Different variants of the *A* algorithm* [103], which finds the least costly route between two nodes in a graph, are widely used in route planning.

Route planning for large numbers of units can be computationally very expensive. However, the performance can be significantly improved by exploiting the parallelism in modern graphics processing units (GPUs) [104].

Terrain Factor	**Explanation**
Observation and fields of fire	Identify areas that provide clear observation and fields of fire for both friendly and enemy forces.
Avenues of approach	Identify possible air or ground routes that can be used for an attack, both by friendly and enemy forces.
Key and decisive terrain	Identify areas of which the seizure, retention, or control affords a marked advantage to either side.
Obstacles	Identify natural or human-made obstacles that may disrupt, turn, fix, or block friendly or enemy forces.
Cover and concealment	Identify areas that provide cover from both direct and indirect fire and concealment from enemy observation.

Table 3.1 Explanation of the terrain factors summarized by acronym OAKOC ([102]).

3.6.2 Modelling sensors and situational awareness (SA)

The term *ground truth* is used to describe the actual state of the simulated reality in a combat model. It represents "[t]he actual facts of a situation, without errors introduced by sensors or human perception and judgment" [6]. Older combat models often used ground truth as the perceived situation, and many simulation systems still use perfect detection for own forces. "In real combat [however], we [certainly] do not know everything about the opposing force, and often not even enough about our own forces. Every headquarter [therefore] continuously works on improving the *situational awareness*" [10].

Situational awareness (SA) can be described as the human operators or constructive units' perception of the environment with respect to time and/or space. This includes the ability to understand how information, events and their own actions will impact goals and objectives, both immediately and in the near future. To build a perceived SA the human operators or constructive units use their sensors and receives situation reports from other units. In a combat simulation, sensors can be understood as filters on the ground truth producing the perceived situation [10].

Conceptually, SA can be described as consisting of three levels of cognitive processes [105][106]:

- Level 1 – *the perceptual level*: This level involves the detection, recognition, and identification of elements that define a specific situation.
- Level 2 – *the comprehension level*: This level reflects an understanding of the current situation, for example identifying the enemy's current activities.
- Level 3 – *the projection level*: This level involves projecting future actions, for example what the enemy plans to do.

This conceptualisation can be employed as a framework for modelling SA for CGF [106].

The term *fog of war* is often used to describe the distorted perception, or uncertainty in SA, experienced by the units participating in a military operation. In "On War" [107] Clausewitz notes that: "War is the realm of uncertainty; three quarters of the factors on which action in war is based are wrapped in a fog of greater or lesser uncertainty". This is stated to be the source of the term fog of war.

Sensors are often categorized based on what *sense* they use to observe the battlefield [10]:

- Acoustic sensors (ears, microphones, hydrophones, etc.)
- Chemical sensors (noses, gas detectors, etc.)
- Electromagnetic sensors (radars, etc.)
- Optical sensors (eyes, binoculars, sights, telescopes, cameras, night vision goggles, electro-optical (EO) sensors, etc.)
- Thermal (infrared (IR)) sensors (thermal sights, thermal cameras, heat detectors, etc.)

It is important that the fidelity of the sensor models match the fidelity of the objects and units that are being observed. For example, if a property is important to guide a decision, this property not only needs to be modelled, but also needs to be observable by sensors and transferred into the perceived SA [10].

A sensor's effectiveness will depend on its range, which may be significantly reduced by weather and darkness, and the signature of the target, which may be camouflaged. Many sensors depend on a line of sight (LOS) to the target. LOS calculations are thus very often performed in sensor simulations. Consequently, there has been a lot of work on developing efficient LOS algorithms with high fidelity [108]. In order for a sensor to detect a target, there are three requirements that generally need to be fulfilled [10]:

1. The sensor has to be able to detect a certain property or a combination of properties.
2. The target exposes at least one of the observable properties.
3. The background does not expose the same observable property, or at least is significantly different.

There are two basic approaches that are commonly used to model sensing: *continuous sensing* and *glimpse sensing*. In *continuous sensing* a detection rate function is used to calculate how long it will take before the detection will occur when a target enters the sensor coverage area. In *glimpse sensing* the sensor regularly scans the coverage area, and for each scan there is a probability of detecting the target [10].

The fidelity of sensor modelling is closely linked to the modelled fidelity of the environment. For scenarios to be realistic it is important that realistic *scene clutter* is included. "Real-life scenarios contain forested areas, urban areas, and maritime features which can compete with or obscure the target, and successful sensor simulations need to model this significant spatial variance in background signature" [109]. Other elements that should be included in the combat model are ambient life with vehicles and humans in urban environments, light sources, shadow effects, dynamic heat sources, and vegetation which move in the wind. In addition, the use of decoys can detract attention from the real systems. Another aspect that should be modelled is the potential uncertainty associated with observing whether an enemy combat vehicle has been sufficiently destroyed, or if it is still a threat. Furthermore, for realistic representations of long-range sensor systems, the earth's curvature becomes an important factor that must be included in the terrain model.

Usually sensor models include different levels of target acquisition. For example, the terms *detection*, *classification*, *recognition*, and *identification* (DCRI) are often used to distinguish between different levels of target acquisition. These four terms can be defined as follows [110][111]:

- *Detection*: The discovery by any means of the presence of a person, object or phenomenon of potential military significance.
- *Classification*: The object is distinguished by class, for example tracked or wheeled vehicle.
- *Recognition*: The determination of the nature of a detected person, object or phenomenon, and possibly its class or type. This may include the determination of an individual within a particular class or type, for example a main battle tank (MBT) or an infantry fighting vehicle (IFV) in the class of tracked vehicles.
- *Identification*: The process of attaining an accurate characterization of a detected entity by any act or means so that high confidence real-time decisions, including weapons engagement, can be made. An example is identifying an MBT as a T-72 or a T-90.

Tracking is when a sensor system follows a detected target. It also includes estimating direction and speed for predicting future target positions [112].

Sensors and SA for aggregated units can be modelled in much the same way as for entities, using either continuous sensing or glimpse sensing. The main differences are that the units' sensors must be aggregated to represent the sum of all sensors in a unit, the SA must be aggregated to represent the units' total SA, and the targets are aggregated units.

High-fidelity sensor models depend on realistic representation of sensor ranges, and probabilities of detection, classification, recognition, and identification, for all types of targets, in all types of environmental and weather conditions, at any time of day [109]. For virtual simulations the visual fidelity of the sensor display and the sensor image is also important. Imaging sensors covering frequency bands outside the visible light range will often require generation of new texture layers for land-cover materials and models of structures and units. Figure 3.19 shows an example of a series of simulated sensor signatures of an MBT with dynamic heat sources.

Recent sensor models use GPU *shaders*[4] and *ray tracing*[5] algorithms for generating realistic sensor signatures based on material properties. An overview of state-of-the-art methods and techniques for implementing more realistic sensor models can be found in [109].

Modelling sensors that ensures fair fights when combining virtual and constructive entities are particularly challenging. In this case the sensor systems for the constructive entities must be modelled in a way that does not provide them with systematic advantages or disadvantages compared to the human operators.

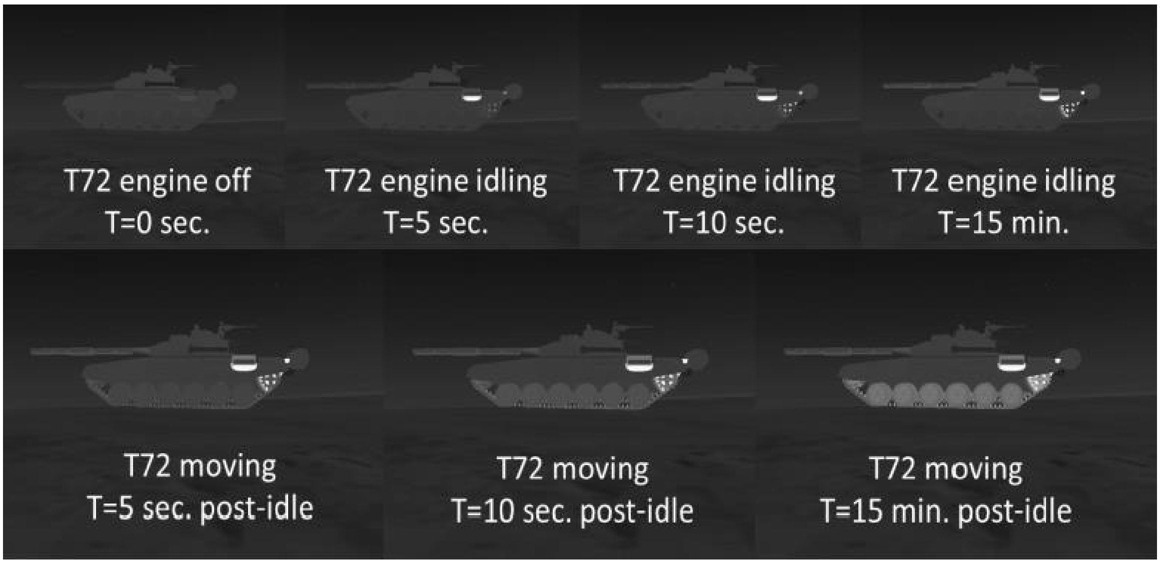


Figure 3.19 Simulated sensor signatures of an MBT with dynamic heat sources (JRM Technologies [109]).

---

[4] A shader is a program that is designed to run on a GPU.

[5] Ray tracing is a technique for generating an image by tracing rays through pixels on the screen and into the virtual scene.

3.6.3 Modelling engagements and weapons effects

Projectile weapons can be divided into *direct fire weapons* and *indirect fire weapons*. NATO defines *direct fire* as "[f]ire directed at a target which is visible to the aimer" [110], and *indirect fire* as "[f]ire delivered at a target which cannot be seen by the aimer" [110]. *Direct fire weapons* (infantry handguns and rifles, MBT guns, etc.) fire projectiles directly at a target. They have a s*ighting device* and depend on a direct line of sight (LOS) to the target. Direct fire projectiles can mainly be divided into *kinetic energy* (KE) *projectiles*, using kinetic energy to penetrate targets, and *high explosive* (HE) *projectiles*, using chemical energy to damage targets. *Indirect fire weapons* (mortars, howitzers, etc.) fire projectiles without relying on a direct LOS to the target. Indirect fire weapons are usually fired into a target area, rather than being fired at a specific target. Effective indirect fire will often require an observer or spotter responsible for directing the fire into the target area. Indirect fire weapons usually use HE projectiles. *Precision-guided munition* or *smart munition* are more advanced projectiles which include sensors that enables them to home in on targets, laser spots, or Global Positioning System (GPS) coordinates (e.g. missiles, bombs, or artillery shells equipped with guidance systems). Another type of weapon is *mines*. *Land mines* are explosive devices placed on, or concealed under, the ground.

An important factor that also should be modelled in a simulation system is active countermeasure systems against projectile weapons. Such systems include flares and active protection systems (APSs).

For entity-level simulations there are mainly two approaches for modelling engagements and duels: probability-based models and physics-based models. Probability-based models are most prevalent, especially for calculating damage. Physics-based models are commonly used for simulating ballistics, but due to increased computational power, physics-based models can now also be used for material damage simulation in real-time [66][90] (see Chapter 7.2.3).

Probability-based models use specified probabilities to calculate the result of an engagement. When a particular target type is hit by a particular type of projectile, a probability of kill (P_{kill}) and a random draw is used to determine the outcome. Often several classes of damage like *mobility kill*, *firepower kill*, *communication kill*, *sensor kill*, and *catastrophic/total kill*, with associated probabilities, are used [10]. The different parts of a target (front, side, rear, etc.) may also have different damage probabilities. In addition, for KE projectiles the damage probabilities can also take into account shooting distance or impact velocity, and impact angle. Combat models must be calibrated with realistic probabilities for different damage classes for all combinations of projectile types and target types.

In many constructive, entity-level simulations probabilities are also used to determine if a target has been hit. A particular weapon type can then be modelled with a probability of hit (P_{hit}) for each shot. Another approach is to model the projectiles as individual objects and simulate their ballistics trajectories. Whether the projectile hits can then be modelled to depend on factors like aiming error, weapon dispersion, and wind. Using this approach an actual hit point can also be calculated, and this hit point can be used in more advanced damage calculations which may use different probabilities of damage for different parts of a target type. In virtual simulation, where human operators aim the weapons, this approach is commonly used.

Indirect fire is usually not dependent on hitting a target directly for inflicting damage, but may cause damage to all targets within the blast radius. The closer the target is to the explosion, the higher probability of being damaged. The probability of damage caused by the explosion is therefore often modelled to take into account the distance between the target and the impact point.

Games and simulation systems based on game technology often use a status bar of *health points* to represent the state of an entity. Each hit will result in a decrease of health, and finally the destruction of the entity when the health is down to zero. Often entities can regain their health by being repaired by a mechanic or treated by a paramedic [10].

Damage is usually visualized by replacing the normal 3D models with pre-modelled damaged versions of the 3D models. In addition, fire and smoke is often emitted from destroyed models, and they usually have a darker texture. Figure 3.20 shows examples of 3D models of a combat vehicle with various damage states. One problem with this approach is that it will often be unrealistically easy for a human operator to observe whether a vehicle has been destroyed or not.

Figure 3.20 Examples of 3D models of a combat vehicle with various damage states (healthy, firepower kill, mobility kill, catastrophic kill) (Simthetiq).

In the recent years it has become possible for computers to perform realistic physics-based material damage simulations in real-time by employing methods for finite element analysis (FEA) and computational fluid dynamics (CFD) [66]. Consequently, detailed physics-based damage simulations for human-made structures [91] and vehicles, and wounds for humans [113], can be included in combat simulations [66][90]. This means that projectiles and war-heads can be modelled in detail, and the target damage will be individually simulated and visualized for each hit by a projectile, fragment, or shock wave [114][115]. Damage caused by vehicle collisions can also be realistically simulated using this approach.

As we mentioned in Chapter 3.5, combining simulation systems which are using different damage models may lead to fair-fight issues. A solution to this issue could be to use a *weapons effects service* (WES) which handles all damage simulations in a federation [116][117].

Aggregate-level combat simulations mainly use attrition models for modelling engagements and duels. The predominant approach is to use one of the several versions of a model based on ideas formulated by Fredric W. Lanchester in 1916 [61][10]. One example is the *Lanchester quadratic model* (which is sometimes called Lanchesters aimed fire model) which states that:

$$\frac{dB_i}{dt} = -\sum_j \alpha_{ji} R_j(t) \tag{3.1}$$

$$\frac{dR_j}{dt} = -\sum_i \alpha_{ij} B_i(t) \tag{3.2}$$

In these equations B_i and R_i are the numbers of blue and red fighting units of each type *i* and *j* respectively, and α_{ji} and α_{ij} are the *attrition coefficients* or *fighting effectiveness coefficients*, which are assumed to be constant for the duration of the battle [61]. The interpretation of this model is that the outcome for each side is determined by the numbers of opposing forces aiming their fire, multiplied by the attrition coefficients.

A final factor that needs to be taken into account is when a battle should be terminated. Battles are rarely fought until all units on one of the sides are completely destroyed. Usually, one side will reach some point where it no longer is able to reach its goal or function as cohesive force. Breakpoints are therefore often used to model that below certain thresholds units or whole forces are no longer functioning [10]. One approach for modelling breakpoints at the unit level is to look at the ratio between the remaining force of a unit and its initial force [10]. When a certain percentage of a unit has been lost, the unit will no longer be able to participate in the battle. Another approach is to look at the ratio between the blue and the red forces in a battle [10]. If a force is heavily outnumbered it may choose to retreat or surrender.

Combined arms is an approach that uses several different weapon systems and support elements in a way that their effects complement each other. By synchronising the different units and applying them simultaneously the combined effect can be much greater than if each element was used separately. An example is using air strikes, artillery fire, MBTs, and infantry together in a synchronized operation. Modelling combined arms operations requires units to be able to communicate and synchronize their effects. We will look closer at different methods for modelling communication, command, and control in Chapter 3.6.4.

3.6.4 Modelling communication, command, and control

Until recently, most combat simulation systems assumed perfect communication between units. This means that any unit could instantly and reliably communicate with any other unit. In reality however, perfect communication in land force operations is rarely achieved, especially not in mountainous or urban areas [10][119]. Factors that may cause interruptions and delays in electronic communication between units include terrain, weather, distance, interference, and available bandwidth. In addition, communication equipment may be damaged, and hostile electronic attacks such as jamming may block communication. As a result, the inability of mission critical applications to receive timely data can affect decision-making and significantly influence the outcome of the battle.

Today, we are in the information age, and modern operation concepts and combat systems rely heavily on electronic communication to be effective. *Network-centric warfare* (NCW) (also often referred to as *net-centric warfare* or *network-centric operations*) is “a military doctrine that seeks to translate an information advantage, enabled in part by information technology, into a competitive advantage through the robust networking of well-informed geographically dispersed forces” [6]. The NCW doctrine contains the following four tenets [120]:

1. A robustly networked force improves information sharing.
2. Information sharing and collaboration enhance the quality of information and shared situational awareness (SA).
3. Shared SA enables self-synchronization, and enhances sustainability and speed of command.
4. These, in turn, dramatically increase mission effectiveness.

Of course, to be effective, NCW requires a robust communication infrastructure [119].

It is obvious that communication simulation should be a key component in combat simulation, equal in importance to moving, observing/sensing, and shooting/engaging. “In modern warfare, sensor components are often separated from shooting components. Without proper communication, effectors do not get targets assigned, and [C2] nodes do not get the information they need to plan, command, and control the execution” [10]. All of these factors should be taken into account in a combat model. Consequently, communication models must include functionality for distributing tactical orders, supporting the coordination of fire and combined arms effects, and transmitting spot reports of enemy units.

Realistic communication simulations must include detailed models of communication equipment (radios, satellites, repeaters, relay stations, etc.), connectivity, and transmitting data (e.g. radio wave propagation models). Probability-based or detailed physics-based models can be used to simulate the data transmission based on the characteristics of the communication equipment and the influence of terrain, line of sight (LOS), distance, weather, interference, and opposing activities such as jamming. Furthermore, the ability to detect and destroy communication equipment must be modelled [10].

“[Land force operations] may include hundreds or even thousands of communicating devices. An accurate analysis of network performance should model the entire deployed network. Analyses based on smaller subsets of the network may yield overly optimistic performance results, as protocols may break down as the network becomes more complex” [119].

Existing communication effects applications provide detailed representations of communication networks and effects, but the simulated units in these applications lack the detailed behaviour, dynamics, and mobility present in modern combat simulation systems [119]. These detailed communication effects models could either be implemented in combat simulation system, or integrated in a combat simulation as a *communication effects service* (CES), which handles the simulation of all communication effects in a federation [116][117][119].

Another challenge when modelling communication is representing voice communication between virtual and constructive entities, which means supporting voice communication between human operators and CGF. This can be done by using technologies for synthetic speech generation, and technologies for automatic speech recognition (ASR) and natural language processing (NPL) [121][122], but especially the latter is still difficult to achieve.

Command and control (C2) is “the exercise of authority and direction by a properly designated commander over assigned and attached forces in the accomplishment of the mission. C2 functions are performed through an arrangement of personnel, equipment, communications, facilities, and procedures employed by a commander in planning, directing, coordinating, and controlling forces and operations in the accomplishment of the mission” [6].

In the early days of combat simulation, all decisions and C2 activities had to be performed by human operators [10]. However, gradually the simulated units have been modelled with the ability to perform more and more complex orders and tasks. This type of combat simulations, which use semi-automated behaviour models, is commonly referred to as *semi-automated forces* (SAF). In modern combat simulation systems, more and more of the C2 activities, and the command posts and headquarters that execute them, are modelled [10]. The better C2 is modelled, the less human commanders are needed to provide realistic combat simulations [10]. We will look closer at different methods for modelling human behaviour and decision processes in Chapter 3.6.5.

3.6.5 Modelling combat service support

In addition to the units that are taking part in actual fighting, land combat models should also include combat service support units. Such units include:

- Combat engineering units
- Logistics and supply units
- Maintenance units
- Medical units

The main tasks performed by combat engineering units are facilitating movement and support for friendly units, and preventing movement for opposing units. Combat engineering units include construction units, bridge layers, and mine clearing units. High resolution models of engineering vehicles can be modelled to interact with dynamic terrain [77]. Figure 3.21 shows examples of detailed models of engineering vehicles in VBS.

Logistics and maintenance are, of course, very important for continuous land force operations. “Cutting the logistics lines can be more effective to deny success to an opponent than engaging his combat troops” [10]. Logistics and maintenance can be modelled using high resolution models based on individual entities and individual processes, or using aggregated models based on supply and recovery rates, damage class distributions, and mean time between failures (MTBF) [10].

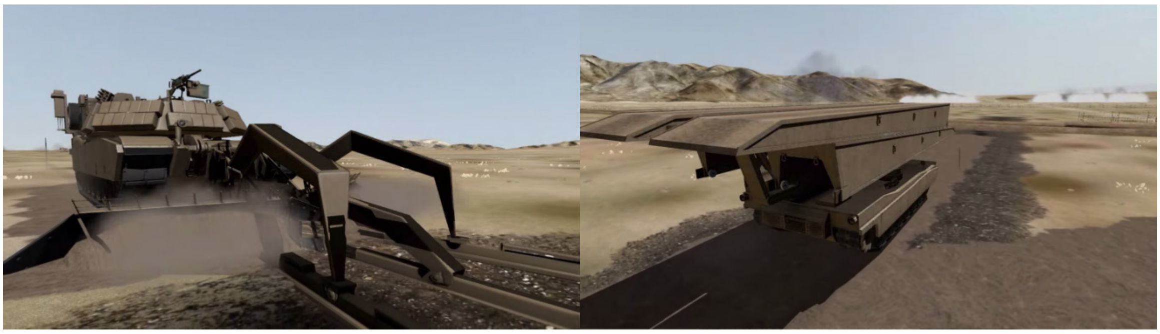


Figure 3.21 Examples of detailed models of engineering vehicles in VBS (Training Brain Operations Center Systems Integration Modeling and Simulation (TBOC SIMS)).

3.7 Modelling human behaviour

“[War] is fundamentally a human issue” [5] that is “waged between complex human organizations” [5]. Including the human dimension in combat simulations is therefore paramount. There are two possible approaches for including human behaviour in combat simulation. One approach is to include real humans in the simulation (see Chapter 2.5). The second approach is to use human behaviour models. Simulations using SAF combine both approaches.

Human behaviour is the “collective set of actions exhibited by human beings, either individually or in groups of various sizes and compositions” [123]. Factors that may determine and affect human behaviour include physical properties (e.g. strength, endurance), cognitive properties (e.g. memory, reasoning), and social properties (e.g. cultural norms, role in social group) [124].

Modelling realistic human behaviour and cognition, including decision-making and creativity, is the hardest and most complex challenge in combat simulation [59]. Human behaviour modelling is challenging because "[h]uman behaviour is not generally yet thought to obey observable laws" [5]. "In general, the behaviour of large number of human beings does not currently appear to behave in accordance with deterministic rules" [5]. Consequently, the current status for human behaviour simulation is that it can be used "to understand, [but] not necessarily predict, the aggregate behavior of an inherently complex system for which we have no better model" [123]. When using human behaviour models "it is often possible to perform sensitivity analysis and identify broad trends as opposed to exact predictions" [123]. For example, a simulation using CGF may show that increasing the number of main battle tanks (MBTs) has a positive effect on the outcome of a scenario, but it cannot be used to pinpoint the exact number of MBTs required to win the battle with a certain probability [123].

Human behaviour can be divided into the *physical*, *tactical*, and *strategic* level, based on the complexity of the goal of the behaviour and the duration of the performed activity [123]. At the *physical level* human behaviour is driven by physiology and automated processes like stimulus response and motor skills. "Decisions are done at an instinctive or reactive level, and emotions have little impact on the process; instead, performance is governed by the level of workload, fatigue, situational awareness, and other similar factors" [123]. Examples of this level of behaviour are walking, driving a vehicle, and firing a weapon. Human behaviour at the *tactical level* is driven by short-term goals and includes tactical decision-making and emotions [123]. At the *strategic level* human behaviour involves long-term planning and complex, high-level decision-making based on experience, intuition, and emotions [123].

The levels of human behaviour can be related to decision-making and behaviour at the different levels of military simulation outlined in Figure 2.4. Human behaviour at the physical level can be modelled using physics-based models and performance data. "[T]actical and strategic behaviors are harder to model due to the adaptive and unpredictable nature of human behavior. When incorporating larger populations, the complexity drastically increases to the point where such models are difficult, if not impossible, to validate" [123].

Human behaviour and decision-making at the tactical level is considered as a decisive factor for success in combat. "Although military decision-making at the operational and strategic levels is sometimes pressured and often of weighty consequence, it does not have the dynamism and suddenness of consequence of tactical decision-making. Since fighting is the currency of war, tactical decision-making is a key, and perhaps *the* key, activity in warfare. It is the mental activity which most directly affects the outcome of combat" [5]. "Tactical decision-making is unique. Only in war is decision-making routinely of lethal consequence to many, whatever the outcome. Only at the tactical level must decision-making be carried out in real time" [5].

There are mainly two schools of thought for modelling the higher levels of human behaviour. The first considers human beings as *rational* entities, and focuses on modelling rational decision-making to achieve a specific goal based on deterministic or stochastic approaches, ignoring the effect of emotions. The second considers human beings as *quasi-rational* entities that still pursue a specific goal, but frequently make suboptimal decisions and even exhibit actions that can act contrary to their goal [123]. "Rational decision making is by far easier to model and simulate than quasi-rational behavior" [123]. The processes that govern suboptimal decision-making are complex, and are not yet fully understood [123].

Artificial intelligence (AI) is the field of study for creating intelligence exhibited by machines or software. An autonomous intelligent entity is referred to as an *intelligent agent* (or an autonomous intelligent agent). An intelligent agent observes the environment through sensors and acts upon the environment using actuators in order to pursue its goals. It may also learn or use knowledge to achieve its goals. An agent can also be an aggregated unit. A *multi-agent system* (MAS) is a system composed of multiple interacting intelligent agents within an environment. Figure 3.22 shows a proposed architecture for generic intelligent agents [125][126]. Agent architectures that more specially attempt to model human cognition are referred to as *cognitive architectures*. It should be noted that, whereas AI systems in general are designed to complete tasks faster and with fewer errors than human beings, human behaviour models are designed to complete tasks in the same way as humans are expected to complete the tasks [127].

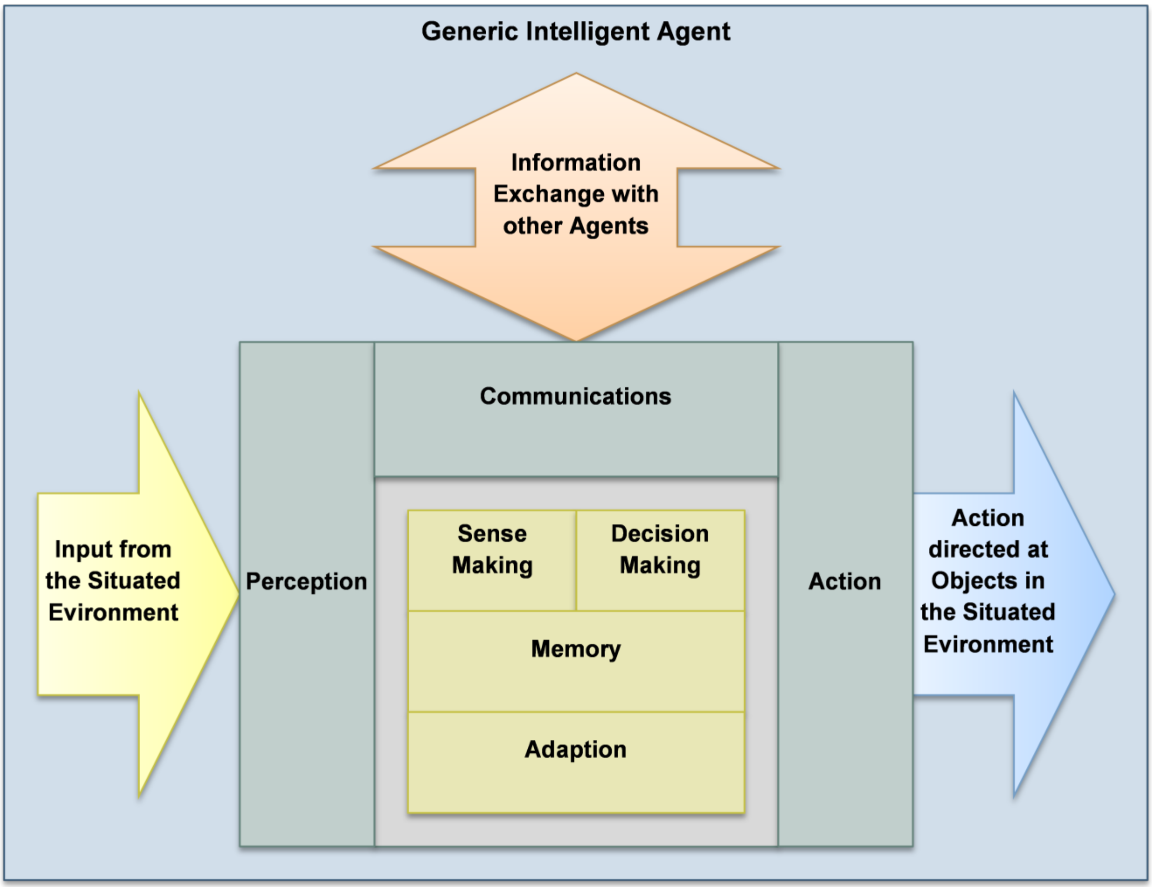


Figure 3.22 Architecture for generic intelligent agents ([125][126]).

Intelligent agents can be computationally very expensive. Consequently, the number of agents that can be simulated by a MAS running on a single computer is limited.

Techniques for developing AI and intelligent agents include *fuzzy logic, finite-state machines* (FSMs), *behaviour trees* (BTs), *utility-based systems*, *rule-based systems* (RBSs), and *pattern recognition* [123][128]. Each of these techniques is briefly described in the chapters 3.7.1 to 3.7.6. Which technique to use depends on the application, and comprehensive behaviour model systems often utilize several of these techniques.

Examples of AI engines (or AI middelware) used in simulation systems are Kynapse from Autodesk, VBS CONTROL from BISim, MASA LIFE from MASA Group, AI.Implant [129] from Presagis, DI-Guy AI from VT MÄK, and B-HAVE from VT MÄK.

Human behaviour is another example of a simulation component that can be implemented as a service, in the form of a CGF service or a MAS service [117][130].

The AI engines used in many of the current CGF systems tend to be rigid and predictable, and lack the ability to adapt to new and unexpected situations [127][131]. Moreover, they rarely incorporate individual differences and human imperfections [122]. For the user, it is also a problem that extending and customizing behaviour is a complex and time consuming task that often requires special expertise [127][132]. A final problem, that is currently being addressed, is the lack of interoperability standards for human behaviour representations [124].

One way of validating human behaviour models is applying the *Turing test* (also known as the *imitation game*) [133], named after the British mathematician and computer scientist Alan M. Turing. A computer system is said to pass the Turing test, and thus exhibit intelligent behaviour, if an observer cannot reliably distinguish between the computer-generated and human-generated behaviour. More specially, a human behaviour model is said to pass the Turing test if an observer cannot reliably distinguish between the model-generated and human-generated behaviour [14]. The Turing test can be seen as a form of face validation (see Chapter 2.9).

Different levels of resolution in combat simulation require different levels of resolution for human behaviour models. It is much more challenging to develop realistic human behaviour models for entity-level simulations than for aggregate-level simulations. With regards to the Turing test, it is much easier to reveal unrealistic behaviour in entity-level combat simulations with high resolution. To increase the realism of current entity-level constructive simulations (automated and semi-automated computer-generated forces), there is first and foremost a need for more realistic tactical AI.

In modern military operation types, including irregular warfare (IW), counter-insurgency (COIN), counter-terrorism (CT), peacekeeping operations (PKO), and stability operations (SO), the human dimension is becoming an even more dominant factor than in traditional force-on-force operations. Consequently, modern and future combat simulation systems need to include even more sophisticated models of human behaviour (including social and cultural factors) to be applicable to the full range of military operations [134][135].

3.7.1 Fuzzy logic

Fuzzy logic is a form of multivalued logic that uses fuzzy truth variables which may take on values between 0 and 1 representing varying degrees of truth. Fuzzy logic is suitable for simulating human perception of the world, which is approximate and not exact, and is often used to model human control tasks [123].

3.7.2 Finite-state machines (FSMs)

Finite-state machines (FSMs) (or finite-state automata) are abstract machines that can be in one of a finite number of states. When FSMs are used for human behaviour modelling, each state is associated with actions that represent a distinct behaviour phase that an agent can exhibit [123]. FSMs can go from one state to another via transitions triggered by events or conditions. Directed graphs can be used to visualize the structure of an FSM. Figure 3.23 shows an example of an FSM modelling the behaviour of a simple patrol.

FSMs is the dominant modelling technique for CGF [136][137]. "FSMs are particularly good for implementing well-defined doctrinal behavior of limited complexity" [137]. A significant limitation of FSMs is that because the set of states, and thus the set of behaviours, is finite, an entity model based on an FSM can never learn [123]. Also, when new tactics, techniques, and procedures (TTP) are introduced, FSMs need to be redesigned [137]. Another limitation of FSMs is that they tend to become very complex as the number of nonmutually exclusive behaviours increases. "The number of states required to implement *N* nonmutually exclusive behaviors is 2^N, something that can quickly become unmanageable. Even when behaviors are mutually exclusive, the number of states and associated transitions required to model complex behaviors quickly escalates" [123].

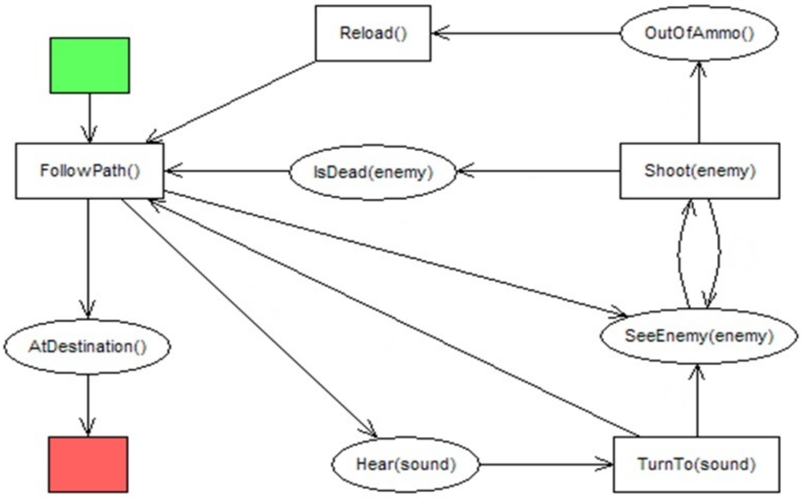


Figure 3.23 Example of a finite-state machine (FSM) modelling the behaviour of a simple patrol (Stottler Henke).

To limit the complexity of basic FSMs they can be extended to include hierarchy and concurrency, as well as domain- and context-driven extensions. In hierarchical FSMs there are multiple levels of states. Concurrency allows more than one state to be active at the same time. Context-based reasoning (CxBR) [138][139], is an example of a context-driven approach which can be modelled through hierarchical FSMs [140]. CxBR is being used to model battle command at FFI [141].

3.7.3 Behaviour trees (BTs)

Behaviour trees (BTs) are represented as directed trees with a hierarchy of control flow nodes and task nodes that control the behaviour of an entity. The control flow nodes contain some decision logic and have one parent node and at least one child node. The task nodes are leaf nodes (nodes without child nodes) and contain conditional tasks which test some property in the simulation, or action tasks which alter the state of the simulation in some way.

BTs have some similarities to hierarchical FSMs with the key difference that their main building blocks are tasks rather than states. They are frequently used to model non-player characters (NPCs) in computer games. There are several different approaches for implementing BTs, and the first unified framework for BTs was published in 2014 [142].

3.7.4 Utility-based systems

Utility-based systems are systems "in which decisions are made on the basis of heuristic functions that represent the relative value (or appropriateness) of each option under consideration in terms of a floating-point value" [131]. Utility-based approaches typically include the following three general steps [131]:

1. Build a list of options.
2. Evaluate each option and calculate one or more floating point values that describe how attractive the option is given the current situation. A key point is that this evaluation occurs at run-time.
3. Select an option (or set of options) for execution on the basis of the values calculated in step 2.

Utility-based human behaviour models are constantly evaluating the situation and selecting the most appropriate option or options at each moment in time [131].

3.7.5 Rule-based systems (RBSs)

Rule-based systems (RBSs) consist of a set of rules (rule-base), a temporary working memory, and an inference engine. When RBSs are used for human behaviour modelling, the rules represent the agent's knowledge, and consist of pairs of "if-then" statements that each encodes a distinct condition and a corresponding rational action. The working memory contains the agent's perception of the environment. The set of rules are tested against the working memory. If a condition is true it can, depending on how the system is implemented, either (1) result in some action, or (2) a modification of the working memory. If multiple conditions are true the inference engine selects which rule is executed. In case (2) the rules continue to be tested against the working memory until a terminal rule is executed or no rule can be executed. The working memory then represents the agent's best possible assessment of the current situation, and actions are executed based on the terminal state of the working memory. The whole process then starts over again with a new working memory [123]. Figure 3.24 illustrates the structure of an RBS used for human behaviour modelling, as described in case (2).

RBSs are easy to extend since it is possible to add new rules without having to worry about explicit dependencies among them [123]. In some RBSs, known as production RBSs, it is also possible for the system itself to dynamically add more rules, and thus simulate learning and adaption. Two frequently used cognitive architectures using production RBSs are Soar [143] and ACT-R [144].

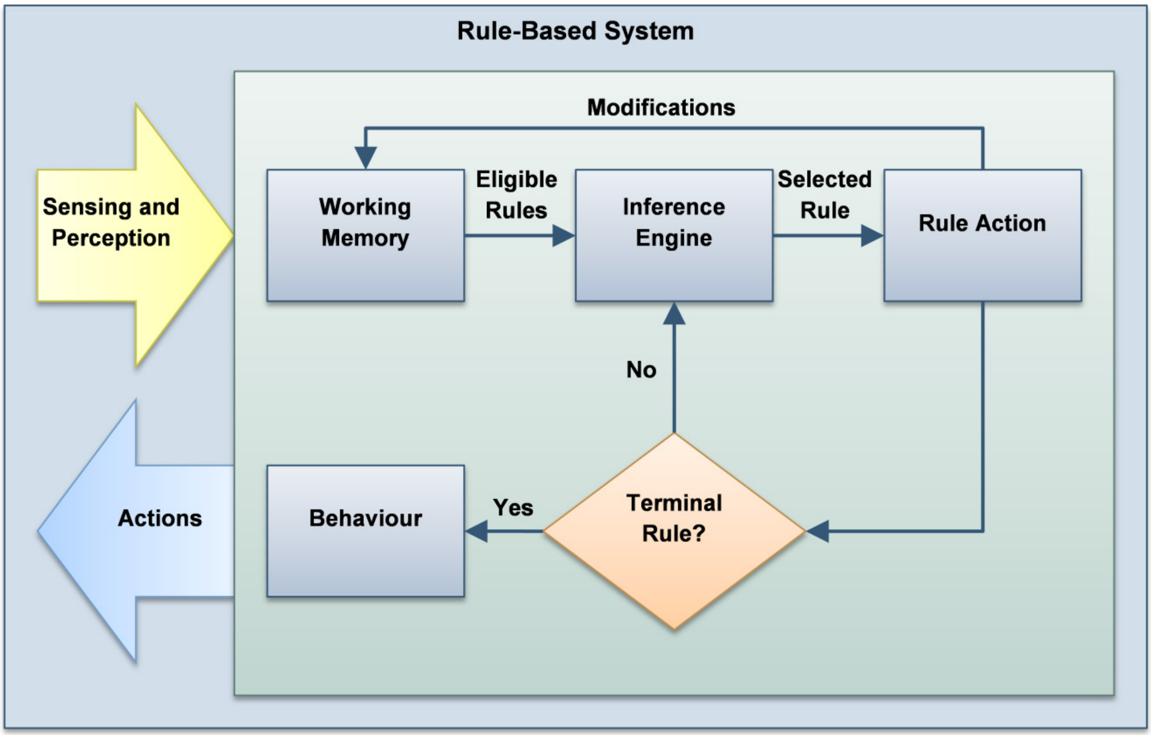


Figure 3.24 Structure of a rule-based system (RBS) used for human behaviour modelling ([123]).

3.7.6 Pattern recognition

Pattern recognition is the "conversion of raw data into meaningful and contextualized data structures that can be used for further processing" [123]. In human behaviour modelling pattern recognition can be used to recognize emerging patterns in the perceived situation. This information can then be utilized to decide on the appropriate course of action [123]. Figure 3.25 illustrates this process.

Two techniques that are commonly used for modelling pattern recognition are artificial neural networks (ANNs) and hidden Markov models (HMMs) [123].

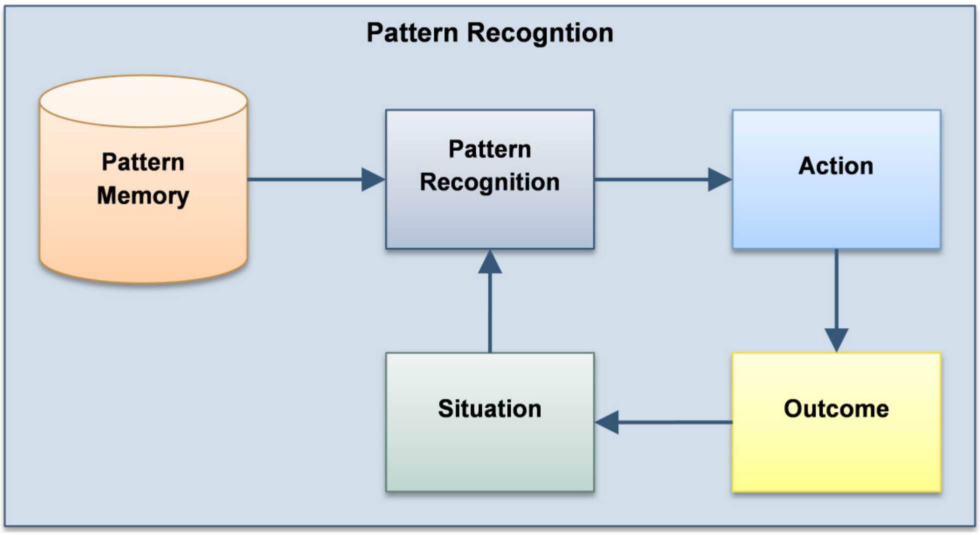


Figure 3.25 Pattern recognition used for human behaviour modelling ([123]).

3.8 Human-in-the-loop simulation

In computer-based combat simulation, *human-in-the-loop* (HITL) simulation means virtual simulation, or constructive simulation with SAF (see Chapter 2.5). Virtual simulations include real system operators, while constructive simulations with SAF include real humans (etc. military leaders) controlling the forces. Including real humans in the simulation, and especially using virtual simulation, is the approach that gives the most realistic representation of the human aspects. Moreover, with the current status for human behaviour modelling (see Chapter 3.7), it is often necessary to use HITL simulation to achieve the desired level of fidelity for representing human behaviour. However, it is expensive and often impractical, or not even technically possible, to use human operators for every entity in a simulation with hundreds of entities. In general, it can also naturally be problematic to execute HITL simulations faster than real-time.

There are, of course, situations where HITL simulations are absolutely necessary. Human operators allows for the identification of problems and requirements that may not be easily identified otherwise. For example when experimenting with human performance systems and user interfaces, human operators are essential. Also, when experimenting with new technology, or novel force structures and operational concepts, including human creativity in the simulation is essential. Such experimentation will often involve a learning process before the new and novel systems are utilized in an optimal manner.

HITL simulations automatically include the mental properties of real humans, but the simulated environment will not trigger the fear and stress humans experience in real combat. Using SMEs, like real system operators or real military commanders, in a simulation will also function as an additional built-in face validation of the simulation. It is usually also easier to understand the results of an HITL simulation, since it is normally followed by an after-action review (AAR) including discussions with the participants. Including military personnel, and especially military leaders, in the simulation, will also often lead to more confidence in the results among stakeholders [3][4].

Including human participants in a simulation will also include the element of competition (since human participants do not like to lose). In a combat simulation with blue and red forces, both teams will do everything they can to win the battle, and the losing team will try to modify and improve their tactics for the next run. This interplay between the opponents, where competition is the engine that motivates innovation, will often continue until a state of equilibrium occurs [59]. It is also possible that some participants will try to "game the system" by exploiting weaknesses in the simulation system to obtain unrealistic advantages. To prevent this, it is important that the simulations are closely monitored by administrators and umpires with in-depth knowledge of the simulation system. Also, to avoid systematic advantages to one of the sides, it is important that the participants' skills are fairly balanced between the blue and red forces (i.e. fair fight).

3.9 Combat friction

In combat the term *friction* can have a variety of meanings, often based on the writings of Clausewitz in "On War" [107]. Clausewitz used friction, or *grand friction,* in a broad sense to describe factors that affect armies in larger campaign, theatre-level action, or national war activities. It can also be described in a narrower sense, as the reduction of combat power due to causes that cannot clearly be laid neither to enemy action nor to own-force deficiencies. The cause of this reduction can be called *combat friction* [145].

The sources of combat friction may vary greatly, and are best described through examples [145]:

- Heavy snowfall occludes visibility of a unit. The reduction of visibility is a friction. Interactions with weather and other physical elements may cause equipment wear or malfunctions, and cause further friction.
- Fatigue as a result of prolonged combat entails several kinds of frictions. Firstly, the reduced effectiveness as a result of fatigue is a friction. Furthermore, personnel within the unit may take faulty cognitive actions, and by their actions cause more friction.
- Orders that are unclear or ambiguous may be incorrectly interpreted. Actions based on these misunderstandings may cause friction.
- The frequently changing of orders in a given mission will cause frictions of many kinds.

"To generalize the many sources of combat friction, we can say that friction arises from inefficient and disorganized activity, redundant activity, damping effects of the combat environment and other constraints, wear and tear and fatigue in individuals and materiel, and (perhaps most important) faults in the functions of command, control and communication" [145].

How combat friction is represented in different levels of combat simulation vary greatly. However, simulated operations generally tend to run smoother than operations in the real world. Often the lack of data is a major factor to why friction is left out. Clausewitz explains that the very nature of friction is that it is random and unpredictable, that there are elements that cannot be anticipated, and that their prediction cannot be learnt theoretically. The most predictable part is that friction will in some way occur.

3.10 Distributed simulation and federations

A distributed or federated simulation is a simulation that consists of multiple individual simulations or simulation components (see Chapter 2.6). “The main reason for building a federation is the coupling of functionality of contributing systems to provide a new capability” [10]. The two most widely used interoperability standards for distributed simulation are Distributed Interactive Simulation (DIS) and High Level Architecture (HLA) (see Chapter 2.7).

There may be several reasons for developing a federation over developing or modifying an individual simulation system [10]:

- Federations allow the reuse of existing solutions.
- Federations support modularity, and therefore help to reduce complexity.
- Federations allow cost to be reduced.
- Federations can be used to compose cross-domain solutions.

In addition, federations use multiple computers, and thus increase the total computational power available for the simulation.

The main challenges with distributed simulation are underlying differences between simulation models and uncorrelated terrain databases, which can lead to strange effects and fair-fight problems (see Chapter 3.5) [18][97][98]. Through this chapter we have looked at different types of simulations (virtual and constructive), different levels of resolution for representing terrain and combat units (entity-level and aggregate-level), different ways of modelling the four core activities of combat (moving, observing/sensing, shooting/engaging, and communicating), and different ways of representing the human aspect. This variety of different ways of representing the real world, illustrates the challenges involved in developing federations. In addition, the simulation systems taking part in a federation may represent time differently, entailing the need for time management [10].

To deal with the problems that may arise when building federations, the following techniques are often useful [97]:

- Limit testing to the scenario or class of scenarios that you intend to run. The more specific the scenarios, the better.
- Divide the federation so that objects that operate in similar environments are all simulated in the same simulation. For example by simulating aircraft in one simulation and ground vehicles in another.
- Use persistent federations, where a history of testing and interoperability modifications gives confidence that the models are compatible, at least for previous purposes.
- Limit the scenario to regimes where the simulation models give consistent results. For example if one simulation doesn't model night operations well, don’t run a night scenario.
- Dumb down simulations that overachieve. Raising the least common denominator is usually much harder.

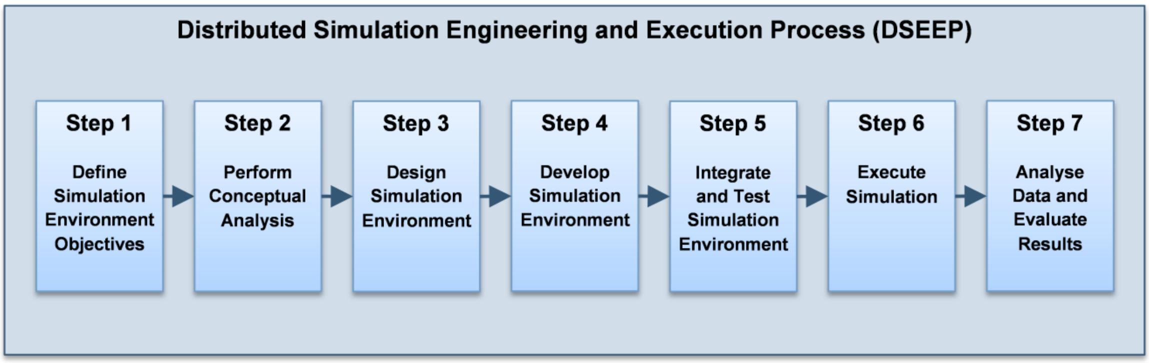


Figure 3.26 The seven step process model for the development and execution of a distributed simulation environment as defined by the Distributed Simulation Engineering and Execution Process (DSEEP) ([146]).

All in all, the challenges associated with building a federation must be weighed against the potential advantages. Many simulation tasks may be easier solved without using a federation.

The Distributed Simulation Engineering and Execution Process (DSEEP) is an IEEE standard (IEEE Standard 1730-2011) [146] which describes a generalized process for building and executing distributed simulation environments. Figure 3.26 illustrates the seven top-level steps in the DSEEP process model [146]. DSEEP is based on, and supersedes the Federation Development and Execution Process (FEDEP) (IEEE Standard 1516.3-2003).

Note that HLA, DIS, and DSEEP use different terminology [10]: HLA uses the terms *federation* and *federates*, DIS uses the terms *simulation exercise* and *simulation applications*, and DSEEP uses the terms *simulation environment* and *member applications*.

Current federated simulation environments have been found to suffer from two major problems [98]:

1. High efforts for preparation, initialization, execution, and analysis of a federation in terms of time, costs, and resources (mainly personnel).
2. High efforts for verification and validation (V&V) of a federation. For example achieving fair fight and credible simulation results (if achievable at all) require a lot of time and resources.

In [98], Siegfried et al. presents a set of high-level requirements for the next generation distributed simulation environments. Furthermore, they present the following list of recommendations on system design for the next generation distributed simulation environments:

1. Design and document for interoperability.
2. Design and document for modularity and composability.
3. Favour open standards.
4. Design for securability.

The next generation distributed simulation environments are expected to rely heavily on open standards and service-based architectures [98]. The combination of service-based approaches for M&S with ideas taken from cloud computing is known as M&S as a service (MSaaS) [98][117]. We will come back to MSaaS in Chapter 7.1.2.

Increased standardization and MSaaS may solve many of the challenges with distributed simulation environments. However, it often takes several years before standards are established for new simulation technologies (e.g. dynamic terrain or destructible buildings).

3.11 Obtaining data and calibrating models

Obtaining realistic data for the simulation models are essential for the validity of the simulations. "The quality of the system is not only driven by the quality of the model itself, but equally by the quality of the data. The best model is useless if no data can be obtained to drive it. Data therefore play a central role" [10].

Different simulation systems require different types of data. Entity-level simulations require performance data for the individual types of entities, while aggregate-level simulations require aggregated performance data. Moreover, simulation tools generally use their own sets of parameters and data formats.

Sources of data for combat simulations include technical system specifications, real-world operations, and real-world experiments and field exercises. Military performance data are often classified, but there are also several openly available data repositories like IHS Jane's [147], GlobalSecurity.org [147], and the Worldwide Equipment Guide [149][150][151].

"If data is not available or cannot be obtained, it is good practice to use the knowledge of subject matter experts to generate the needed data. However, if empirical data become available at a later phase of the study, it is good practice to replace expert-generated data" [10].

Data from simulations with higher resolution can also be used to calibrate simulation models with lower resolution and increased aggregation. For example, data from detailed, physics-based simulations of material damage from projectile penetration (which may not run in real-time) can be used to calibrate damage calculations for entity-level simulations. Figure 3.27 illustrates this.

Obtaining realistic data for combat simulations can be an extensive and time consuming task. Furthermore, as models and simulations become more complex, the amounts of data which are required will continue to increase. Much work can potentially be saved by facilitating sharing and reuse of collected data, for example by collecting data in databases.

An approach for generation, storage, verification and validation of performance data for combat simulations is presented in [152]. The described system also includes tools for exporting data to the formats used by several different simulation systems.

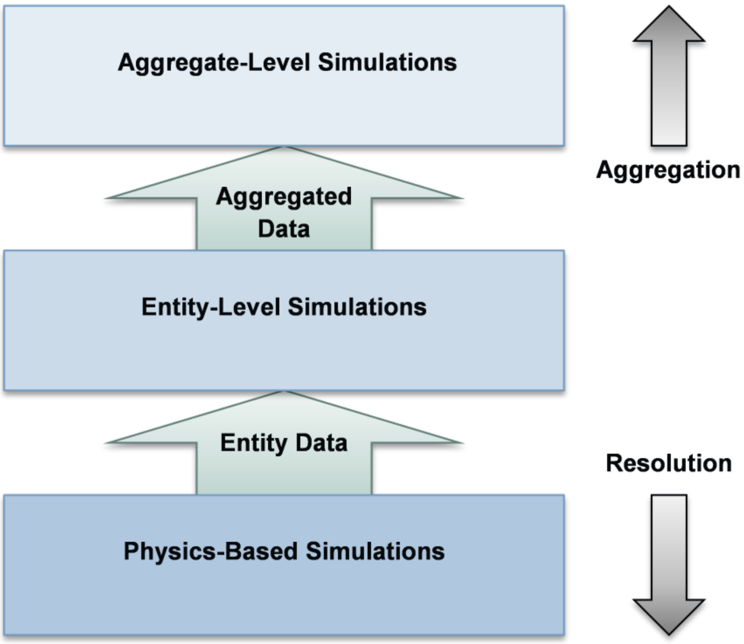


Figure 3.27 Simulations with higher resolution can be used to calibrate simulation models with lower resolution and increased aggregation.

Most simulation systems are usually delivered with models of the most common vehicles, weapon systems, and sensors. However, the data used in these models are often very generic, and do not capture the differences of the individual combat systems (e.g. all MBT types use the same generic data). Although this may be good enough when using the simulation system for training battle command, it is usually not good enough for conducting detailed experimentation and analysis, where the goal often is to compare the performance of different combat systems.

3.12 Scenarios and tactical vignettes

A *scenario* can be defined as the "description of the area, the environment, means, objectives, and events related to a conflict or crisis during a specified time frame suited for satisfactory study objectives and the problem analysis directives" [153]. In more detail, a scenario usually includes a description of the following [10]:

- The geospatial definition of the scenario.
- A description of the environment, including terrain and weather, relevant to the scenario.
- The definition of the mission and the means required to achieve the mission of the scenario.
- The objectives that define the mission of the scenario on all relevant levels.
- The events that will take place during the time the scenario is intended to span.

A *tactical vignette* can be defined as a specific playable course of battle extracted from a larger scenario.

For a simulated scenario or tactical vignette to be *operationally realistic* it needs to include realistic numbers of entities, realistic operational areas, and realistic complexity. Large areas, high numbers of entities, and the complexity of their behaviours and appearance are important for achieving operational realism in virtual environments [154].

In simulation for analysis it is important that all problems and issues that need to be analysed are addressed in at least one of the scenarios being used. If an issue has been identified but does not become a part of a scenario, it cannot be evaluated [10].

Most simulation systems include integrated tools for scenario generation, and they typically use their own formats for storing scenarios. The *Military Scenario Definition Language* (MSDL) (see Chapter 2.7.4) was created to provide a mechanism for loading military scenarios independent of the application generating or using the scenario.

In our simulations we mostly use tactical vignettes derived from national defence scenarios. The tactical vignettes range from brigade level (e.g. used in constructive simulations for analysing army structures and operational concepts) to company level (e.g. used in virtual simulations to evaluate new technologies).

3.13 Output data and results

In a combat simulation it is important to capture output data and results throughout the whole simulation. In simulation for analysis, understanding the outcome of the battle can be just as important as the outcome itself.

It is important to not only look at the numbers in the kill matrices, since “this direct computation of system values only taking direct effects of destroying systems into account is insufficient for analyses, as the effect of enabling – or disabling – supporting systems is not taken into account. A tank company can only shoot at enemies if they have enough ammunition and fuel, so even if the supporting logistic systems do not kill a single enemy their contribution is enormous” [10]. Another example is air defence systems that may not shoot down many enemy air-planes, but the enemy air-planes have to stay out of the air defence systems' range.

To get useful output data for analysis, careful design of both experiments and scenarios is important. We have already mentioned that all issues that need to be analysed must be included in the scenarios. Typically, the following output data may be obtained from a combat simulation:

- Logs of data (i.e. events, detections, and kills) from the simulation system
- Kill matrices
- Video recordings of the scenario
- Answers from questionnaires for the operators in human-in-the-loop simulations
- Notes from after-action review (AAR) sessions

In our simulation experiments we generally strive to capture as much output data as possible.

Generally, more computing power allows increase in the fidelity of simulations, which correspondingly increases the volumes of data simulations are capable of generating. However, "unless [all this] data are analyzed and converted into information, simulations will provide no useful knowledge" [155].

We have found that human-in-the-loop (HITL) simulations (virtual simulation or constructive simulation with SAF) with military participants make it easier to understand the results [3]. In addition HITL simulations have a built-in face validation, which in our experience leads to more confidence in the results [3][4].

Finally, an assessment of the simulation and the analysis must be done. This means clarifying the limitations of the simulation, and evaluating how reliable the results are. V&V is the key activity in evaluating the reliability of the output data and results.

3.14 Verification and validation (V&V)

V&V (see Chapter 2.9) is essential to credible and reliable use of M&S. It is important to remember that a model or simulation is validated for an intended use. Models are always simplifications of the real world, and can therefore never be absolutely valid.

The following four principles of V&V will help ensuring good results [10]:

1. V&V must be an integrated activity with model selection, development, and integration activities.
2. The intended purpose of a model or simulation must be specified precisely.
3. Sufficient knowledge about the system to be simulated is necessary.
4. Sufficient V&V tools and V&V experts must be available.

Generally, available V&V methods can be divided into the following four categories [156][14][10]:

- *Informal V&V methods* rely heavily on human intuition and subjective evaluation, and are conducted by SMEs based on their experience with comparable solutions that can be used as a reference. Examples of such methods are: inspection, face validation, and the Turing test.
- *Static V&V methods* base their assessment on the characteristics of the model design and implementation without execution thereof. Examples of such methods are: data analysis and cause-effect graphing.
- *Dynamic V&V methods* conduct their assessment by executing the simulation system and evaluating the results, including comparison with other models or observations in experiments conducted in the real world. Examples of such methods are: execution tracing, sensitivity analysis, comparison testing, and statistical methods.
- *Formal V&V methods* are based on rigorous mathematical proofs of correctness. Examples of such methods are: inductive assertions and predicate calculus.

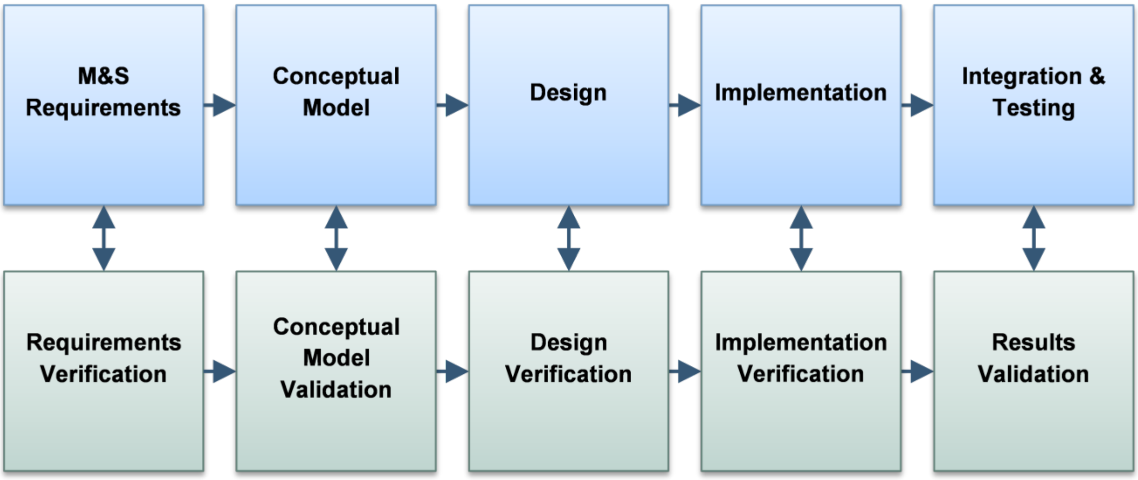


Figure 3.28 High-level view on V&V activities ([10]).

Figure 3.28 shows a high-level view of what needs to be verified and validated in a simulation system [10]. When building a federation it should be pointed out that even if the federates have been validated separately, the federation can still produce invalid results. It is therefore essential that the federation is validated as a whole.

How much resources that should be spent on V&V depends on the purpose of the simulation. Obviously, the V&V requirements for a high-fidelity flight simulator are much higher than for an aggregate-level battalion commander training system. Combat simulations for the purpose of conducting analysis are often used to support defence planning and acquisition. Increased V&V in such simulations will help reducing the uncertainty of the results, increase the credibility of the results, and reduce the risk of making bad decisions. Figure 3.29 illustrates the relationship between a model's cost, credibility, and utility [156]. Generally, increasing a model's credibility leads to increased development cost. At the same time, it increases the model's utility, but usually at a decreasing rate. At some point increasing the model credibility becomes very costly, while providing very little increase in model utility.

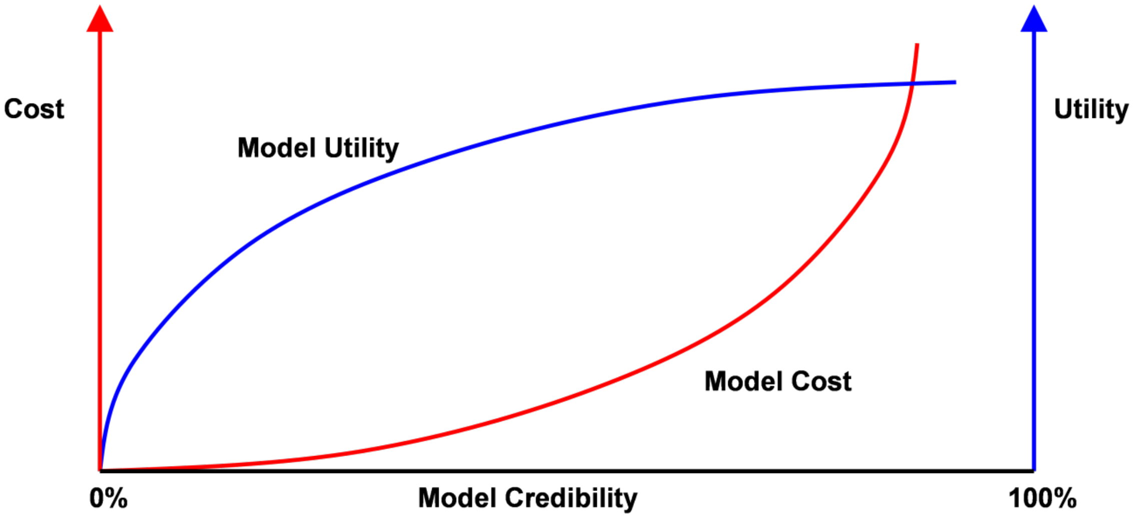


Figure 3.29 The relationship between a model's cost, credibility, and utility ([156]).

The main challenge with validating combat simulations is that it is most often not possible to compare the results to real-world situations (which of course is not a bad thing). Even if the technical properties of the individual combat units have been validated, combat is so complex, especially in its human aspects, that we have no way of predicting the behaviour of the overall system.

3.15 The complexity of war

Modern combat is highly complex and highly dynamic. In [5] Jim Storr gives the following description of the complexity of war: "We appear to be observing complex and perhaps chaotic behaviour. The fundamentals of complexity theory consider the behaviour of large numbers of elements and the relationship between them. The way those elements are organized results in lots of feedback loops. Any system with a large number of elements and feedback loops displays emergent properties or behaviour. Such 'emergent behaviour' is not in practice predictable in advance from the characteristics of the individual elements. Nor is it predictable from an understanding of the relationships between any two elements. This appears to apply to combat."

Brigade operations include thousands of entities. In an entity-level simulation each of these entities must have a position, a state that includes combat mode or rules of engagements (ROE), and a plan or mission to execute. In addition each of these entities has the possibility of interacting with, or affecting, all the other entities. The number of possible states for the system as a whole thus becomes inconceivably large. The same applies to the number of possible outcomes.

3.16 Determining the exact outcome of a battle

We are sometimes met with the expectation that our simulations are perfect representations of reality, and therefore will determine the exact outcome of a battle. However, due to its high complexity, "[w]ar does not seem to be strongly determined; least of all in its human aspects" [5].

As we argued in Chapter 3.15, the number of possible outcomes of a battle is inconceivably large. "The number of possible states of a system so complex as an army is inconceivably large. The number of possible states for two armies locked in combat (with thousands of soldiers fighting thousands of their enemies, in ways that can vary considerably) is bigger yet. The actual and perceived complexity of a battle will be utterly inconceivable, and our ability to predict the precise outcome to it will be effectively nil. There will in reality be only one outcome to a battle, but the probability of predicting with precision what that outcome will be is one, divided by the number of possible outcomes: an inconceivably slim chance" [5].

Theoretically, war may be determined, but in the foreseeable future it is not useful to consider it so. This idea is summarized as follows by Jim Storr in [5]: "For all we know, war may be determined. It might even be possible to predict human behaviour from biochemical processes. To do so would require immense computing power and knowledge of the precise state of every participant down to the molecular level at the start of the battle. However, that would be extremely difficult to measure, and may well be unknowable. It is unlikely that even the most powerful computer will be able to achieve the necessary precision in the foreseeable future. Thus the likelihood of being able accurately to predict the outcome of a small engagement, let alone a war, is vanishingly small. To that extent *war may be determined but it is not useful to consider it so.*"

3.17 Simulation support to operations

Finally in this chapter, we will look at how combat simulations can be used to directly support military operations. Direct simulation support has the potential to increase the combat effectiveness, but so far these possibilities have only been exploited to a very limited extent. Possible simulation-based capabilities that can directly support operations are [157][159]:

- Battlefield visualization [59]
- Simulation for mission rehearsal [158]
- Simulation for course-of-action (COA) development/analysis [159][160]

In order for simulations to reach their full potential to support military operations, purpose-built simulation tools must be created [157]. Requirements for such simulation tools are:

- They must be easy to operate and require few operators. The simulations should run with minimal human interaction [157].
- They must be easy to initialize [157], for example from a C2 system using MSDL and C-BML.
- They should be integrated and compatible with C2 systems, and use the same terrain databases as the C2 systems [161].
- They must run hundreds of times faster than real-time, to be able to analyse multiple possible COAs within short deadlines [157][159].
- They should be able to run on a single computer [157] (e.g. in a mobile command post).

Simulations are validated for a specific purpose, so one important question is: how accurate do simulations have to be in order to be useful in an operational environment? The initial state of simulations will nevertheless always be based on the perceived situation, which can differ significantly from the real situation, especially regarding the opposing force.

For interactive mission rehearsal to be useful, high accuracy of the terrain database for the mission area is important. "That means to be able to determine how far one can see and decide how to exploit minor differences in elevation, density in foliage, ditches, culverts, and details in these objects" [158].

In COA analysis and planning, simulations are capable of taking into consideration a wider range of factors than human planners [159]. Also, "after prolonged periods of stress and fatigue, human decision makers may begin to forget details or make mistakes" [157].

4 Simulation tools

Most commercial combat simulation tools are primarily created for training purposes. In this chapter we first look at the components of a typical simulation tool. Then we briefly describe some of the simulation tools that are most widely used today. These simulation systems can be categorized as aggregate- or entity-level constructive simulations, or as virtual simulations. Some simulation systems also include support for both virtual and constructive simulation. Finally in this chapter we provide a graphical overview of the suitability of the different simulation systems for different use cases.

4.1 The components of a simulation tool

Interactive simulation tools or systems for virtual or constructive simulation will usually include most of the following components:

- An user interface (including input devices like mouse and keyboard, or more advanced input controllers like joysticks or steering wheels and pedals)
- A main simulation loop
- A model library (including simulation models and 3D models)
- A graphics engine (often also including a scene graph[6])
- A physics engine
- A sound engine
- An AI engine
- A networking interface
- A data logger

In addition a simulation tool will require a scenario and an environment/terrain database. Figure 4.1 illustrates the components that are typically found in a simulation system. Some of these components may also be implemented as external services.

The main simulation loop executes the simulation. For each round, the main simulation loop typically increases time with one *time step*/*simulation step*, or *tick*, and updates the state of the simulated objects and units. In some simulation systems (using a *fixed time step* simulation model) the tick is the smallest time unit of the simulation, and everything happening within the same tick is regarded as happening simultaneously. In other simulation systems (using an *event-to-event* simulation model) a time-stamp is assigned to each event in the simulation, and for each stimulation step the simulation time is advanced to the next event. Depending on the complexity of the scenario and the degree of human interaction, the length of a simulation step may vary from a few milliseconds in virtual simulations, to several seconds in some constructive simulations. In virtual simulations, visualization of fast-moving objects like projectiles typically requires high simulation update rates.

---

[6] A scene graph is a general data structure which arranges the logical and often spatial representation of a scene.

A simulation system may also include functionality or additional tools for creating scenarios, generating environment/terrain databases, and performing after-action review (AAR). Some simulation tools also include programming or scripting interfaces for developing additional functionality. Examples are VBS Fusion, which is a C++-based application programming interface (API) for VBS, and the VR-Forces software development kit (SDK). The ability to develop additional functionality is often very important in simulations used for experimentation purposes.

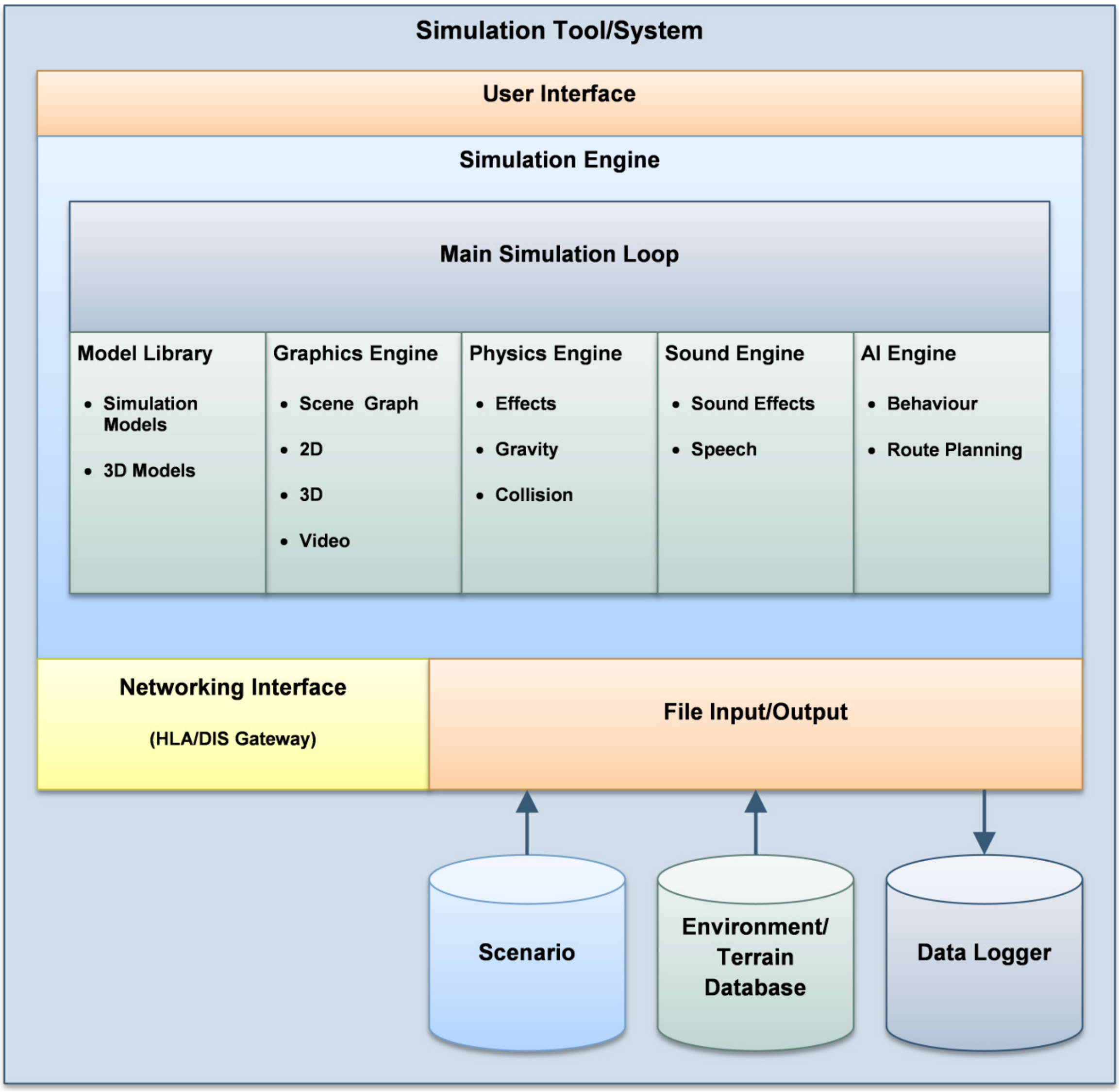


Figure 4.1 Typical components of an interactive combat simulation tool.

4.2 GEfechts-SImulation System (GESI)

GEfechts-SImulation System (GESI) is a constructive combat simulation system developed by CAE. GESI is designed to run exercises ranging from company to division level, both in CAX and classroom training environments. It is mainly used within the land domain, but can also include air and naval units. The most common use case is command staff and decision making training [162].

Several interfaces have been developed for GESI, and do now include, but are not limited to: Multi-Lateral Interoperability Program (MIP), HLA, Extensible Markup Language (XML), and Keyhole Markup Language (KML). These interfaces allow GESI to connect with C2 systems, exercise management and control systems, as well as other simulation federations [163].

The current version of GESI is version 7.0. Version 8.0 is expected to be released late 2015. Figure 4.2 shows the GESI workstation in use during an exercise. Table 4.1 summarizes the main characteristics of GESI [162][163].

Figure 4.2 The GESI workstation in use during an exercise (CAE).

Type of simulation	Constructive
Level of simulation	Aggregate-level
Simulation speed	Real-time and faster than real-time
Terrain representation	Hexagonal grid
User display	2D and partially 3D (only in observation mode)
Main area of use	Training
Domain	Land

Table 4.1 Main characteristics of GESI.

4.3 Joint Theater Level Simulation (JTLS)

The development of Joint Theater Level Simulation (JTLS) began in 1983 funded by U.S. Readiness Command, the U.S. Army War College, and the U.S. Army Concepts Analysis Agency. JTLS was created and developed by ROLANDS & ASSOCIATES Corporation. JTLS is used in well over 20 nations across NATO, including Norway (Norwegian Joint Headquarters (FOH) and FFI) [164][165].

JTLS is an interactive, internet-enabled simulation tool that models multisided air, ground, and naval civil-military operations with logistical, special operations forces (SOFs), and intelligence support. The simulation also supports limited nuclear and chemical effects, low-intensity conflict, pre-conflict operations, as well as support for humanitarian aid and disaster relief operations. The primary focus of the JTLS system is conventional joint and combined operations at the operational level of war [165].

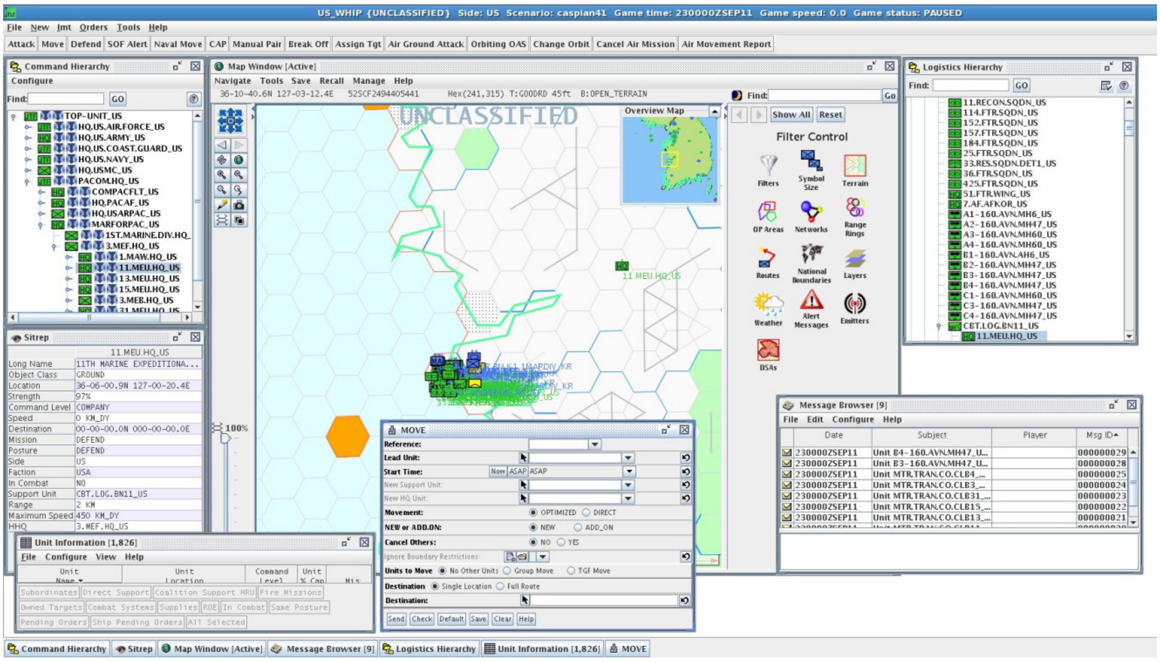
JTLS is designed for use in the following areas [164]:

- Analysis, development, and evaluation of contingency plans and joint tactics
- Evaluation of alternative military strategies
- Analysis of combat unit structure with respect to assigned combat systems
- Training of military joint staff headquarters

The JTLS system consists of a range of simulation models for different military operations. The land warfare module uses a Lanchester attrition methodology to aggregate the effects of direct fire, where the adjudication is deterministic. Air-to-air and surface-to-air are modelled on the entity level where the adjudication is stochastic. Air-to-ground and fire support use both approaches. In air-to-ground and fire support, area effects are modelled stochastically and precision munitions are modelled deterministic. JTLS also includes functionality and models for detailed logistics simulations [164].

The terrain used in JTLS is based upon hexagonal tiles. The environmental characteristics of the terrain are aggregated in each tile. The Compressed ARC Digitized Raster Graphics (CADRG) maps used in JTLS permits the map area to cover the entire world. Roads, railroads, and rivers are represented as network overlays on the hexagon terrain [164].

The current version of JTLS is version 4.0, with an upcoming release of version 5.0 mid 2015. Figure 4.3 shows the JTLS user interface. Table 4.2 summarizes the main characteristics of JTLS [164][165].



US_WHIP {UNCLASSIFIED} Side: US Scenario: caspian41 Game time: 230000ZSEP11 Game speed: 0.0 Game status: PAUSED

36-10-40.6N 127-03-12.4E 52SCF2494405441 Hex(241,315) T:GOODRD 45ft B:OPEN_TERRAIN

Sitrep

	11.MEU.HQ_US
Long Name	11TH MARINE EXPEDITIONA...
Object Class	GROUND
Location	36-06-00.9N 127-00-20.4E
Strength	97%
Command Level	COMPANY
Speed	0 KM_DY
Destination	00-00-00.0N 000-00-00.0E
Mission	DEFEND
Posture	DEFEND
Side	US
Faction	USA
In Combat	NO
Support Unit	CBT.LOG.BN11_US
Range	2 KM
Maximum Speed	450 KM_DY
HHQ	3.MEF.HQ_US

Message Browser [9]

Date	Subject	Player	Msg ID
230000ZSEP11	Unit B4-160.AVN.MH47_U...		000000029
230000ZSEP11	Unit B3-160.AVN.MH47_U...		000000028
230000ZSEP11	Unit MTR.TRAN.CO.CLB4_...		000000025
230000ZSEP11	Unit MTR.TRAN.CO.CLB3_...		000000024
230000ZSEP11	Unit MTR.TRAN.CO.CLB31_...		000000023
230000ZSEP11	Unit MTR.TRAN.CO.CLB15_...		000000022
230000ZSEP11	Unit MTR.TRAN.CO.CLB13_...		000000021

MOVE

Reference:
Lead Unit:
Start Time: ASAP
New Support Unit:
New HQ Unit:
Movement: ● OPTIMIZED ○ DIRECT
NEW or ADD.ON: ● NEW ○ ADD_ON
Cancel Others: ● NO ○ YES
Ignore Boundary Restrictions:
Units to Move ● No Other Units ○ Group Move ○ TGF Move
Destination ● Single Location ○ Full Route
Destination:

Figure 4.3 The JTLS user interface (U.S. Department of Defense).

Type of simulation	Constructive
Level of simulation	Aggregate-level
Simulation speed	Real-time and faster than real-time
Terrain representation	Hexagonal grid
User display	2D
Main area of use	Training, research and analysis, experimentation
Domain	All

Table 4.2 Main characteristics of JTLS.

4.4 MASA SWORD

MASA SWORD is a constructive simulation system for multi-level command post training and course of action (COA) analysis. The system is designed to run training exercises ranging from battalion level through corps, including functionality for logistics and communication. The unit behaviour is automated for platoon- and company-level units, so the operator can focus on higher-level orders. The behaviour system can be tailored to fit specific doctrines. MASA SWORD is mainly used in the land domain, but do include elements from both naval and air [166].

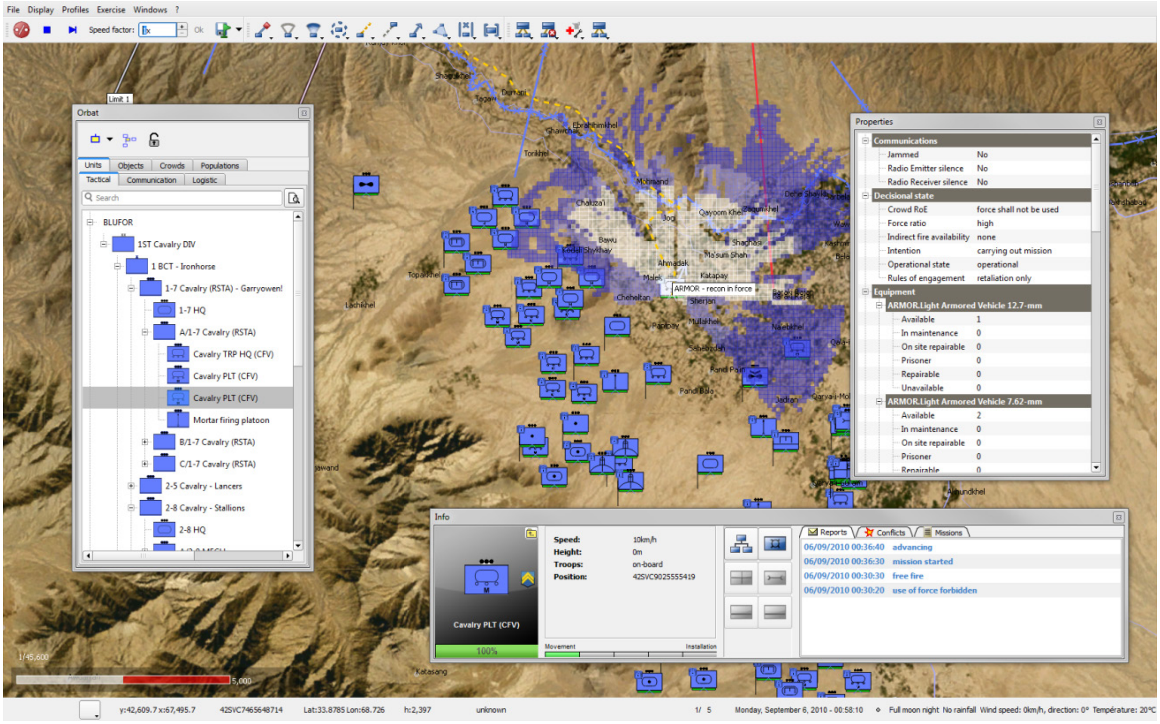
In addition to traditional military operations, MASA SWORD also includes functionality for OOTW and low intensity conflicts, such as CBRN incidents, counter terrorism, natural disasters, and civilian and refugee management. There is also functionality for human factors, such as morale, fatigue, and experience, which evolve during the course of an exercise [166].

The typical use cases for MASA SWORD are large-scale military exercises, C2 stimulation, officer training, and doctrine or equipment analysis.

The software is mainly based on open architectures and common communication formats, therefore compliant with HLA and DIS. MASA SWORD also supports the following terrain formats [167]:

- Altimetry (e.g. DTED and USX)
- Vector (e.g. VMAP, ESRI Shapefiles, and ESRI ArcSDE Database)
- Raster (e.g. USRP and GeoTiff)
- GDAL/OGR library

The latest version of MASA SWORD is version 6.2.1. Figure 4.4 shows the MASA SWORD user interface. Table 4.2 summarizes the main characteristics of MASA SWORD [167][166].



Communications
- Jammed: No
- Radio Emitter silence: No
- Radio Receiver silence: No

Decisional state
- Crowd RoE: force shall not be used
- Force ratio: high
- Indirect fire availability: none
- Intention: carrying out mission
- Operational state: operational
- Rules of engagement: retaliation only

Equipment
- ARMOR.Light Armored Vehicle 12.7-mm
  - Available: 1
  - In maintenance: 0
  - On site repairable: 0
  - Prisoner: 0
  - Repairable: 0
  - Unavailable: 0
- ARMOR.Light Armored Vehicle 7.62-mm
  - Available: 2
  - In maintenance: 0
  - On site repairable: 0
  - Prisoner: 0
  - Repairable: 0

Speed: 10km/h
Height: 0m
Troops: on-board
Position: 42SVC9025555419

06/09/2010 00:36:40 advancing
06/09/2010 00:36:30 mission started
06/09/2010 00:30:30 free fire
06/09/2010 00:30:20 use of force forbidden

Figure 4.4 The MASA SWORD user interface (MASA Group).

Type of simulation	Constructive
Level of simulation	Aggregate-level
Simulation speed	Real-time and faster than real-time
Terrain representation	Hexagonal grid
User display	2D
Main area of use	Training, research and analysis, experimentation
Domain	Land

Table 4.3 Main characteristics of MASA SWORD.

4.5 One Semi-Automated Forces (OneSAF)

One Semi-Automated Forces (OneSAF) is an entity-level, constructive simulation system initially developed in the 1990s by U.S. Army Simulation, Training and Instrumentation Command (STRICOM), currently known as Program Executive Office for Simulation, Training and Instrumentation (PEO STRI), to support the future needs for training, experimentation, and acquisition activities. It has since been incorporated in several virtual trainers, such as the Aviation Combined Arms Tactical Trainer (AVCATT), the Non-Rated Crew Member Manned Module (NCM3), and Close Combat Tactical Trainer (CCTT). OneSAF can be used in different domains, ranging from training to analysis and research [168][169].

OneSAF is interoperable with standards such as DIS, HLA, MSDL, Joint Consultation Mission Command Information Exchange Data Model (JC3IEDM), and Army Mission Command Systems (AMCS). OneSAF also supports a wide range of terrain database formats [168].

The current version of OneSAF is version 8.5. Figure 4.5 shows an image of the user interface of OneSAF (to the left), and OneSAF in use (to the right). Table 4.4 summarizes the main characteristics of OneSAF [168][169].

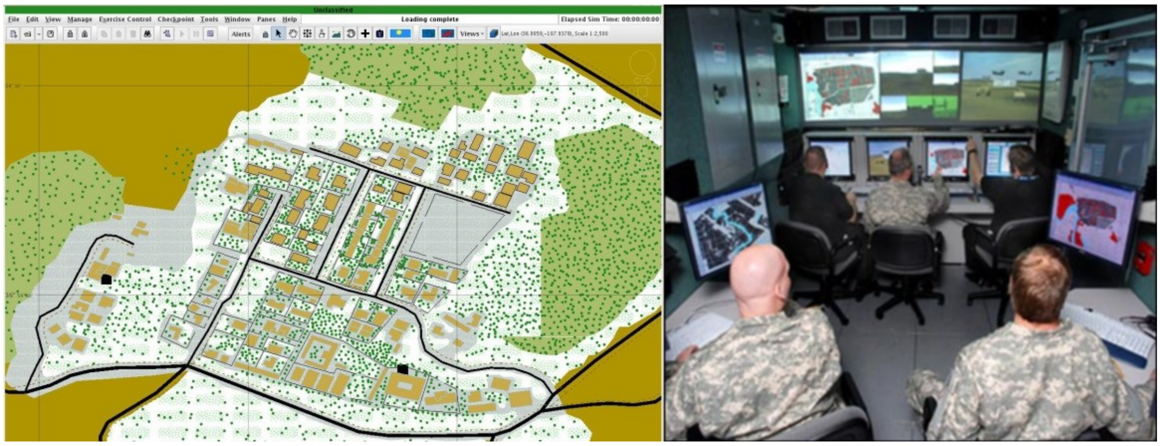


Figure 4.5 An image of the user interface of OneSAF (to the left), and OneSAF in use (to the right) (Calytrix/Terrasim, U.S. Army).

Type of simulation	Constructive
Level of simulation	Entity-level
Simulation speed	Real-time and faster than real-time
Terrain representation	N/A
User display	2D (3D with extension software)
Main area of use	Training, research and analysis, experimentation
Domain	Land

Table 4.4 Main characteristics of OneSAF.

4.6 Steel Beasts

The initial development of Steel Beasts started in 1995 by eSim Games as a virtual simulator for fire control systems of armoured fighting vehicles. It has since developed to a product line called Steel Beasts Professional. eSim Games describes Steel Beasts Professional as a vehicle-centric combined arms combat simulator, typically used by crew members and commanders of different armoured vehicles. Steel Beasts Professional supports a wide range of fire control systems, including the CV90 family, Leopard 2, M1 Abrams, and M2 Bradley, amongst others. The system includes models for ballistics and damage calculations for the included vehicles [170].

Steel Beasts is designed primarily as a lower-level skills trainer, for example to improve gunnery skills, crew coherence, and improve the vehicle commander tactical decision-making. Steel Beasts can be run either as a desktop training solution or be integrated as a module in a containerized or stationary hardware simulator system [170][171].

The system also includes support for third party terrain generation, currently through Re-Lion Builder [172] and TerraTools 5 [173]. Steel Beasts Professional is also compliant with the most common interoperability standards such as DIS and HLA.

The current version of Steel Beasts Pro is version 3.0. Figure 4.6 shows an image of the Steel Beasts tactical interface (to the left), and a virtual representation of the gunner position in a combat vehicle (to the right). Table 4.5 summarizes the main characteristics of Steel Beasts [170][171].

Figure 4.6 An image of the Steel Beasts tactical interface (to the left), and a virtual representation of the gunner position in a combat vehicle (to the right) (eSim games).

Type of simulation	Virtual
Level of simulation	Entity-level
Simulation speed	Real-time
Terrain representation	Polygon mesh
User display	2D and 3D
Main area of use	Training
Domain	Land

Table 4.5 Main characteristics of Steel Beasts.

4.7 Virtual Battlespace (VBS)

Virtual Battlespace (VBS) is a virtual simulation tool, based on game technology, developed by Bohemia Interactive Simulations (BISim). VBS is widely used within NATO as a tool for game-based virtual training on tactical level. Virtual Battlespace 3 (VBS3) is the most recent release of the product line, and has recently been chosen as one of the tools in the U.S. Army Games for Training program [174][175].

The most common use is land-based lower-level tactical training and mission rehearsal from squad to company level. In recent years there has also been major development on naval functionality, enabling users to do shoreline and naval mission training. In addition to lower-level tactical training and rehearsal, VBS is also used as an IG [176] and as a tool for CD&E and analysis. Due to the upcoming release of VBS Tactics [177], a user interface for control of doctrinally correct SAF in VBS, it will also be feasible to use VBS as a command and staff training tool.

VBS is delivered with a large library of content funded by different nations over the years. Version 3.7 of VBS contains nearly 300 tracked vehicles, about 100 planes and unmanned aerial systems (UASs), and nearly 1,000 wheeled vehicles. There are also several high detailed terrain databases available [175].

The system features possibilities for accurate ballistics and detailed damage calculations for vehicles and personnel. The system is also extendible by using VBS Fusion, a C++ application programming interface (API) that gives developers access to the system core. Several third party applications have been developed to support specific use cases [175][178].

In the more recent versions of VBS, terrain databases are generated with a different approach than traditional simulation tools. By utilizing technologies for procedurally generated environments, the previous limitation on terrain database size is increased. According to BISim, they expect to be able to stream worldwide high detailed playable areas with the technology developed in the VBS Blue project [179].

VBS is also compliant with the common interoperability standards DIS, HLA 1.3, HLA 1516, and HLA evolved [175].

The current version of the system is VBS version 3.7. Figure 4.7 shows an image from VBS3 (to the left), and an image of VBS3 (VBS IG) being used as an image generator for a flight simulator (to the right). Table 4.6 summarizes the main characteristics of VBS [175].

Figure 4.7 An image from VBS3 (to the left), and an image of VBS3 (VBS IG) being used as an image generator for a flight simulator (to the right) (Bohemia Interactive Simulations).

Type of simulation	Virtual
Level of simulation	Entity-level
Simulation speed	Real-time
Terrain representation	Polygon mesh
User display	3D
Main area of use	Training
Domain	All

Table 4.6 Main characteristics of VBS.

4.8 VR-Forces

VR-Forces is a CGF framework developed by VT MÄK. VR-Forces comes with a set of capabilities that enables the user to create, execute, and distribute simulation scenarios. The system is designed to be used in a wide variety of areas, like a tool for training and mission rehearsal, a synthetic environment for experimentation, or an engine to stimulate C4I systems [180][181].

The most common use cases are command and staff training, CGF for different C2 systems, pre-deployment training, and a tool for different analysis purposes. VR-Forces is also used for civilian purposes, like air traffic management, and emergency response preparedness.

The VR-Forces system includes simulation models for hundreds of battlefield units and systems, in all domains (land, naval, air, and space). The system can be used in aggregate- or entity-level mode, which enables the user to do simulations of large combat operations or single unit tactics. The system also features a wide variety of unit commands, such as moving to waypoints, following specific routes, or more complicated tasks such as sector search and rescue (SAR). The user can also extend the commands with more complex tasks through scripting [181].

According to VT MÄK, VR-Forces is described as *terrain agile*. This means that it is capable of using a multitude of terrain formats and terrain loading strategies. The VR-Forces toolkit supports all of the common terrain formats, including terrain streaming. The VR-Forces terrain databases can be a mix of different approaches, which enables the users to cover vast areas in different resolutions. There is also a library of different terrain databases included in the baseline distribution of VR-Forces, including a simplified database of the whole world. The terrain databases in VR-Forces can also include procedurally generated dynamic environment effects, such as dynamic ocean effects, varying light conditions, and accurate atmospheric and weather effects. All of these dynamic effects impact the system's sensor models [181].

The VR-Forces framework is customizable, and provides several C++ APIs for different development tasks. There are also available a scripting interface in Lua for higher-level access [181].

In addition to VR-Forces, VT MÄK develops several other products that could be included as extensions to VR-Forces. The current product portfolio consists of different interoperability applications, artificial intelligence modules, sensor and visualization modules, and terrain streaming servers [180].

VR-Forces is compliant with the standards DIS, HLA 1.3, HLA 1516, and HLA Evolved through VR-Link, a built in interoperability module. The current version of VR-Forces is version 4.3.1. Figure 4.8 shows the VR-Forces user interface in 3D view. Table 4.7 summarizes the main characteristics of VR-Forces [181].

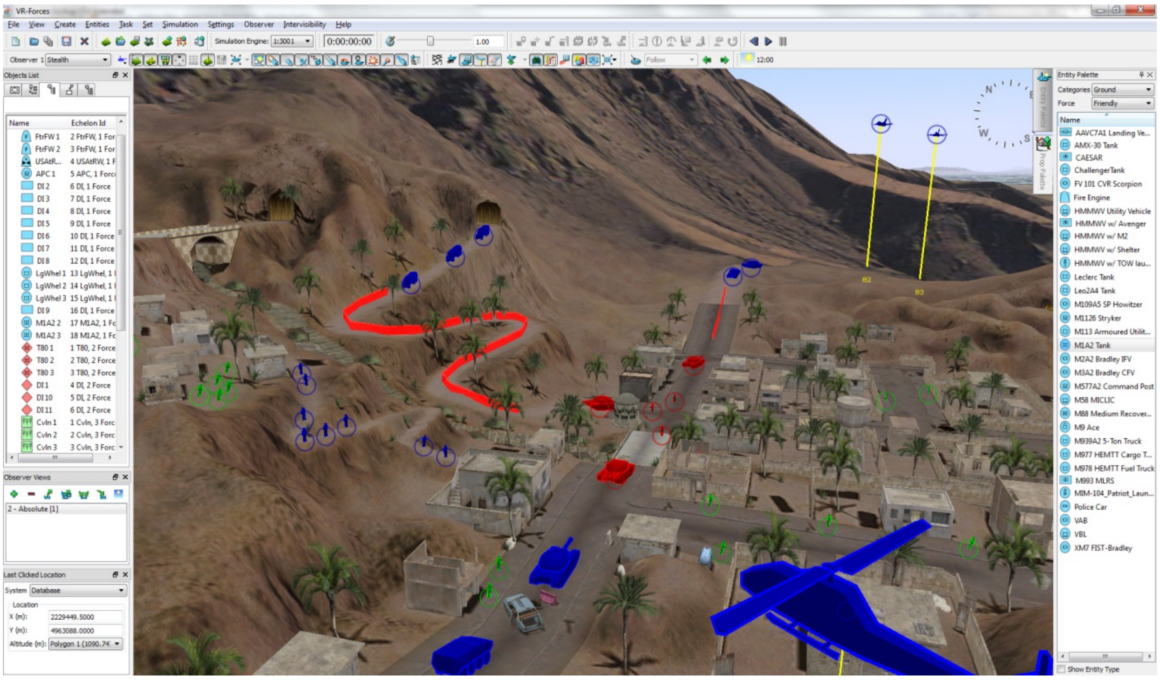


Figure 4.8 The VR-Forces user interface in 3D view. The viewport is highly customizable, and can be used in both 2D and 3D (VT MÄK).

Type of simulation	Constructive and virtual
Level of simulation	Entity-level and aggregate-level
Simulation speed	Real-time and faster than real-time
Terrain representation	Polygon mesh
User display	2D and 3D
Main area of use	Training, research and analysis, experimentation
Domain	All

Table 4.7 Main characteristics of VR-Forces.

4.9 Simulation system comparison tables

This chapter gives a short graphical overview of the suitability of the different systems mentioned in this chapter for different use cases. This is based both on experience and developer documentation. In cases where we lack hands-on experience, it is purely based on documentation such as manuals or product presentations. The systems that are based only on documentation are marked by a star (*). Table 4.8 lists the different levels of suitability used to rank the simulation systems, ranging from good representation (green circle) to no representation (red square).

Color	**Combat model**	**System hierarchy**
🟢	Good representation	Well suited for
🟡	Fairly good representation	Fairly well suited for
🟠	Limited representation	Poorly suited for
🟥	No representation	Not suited for

Table 4.8 The different levels of suitability used to rank the simulation systems, ranging from good representation (green circle) to no representation (red square).

4.9.1 Virtual simulation system suitability

There are only two simulators mentioned in this chapter that resides in the virtual simulator category. Comparison tables for the systems that have functionality for both virtual and constructive simulation are found in Chapter 4.9.2.

Virtual Simulator Combat Models		Combat Manoeuvre	Fire Support	Combat Engineering	Air Defence	Aviation	C2	ISTAR	Communication	Medical	Maintenance	Electronic Warfare	CBRN
	Steel Beasts	🟢	🟡	🟥	🟥	🟠	🟡	🟥	🟥	🟥	🟠	🟥	🟥
	VBS	🟢	🟡	🟠	🟠	🟡	🟠	🟡	🟠	🟡	🟠	🟥	🟥

Virtual Simulator Command Level		Single Unit	Squad	Platoon	Company	Battalion	Brigade	Regiment/Division	Corps
	Steel Beasts	🟢	🟢	🟢	🟢	🟡	🟠	🟥	🟥
	VBS[7]	🟢	🟢	🟢	🟢	🟡	🟠	🟥	🟥

---

[7] Included in this estimation is the extension VBS Tactics, which enables battalion and brigade behavior functionality. Without this extension, both battalion and brigade will have limited functionality.

4.9.2 Constructive simulation system suitability

The following tables include the constructive simulation tools mentioned in this chapter.

	Combat Manoeuvre	Fire Support	Combat Engineering	Air Defence	Aviation	C2	ISTAR	Communication	Medical	Maintenance & logistics	Electronic Warfare	CBRN
GESI	🟢	🟡	🟡	🟠	🟠	🟠	🟠	🟠	🟠	🟠	🟠	🟠
JTLS*	🟠	🟡	🟡	🟡	🟡	🟡	🟡	🟡	🟡	🟡	🟡	🟠
MASA SWORD	🟢	🟡	🟡	🟡	🟠	🟡	🟠	🟠	🟡	🟢	🟡	🟡
OneSAF*	🟢	🟡	🟡	🟡	🟠	🟡	🟠	🟠	🟡	🟡	🟡	🟠
VR-Forces[8]	🟢	🟡	🟡	🟡	🟠	🟡	🟡	🟡	🟡	🟡	🟠	🟠

	Single Unit	Squad	Platoon	Company	Battalion	Brigade	Regiment/ Division	Corps
GESI	🟠	🟡	🟡	🟢	🟢	🟡	🟠	🟥
JTLS*	🟥	🟥	🟥	🟠	🟡	🟢	🟢	🟢
MASA SWORD	🟠	🟡	🟢	🟢	🟢	🟡	🟠	🟥
OneSAF*	🟡	🟢	🟢	🟢	🟢	🟡	🟠	🟥
VR-Forces	🟡	🟡	🟡	🟢	🟢	🟡	🟠	🟥

---

[8] This estimate is based on the entire VR-Forces product line. The VR-Forces baseline itself may not be as suitable as illustrated in the table. The different levels can be achieved with different extension modules available from VT MÄK.

5 Examples of simulation experiments conducted at FFI

In this chapter we look at two examples of how simulation of land force operations has been used for experimentation and analysis at FFI. In the first example we have used virtual simulations to experiment with, and evaluate the operational benefit of, an augmented reality (AR) system for combat vehicles. In the second example we have used constructive simulations to support land force structure analysis by experimenting with, and comparing the performance of, a set of fundamentally different land force structure concepts.

5.1 FFI's battle lab facility

FFI’s battle lab facility was opened in 2005. It offered new possibilities for experimentation with new technologies and new concepts, and is particularly suitable for conducting human-in-the-loop (HITL) simulation experiments [182]. The battle lab has become an important arena for collaboration between various projects at FFI, and between scientists/engineers and military personnel. The two simulations that are described in Chapter 5.2 and Chapter 5.3 are examples of simulations that have been conducted in FFI's battle lab.

5.2 Evaluating the operational benefit of augmented reality (AR)

At FFI we have been experimenting with simulated augmented reality (AR) since 2006 [1][2]. We have been using virtual simulations to evaluate the operational benefit of AR functionality in combination with a battlefield management system (BMS) in combat vehicles. So far we have carried out three experiments with professional combat-vehicle crews playing through a set of scenarios. The size of the experiments has ranged from platoon to company level, and each of them lasted for one week.

AR is a technology for real-time mixing of virtual, computer-generated data with data we perceive from the real world. This gives the user an augmented perception of reality. Mainly, AR means adding virtual objects, in the form of computer graphics, to visual data from the real world. The virtual objects typically provide information in a way that improves the user’s situational awareness (SA), thus helping him or her to perform real-world tasks better. We have developed a *simulated* AR system which adds graphical AR objects to a virtual scene instead of the real-world.

The simulated AR system [178] is designed for use in combat vehicles like infantry fighting vehicles (IFVs) and main battle tanks (MBTs). It works in conjunction with a BMS, and visualizes information like blue-force tracking (BFT), observations, and waypoints, in the form of graphical objects displayed directly in the sights and periscopes of the commander, gunner, and driver. This enables the vehicle crew to exploit the BMS information without taking their eyes off what is going on in the battlefield. The AR system also makes the BMS information more intuitive, and increases the vehicle crew's overall SA.

We have also integrated a laser range finder (LRF) with the BMS. When this is triggered, the target position is sent to the BMS, and this makes it possible to select positions directly from the terrain. Moreover, basic input to the BMS can be given as simple voice commands, to provide a handsfree user interface.

For the first experiment, which was conducted in 2006, we developed a simple combat vehicle simulator based on UT2004. In the two subsequent experiments, which were conducted in 2008 and 2011, we used VBS as simulation system. The AR system in VBS was developed using VBSFusion, which is a C++-based application programming interface (API) for VBS.

Figure 5.1 illustrates an AR object used in the system. All AR objects have the same structure, and consist of the following five components:

1. A symbol that shows the AR object's affiliation and type. We have used symbols from the MIL-STD-2525C standard for military map marking symbols [183].
2. A unique text string that represents the AR object's ID.
3. A number giving the distance in meters from the vehicle to the AR object.
4. A dot that represents the actual position of the AR object on the ground. This dot is in white colour if the vehicle has line of sight to the AR object's position; otherwise it is in red colour.
5. A vertical bar connecting the dot and the symbol. The bar has the same colour as the symbol, in accordance with the AR object's affiliation.

To avoid too much cluttering, the AR system has a minimum and a maximum distance for when AR objects are shown. Preferences like transparency, size, and whether or not the AR objects should be scaled with distance, are set in a configuration file for the AR system.

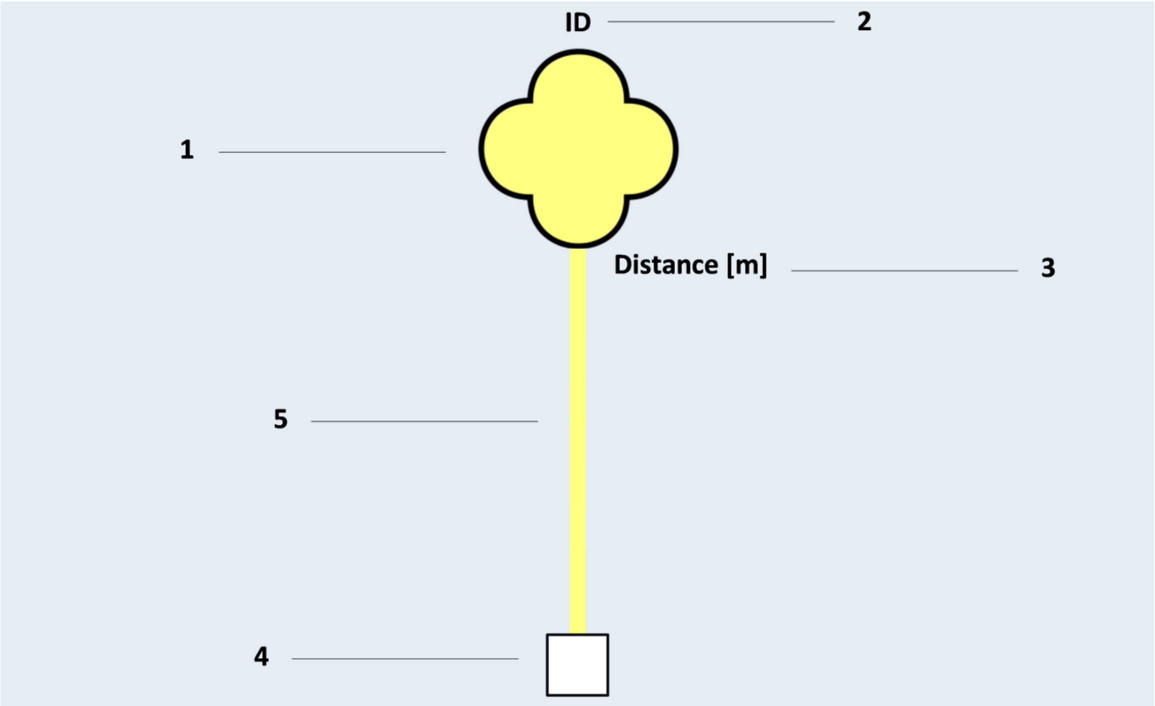


Figure 5.1 The five components of an AR object ([178]).

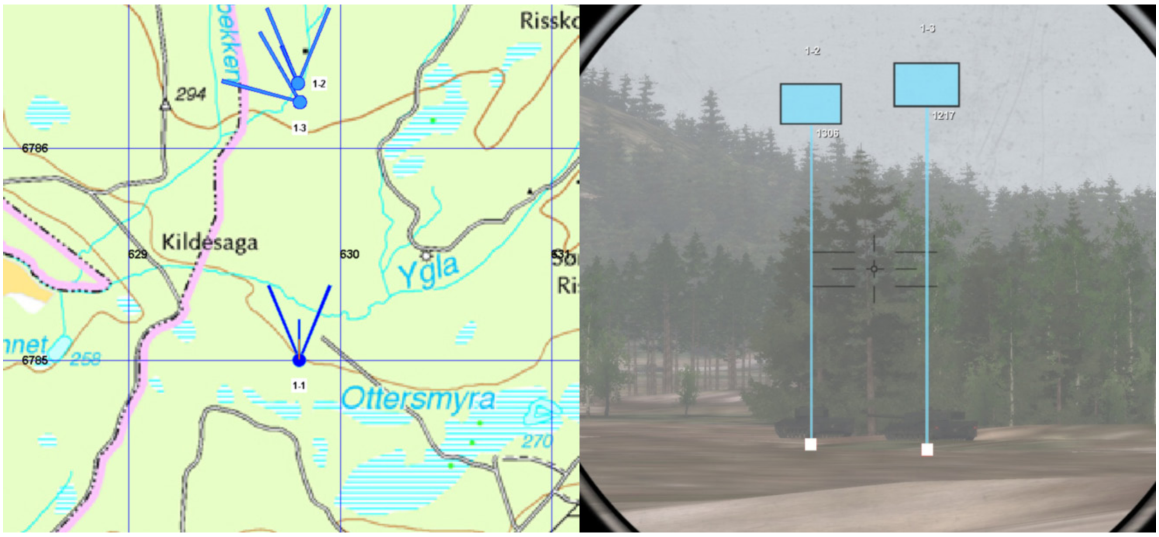


Figure 5.2 Blue-force tracking on the BMS screen (to the left) and through the commander's sight (to the right) ([178]).

Figure 5.2 shows an example with BFT symbols drawn on the BMS screen (to the left), and the virtual scene viewed through the vehicle commander's sight with AR objects marking the blue vehicles (to the right). The information is shown from the perspective of the vehicle with ID 1-1, looking at two friendly vehicles with IDs 1-2 and 1-3. On the BMS screen the blue dots mark the vehicles' position, the short blue lines mark the vehicles' direction, and the pairs of two long blue lines in a "V"-shape mark the gunners' viewing sectors.

It is possible to mark positions of interest by adding observations in the BMS. Observations can be assigned an affiliation and a type. Figure 5.3 shows four observations displayed on the BMS screen (to the left), and the corresponding image seen from vehicle 1-1 commander's sight with AR objects (to the right). The observations have unique two-letter IDs.

Figure 5.3 Observations on the BMS screen (to the left) and through the commander's sight (to the right) ([178]).

Figure 5.4 Operators and components of the simulated combat vehicle (to the left), and a simulated combat vehicle in use (to the right) ([1][2]).

The operators and components of the simulated combat vehicle are shown in Figure 5.4 (to the left). The BMS was placed between the gunner and commander. The driver was given a screen displaying the BMS image, but did not have the ability interact with it. Figure 5.4 (to the right) shows a picture of a simulated combat vehicle in use during an experiment.

We have used two types of scenarios in the experiments. In the first type the blue force was given a mission, and could freely choose how to solve it. This allowed us to observe how the participants adapted to the new technology. The data collected from these scenarios were feedback from the participants through questionnaires and after-action review sessions, and general observations of how the system was used. In the second scenario type the participants were given a specific task, like performing an attack by fire on an enemy position, or locating a specific target in the terrain. The data collected from these scenarios where quantitative performance data. Moreover, all executions of the scenarios were logged and recorded on video. All scenarios were completed both with and without AR, for comparison.

The experiments showed that the AR system results in faster and more accurate perception of the BMS information, and thus better overall SA. In small test scenarios we observed an average reduction of up to two thirds in target acquisition times.

The general idea behind this work has been to test new technologies or new concepts in a virtual environment by developing virtual prototypes. Virtual prototypes make it possible to experiment with new technology and new concepts in situations that are hard or impossible to achieve in the real world because of cost, safety or availability issues. With virtual prototypes it is also possible to involve the users at an early stage of the development process.

Parallel to the experimentation with the virtual prototype, there has been a project for developing a real-world prototype of the new technology. There have been several iterations with further development of the real-world prototype, followed by new experiments with a more refined version of the virtual prototype. Figure 5.5 illustrates the concept behind this approach.

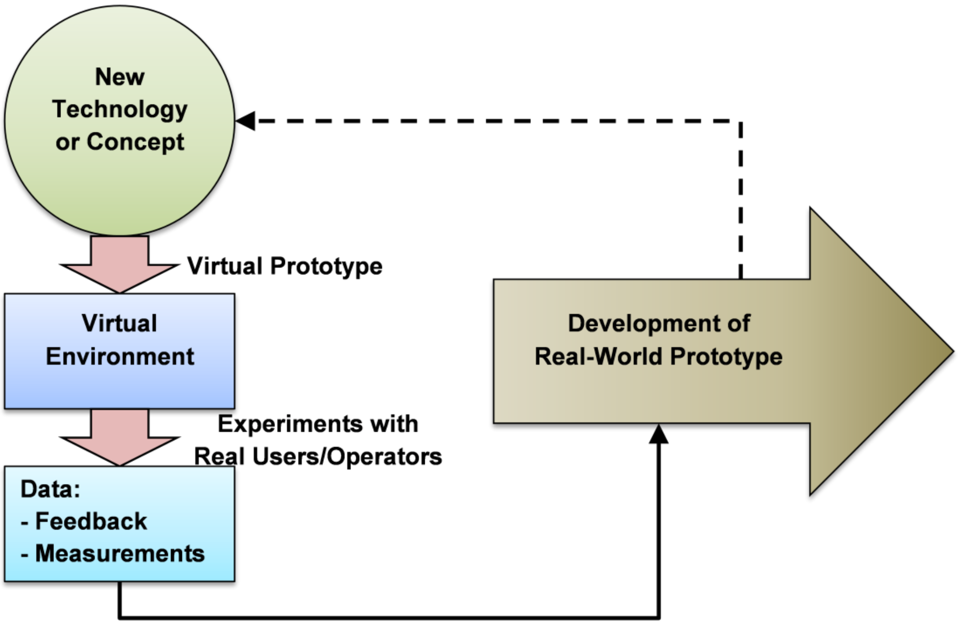


Figure 5.5 General method for experimenting with new technologies or new concepts, by using virtual environments ([2]).

5.3 Land force structure analysis

In 2009 FFI-project "Future Land Forces" was initiated, with the goal to analyse future requirements for the military land power in a national, allied, and multinational context. The main objective was to ensure cohesion and balance between resources and requirements in the development of military structures. In this project interactive constructive simulations were carried out to support evaluation of alternative land force structures [3][4]. Through a series of experiments the performance of five fundamentally different land force structures was tested in a set of chosen scenarios. The goal of the experiments was to rank these structures based on their relative performance.

Prior to the experiments, a number of different land force structures were developed [184] based on three war fighting concepts: manoeuvre theory, exchange theory, and positional theory [185]. Most of these structures where filtered out according to a capability-based method. Finally, we chose to evaluate the performance of five land force structures. Three of these structures were mechanized manoeuver structures. In addition we tested a light structure with units equipped with man-portable antitank weapons, and a distributed manoeuver structure largely based on network-centric warfare [186] and long-range precision-guided fire. The five land force structures were tested in three chosen tactical vignettes. We used a fixed opposing force for the experiments, which was based on a generic mechanized infantry brigade. Figure 5.6 illustrates this whole process.

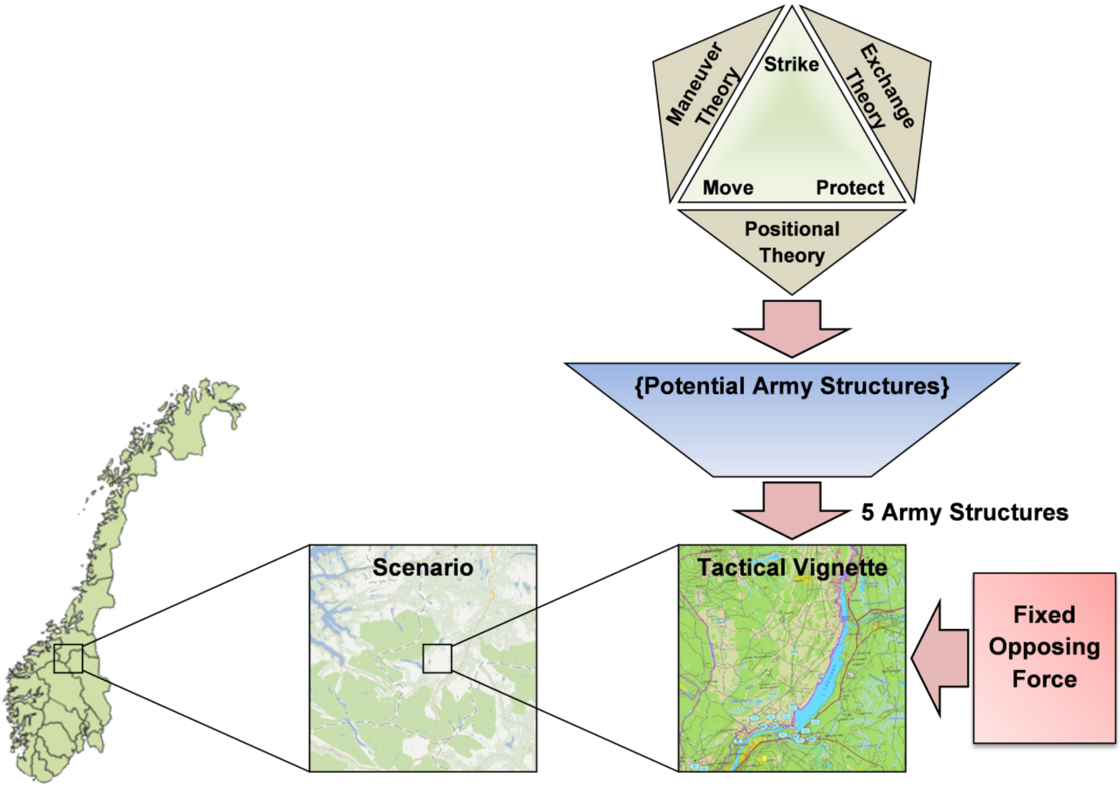


Figure 5.6 Land force structure analysis at FFI ([4]).

The performance of the land force structures was evaluated through a set of simulation experiments. The approach was to use interactive constructive simulations with semi-automated forces (SAF), where humans were in the loop as military commanders to control the course of the battle. The main advantage of this type of simulation is utilization of human creativity, decision-making, and their ability to find solutions along the way. Military commanders plan and control the operations in the simulation, while the simulation system keeps track of the movement of the units and calculates the results of duels and indirect-fire attacks. This approach can be described as computer-aided wargaming. Figure 5.7 illustrates the concept behind this approach.

The simulation tool mōsbē from BreakAway was used in the experiments. The main reason for this choice was that mōsbē is based on technology for real-time strategy games, and has a user interface that makes it easy to control large groups of entities. Figure 5.8 shows examples of a two-dimensional theater view (to the left), and a three-dimensional tactical view (to the right) in mōsbē. The development of mōsbē has been discontinued, and the latest version was released in 2008.

Before the experiments, virtual representations of all units in the potential force structures had to be created. This included calibration of sensor capabilities and damage calculations. However, all simulation systems have shortfalls. To overcome these, we worked closely together with military SMEs.

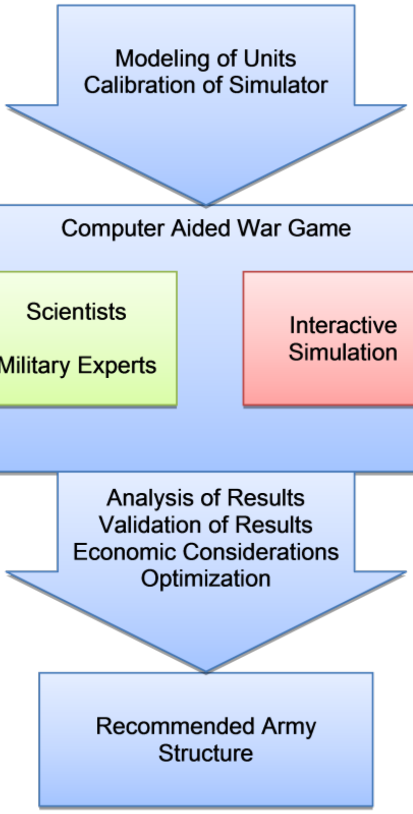


Figure 5.7 Simulation to support land force structure analysis ([3][4]).

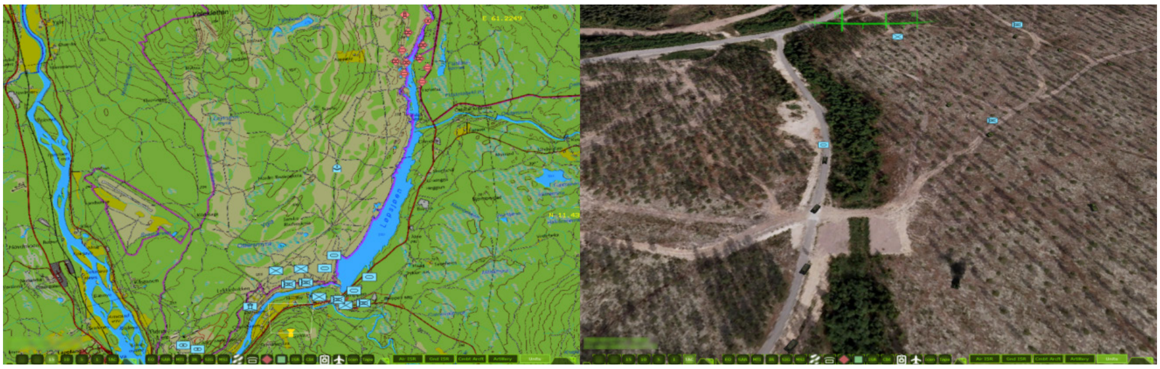


Figure 5.8 Two-dimensional theater view (to the left), and three-dimensional tactical view (to the right) in mōsbē ([3][4]).

A total of 14 simulation clients were used in the experiments. Four of these clients were used for controlling red forces, while up to six clients were used by the blue players. The remaining four clients were reserved for the white cell. The white cell functioned as administrators and umpires, and handled issues not represented in the simulation system. Figure 5.9 shows a typical allocation of players on blue side. Red side was organized in a similar manner, with one player controlling the air units, one player controlling artillery, and two players controlling the manoeuvre battalions. I addition to the players operating the simulation clients, both sides had a brigade commander leading the battle.

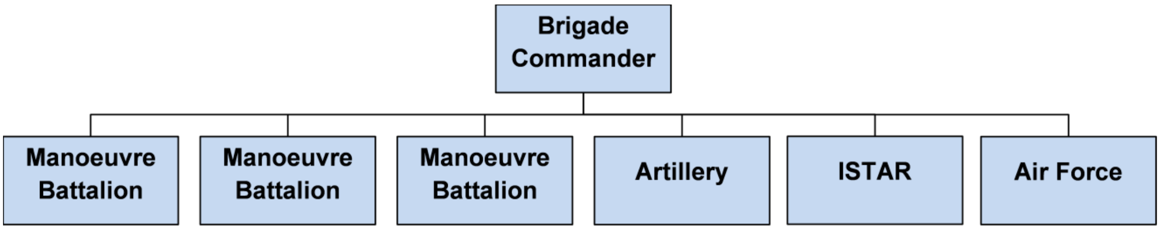


Figure 5.9 Example of the distribution of players on blue side ([4]).

Each simulation experiment began with a separate military planning session for each side, where the course of action (COA) was discussed and chosen. The simulation typically lasted between four and six wall clock hours, and was stopped when one side undoubtedly was unable to achieve its goal. After each simulation experiment there was an after-action review (AAR) session. This included discussions and evaluations regarding events on both tactical and operational levels. The combined-arms effects were typically evaluated through the AAR. Figure 5.10 shows a picture from a simulation session (to the left), and a picture from a planning session (to the right).

The main categories of collected data from the experiments were answers from questionnaires, experiences revealed during AAR sessions, and log files generated by the simulation system. The administrators also recorded video, took screenshots, and documented relevant events from the simulation sessions. Questionnaires were used to reveal each participant's perception about the different force structures, both in terms of their expectations before the execution and afterwards, based on the outcome of the battle.

After the experiments the results were analysed and used in a larger context together with considerations about economy and force production. They were combined with outputs from the KOSTMOD[9] cost model, and a model estimating the total force production, to arrive at land force structures that are effective both in battle and in terms of cost. Finally, this work resulted in a set of recommendations for potential new structures for the Norwegian military land power.

Figure 5.10 A simulation session (to the left) and a planning session (to the right) during the land force structure analysis at FFI ([3][4]).

---

[9] KOSTMOD is the main cost model used in Norway and at FFI for long term defence planning.

6 Lessons learned and recommendations for the future

In the previous chapter we described two examples of simulation experiments that we have conducted in FFI's battle lab facility. In this chapter we summarize the experiences and lessons learned from our simulation experiments. We also give some recommendations for future simulations of land force operations.

6.1 Virtual versus constructive simulation

As we have mentioned earlier, we have generally used virtual simulation in experiments where human system operators are essential (for example when experimenting with technology that directly affects human performance or how humans operate). The size of these experiments has been limited to a few platoons (reduced company level). To simulate operations at the battalion and brigade level we have used constructive simulation for conducting interactive war games with SAF. The virtual simulations can be very realistic, and include detailed models of equipment and systems, but the size of the operations that can be simulated are very limited. It is not feasible to simulate brigade-size operations, and thus include all the aspects of larger operations (e.g. combined arms and joint operation synergies), using virtual simulation. Constructive simulations can be used to simulate large operations, but they currently lack realistic representations of low-level tactics. There are mainly two factors that have the potential of improving the fidelity of our constructive simulations. The first factor is increased terrain resolution. The second factor is better tactical AI that can take advantage of this terrain.

BISim recently announced that they are developing a new tool named VBS Tactics [177]. VBS Tactics will be a tool for conducting constructive, entity-level simulations in VBS. The objective of VBS Tactics is to provide a doctrinally correct tactical AI for VBS, which can be controlled via an easy-to-use interface. With the use of VBS Tactics and VBS we hope to establish a capability for more realistic simulation of brigade-size land force operations for analysis and capability planning. This solution will have the high terrain resolution that is available in VBS, and a tactical AI that hopefully will be able to take advantage of this terrain. VBS in combination with VBS Tactics will also make it easier to combine virtual and constructive simulations, for example by including one or two virtual platoons (operated from VBS) in a brigade-size constructive simulation (controlled from VBS Tactics).

6.2 Experimenting with new technologies or concepts is an iterative process

When experimenting with new technologies, new defence structures, or new operational concepts, an iterative process is necessary. It is important with a sufficient number of iterations, so that the participants learn how to use the new technology or force structure in an optimal manner. When experimenting with new force structures and new operational concepts, it is also important that the structure and operational concept of the opposing force is adapted to provide effective and realistic resistance. The iterative competition between the blue and red forces will ultimately ensure that both force structures are employed in an optimal manner.

6.3 Conducting simulation experiments

When conducting simulation experiments we recommend including military SMEs and experiment operators/participants in all the stages of the experiments. Early involvement of military SMEs, starting with the planning and preparations before the experiments, gives transparency to stakeholders. By participating in the planning of the experiments, and the calibration of the simulations, they gain insight into how the simulation system works, including its limitations. At the same time military SMEs are important for the validation of the simulation system, and the credibility of the results.

After conducting several simulation experiments, we have learned that time spent on preparation and testing is essential for a successful experiment execution. It is important with extensive testing with the same number of operators/participants and simulated units as will be used in the experiments, to generate the same workload on the computers and the same amount of network traffic, for identifying, and if possible fix, any problems and bottlenecks.

From time to time we see that some operators/participants try to “game the system” by exploiting weaknesses in the simulation system to obtain unrealistic advantages. As we have mentioned earlier, it is therefore important with administrators and umpires with in-depth knowledge of the simulation system, who can monitor the simulation experiments.

Carrying out experiments in virtual environments with virtual prototypes of new technologies has proved to be very useful, both for evaluating operational benefit and for improving design and functionality in the development phase. Even with fairly simple simulators, it is possible to evaluate operational benefit of a system. If the purpose of the simulation is to compare similar systems (e.g. in an acquisition process), it is important that the simulation system and the virtual prototype have a resolution and fidelity that is high enough to capture the differences between the evaluated systems.

6.4 Developing simulation components

We mainly use commercial off-the-shelf (COTS) simulation tools for our simulations, but we often need to develop additional functionality in the form of plug-ins[10] or scripts that are used by the simulation tools. Such software components are usually developed to be used in a certain experiment, and they are not regularly maintained after the experiment.

Major upgrades to the simulation tools will often require the plug-ins or scripts to be updated, and this can be a time consuming process. In addition, when simulation tools are replaced the plug-ins or scripts can no longer be used. In the future we therefore recommend that new simulation functionality, if possible, is developed as independent HLA compatible services. Only functionality that has to be tightly coupled to a simulation tool should be implemented as a plug-in. This practice will also be in compliance with the M&S as a service (MSaaS) concept (see Chapter 7.1.2).

---

[10] A plug-in is a software component which adds a specific feature to an existing application.

6.5 Summary of recommendations and requirements for future simulations

We conclude this chapter by summarizing the most important recommendations and requirements for our future simulations of land force operations:

- Virtual simulation is suitable for experiments where human system operators are essential, for example when experimenting with technology that directly affects human performance or how humans operate. However, virtual simulation requires many operators, and the size of the operations that can be simulated are limited.
- Constructive simulations with SAF are suitable for simulating large operations, but they currently lack realistic representations of low-level tactics.
- We need to increase the terrain resolution and represent micro-terrain features in our constructive simulations. Low terrain resolution, and lack of micro-terrain features and realistic vegetation density, systematically favours simulated platforms with long weapon and sensor ranges.
- We need to represent weather effects in the simulations. Weather effects may significantly affect military operations [92].
- For our constructive simulations we need better tactical AI that behaves according to blue and red doctrine, and are able to intelligently take advantage of the terrain. We need to be able to simulate brigade-size operations using no more than five to six operators on each side. This means that one operator should be able to control a manoeuvre battalion. The SAF should be fully automated below platoon level for vehicles, and below squad level for infantry.
- We need to include better representations of combat service support, which includes combat engineering, logistics and supply, maintenance, and medical. Combat service support has so far largely been handled by the white cell.
- We need to represent communication effects. So far we have assumed perfect communication between units.
- We recommend constantly seeking to improve the fidelity of our simulations by always employing state-of-the-art simulation tools and technology (for both virtual and constructive simulation).
- If possible, new simulation components should be implemented as independent HLA compatible services.
- Open standards should always be favoured. The latest version of HLA should be used for communication between simulation components. For communication between platforms and systems in the simulation the same standards and protocols that are used by the real-world systems should be used, if possible.

Our plan for the future is to establish a new capability for constructive, entity-level simulation of brigade-size land force operations utilizing VBS, VBS Tactics, and VR-Forces, possibly supplemented by independent HLA compatible services. This capability will also include the possibility of mixing virtual and constructive simulation.

7 Latest trends and speculations about the future

In this chapter, which is at the very end of this survey, we take a look at some of the latest trends in military M&S. We will also present some speculations on how we think combat simulation will evolve in the near future.

7.1 Latest trends

To get professional input and to keep ourselves updated on the latest trends in military M&S, we regularly attend defence-related M&S conferences like the Interservice/Industry, Training, Simulation and Education Conference (I/ITSEC), the International Training and Education Conference (ITEC), and the NATO Modelling and Simulation Group (NMSG) annual symposium. In the chapters 7.1.1 to 7.1.4 we briefly describe some of the latest trends in military M&S.

7.1.1 Web technology for M&S

In the recent years we have seen an increasing use of web technology for M&S. Especially, the new version of the HyperText Markup Language (HTML), HTML5 (which was finalized in October 2014), provides new abilities for creating interactive web-interfaces for simulations and games. In addition, webGL (Web Graphics Library) provides the possibilities for creating interactive GPU-accelerated two- and three-dimensional graphics in web-applications. The process of converting information or applications from their original form to something that can be accessible through a web-browser is sometimes referred to as *webifiaction*.

The biggest advantage of using web technology for simulation is that the simulation system is accessible, and can be interacted with, through a web-browser. This means that no simulation software needs to be installed on the client computers.

WebLVC [187] is a protocol for enabling interoperability between web-based applications and the standards for distributed simulation, DIS and HLA. The WebLVC protocol represents the semantics from DIS and HLA using messages in the JavaScript Object Notation (JSON) format, which are typically sent between server and client using WebSockets. Figure 7.1 illustrates how the WebLVC architecture works. WebLVC is currently in the process of being further developed to become a SISO standard.

WebLVC is well suited for *M&S as a service* (MSaaS) (see 7.1.2). It guarantees transparency about the final standard used on the cloud side, and can potentially be used for any possible standard and architecture [188].

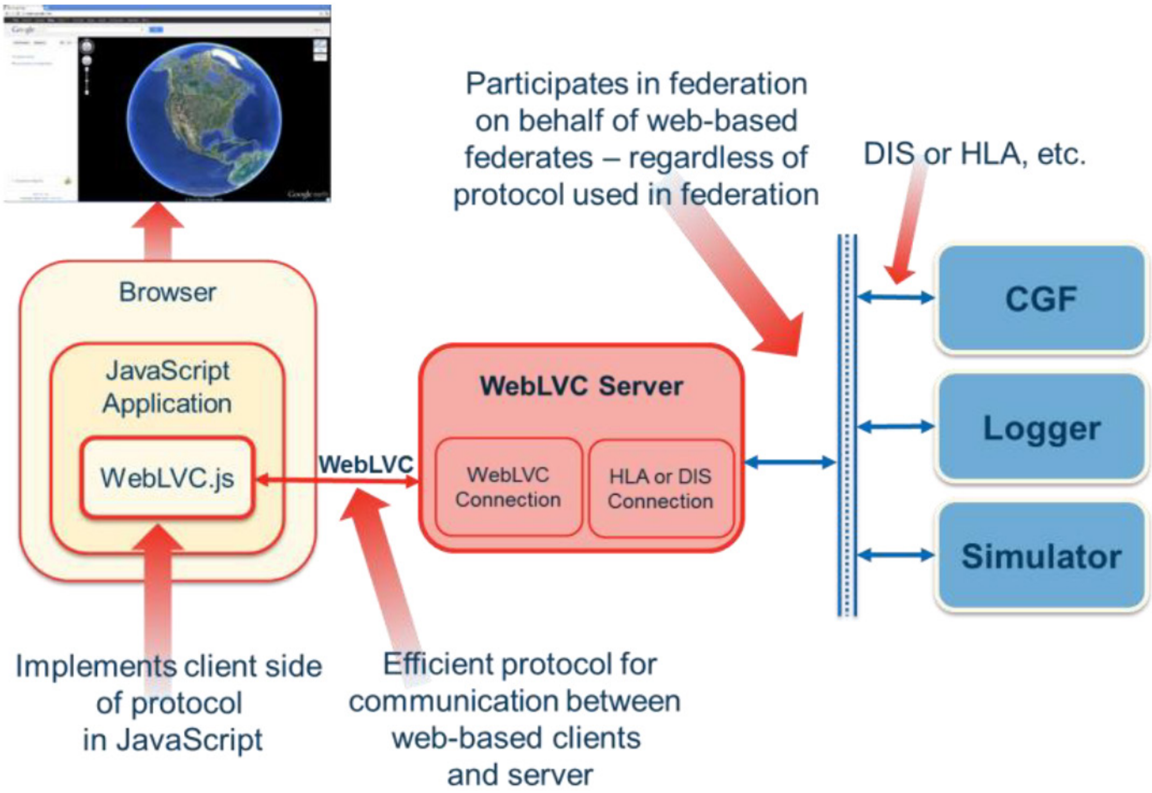


Figure 7.1 WebLVC enables interoperability between web-based applications and the standards for distributed simulation, DIS and HLA (VT MÄK [187]).

7.1.2 Modelling and simulation as a service (MSaaS)

M&S as a service (MSaaS) is the combination of *service-based approaches* for M&S with ideas taken from *cloud computing* [98][117]. The vision of MSaaS is to “offer users M&S solutions wherever they are and whenever they need them” [188].

The National Institute for Standards (NIST) defines *cloud computing* as: “a model for enabling ubiquitous, convenient, on-demand network access to a shared pool of configurable computing resources (e.g. networks, servers, storage, applications, and services) that can be rapidly provisioned and released with minimal management effort or service provider interaction” [189]. The NATO Modelling and Simulation Group (NMSG) research task group NMSG-131, “Modelling and Simulation as a Service: New Concepts and Service Oriented Architectures”, defines MSaaS as follows: “*M&S as a Service* (MSaaS) is a means of delivering value to customers to enable or support modelling and simulation (M&S) user applications and capabilities as well as to provide associated data on demand without the ownership of specific costs and risks” [117][118]. “MSaaS is an architectural and organizational approach that promotes abstraction, loose coupling, reusability, composability and discovery of M&S services” [117][118]. “The objective of MSaaS is to effectively and efficiently support operational requirements (e.g. executing an exercise) and to improve development, operation, and maintenance of M&S applications” [118].

Offering MSaaS has the potential of providing several benefits relative to the way things are done today [190]:

- Centralized hosting of M&S resources may reduce the cost of ownership by reducing licensing requirements, hardware and software maintenance/upgrades costs, and facility resources.
- With flexible and scalable M&S environments, setting up new users and exercise and experimentation environments may be performed more quickly and at lower costs. In addition, the M&S environment can scale according to need (by increasing in size when expanded capability is needed and decreasing when needs are reduced).
- MSaaS will provide an environment that is device and location independent, and facilitate increased accessibility of the M&S resources.
- MSaaS may facilitate increased collaboration amongst the users of the M&S environment. Updates to the M&S resources will provide all the users with access to the same capabilities.

Examples of potential M&S services are [116][117][118]:

- Weapons Effects Service (WES)
- Communication Effects Service (CES)
- Exterior Ballistics Service (EBS)
- Synthetic Environment Service (SES)
- Synthetic Dynamic Environment Service (SDS)
- Initialization Service
- Computer-Generated Forces (CGF) Service
- Weather Simulation Service
- Terrain Generation Service
- Verification and Validation (V&V) Service

Figure 7.2 shows examples of M&S services in the System Demonstrator Distributed and Integrated Test Bed (SD VIntEL[11]), which is a Research and Development (R&D) project in the German Army [116].

Some examples of open issues with regards to MSaaS identified in NMSG-131 are [117][118]:

- How to specify technical services, especially with regards to non-functional requirements?
- How to establish permanent/persistent services (in NATO)?
- What are the requirements on governance, and how should service management be organized?

---

[11] In German: Verteilte Integrierte Erprobungslandschaft (VIntEL)

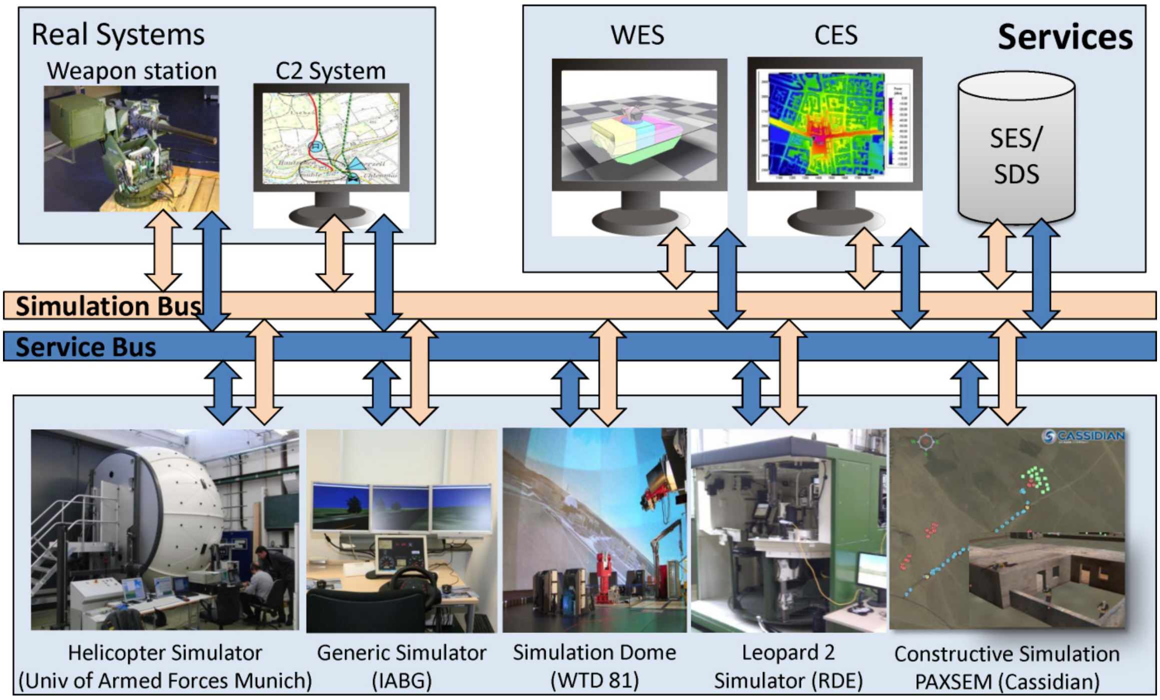


Figure 7.2 Examples of M&S services in SD VIntEL (aditerna GmbH).

A service oriented M&S environment requires that the simulation applications need to satisfy the following requirements [191]:

- Use services instead of internal algorithms: Simulation systems need to be prepared to use external services (e.g. to use damage results due to weapon effects calculated elsewhere). If a simulation system is designed to work in a non-service environment and has built-in algorithms (e.g. for damage calculations) these must be switched off.
- Harmonize conceptual models: Simulation systems and M&S services need harmonized conceptual models (e.g. same number and understanding of different damage levels). Without a common understanding service usage may be possible on a technical level, but meaningful interoperability on higher levels (i.e. on pragmatic level) is not possible.

As Richbourg et al. point out in [97], a major concern is that "[i]f these [service oriented] models don't share a common environment, common sensor physics, and common target detection and acquisition models (i.e., the underlying world processes and data), then they will have the same interoperability problems as our current federations without the people in the loop to solve them".

7.1.3 Early synthetic prototyping (ESP)

Early synthetic prototyping (ESP) is a new concept the U.S. Army is exploring that will use simulation systems based on game technology to assess novel system designs and concepts early in the acquisition cycle by developing virtual prototypes. ESP enables warfighters to assess emerging technologies within scenarios to provide feedback that will inform decisions. Additionally, ESP can be used to explore force design and force employment in conjunction with capability development at the operator and small unit levels [192]. Through a persistent distributed game network, ESP will offer a collaborative environment for warfighters, scientists, acquisition professionals, and decision makers. Instrumented scenarios will be used to collect data from gameplay for evaluation to impact system development and refinement. “The goal is to engage the whole [U.S.] Army in defining the future of the [a]rmy and to ensure that the [s]oldier remains the centerpiece of future development” [192].

ESP differs from traditional simulation-based acquisition (SBA) in two important areas [192]:

1. ESP is focused on early concept development, when costs are relatively low but when it is critical to get major design decisions right.
2. ESP allows for the consideration of orders of magnitude more design options than SBA (or any known acquisition process). ESP operates on the premise that disruptive ideas are more likely to appear when 10,000 design variations are considered rather than just ten.

ESP involves the warfighters earlier in the design process, and shifts change demands earlier in the acquisition cycle, when they are less expensive to execute [192]. Figure 7.3 illustrates this.

A number of critical steps need to be taken before ESP can be fully realized [192]:

- The network must be distributed worldwide and playable by any authorized player.
- Scenario editing must be simple yet expressive.
- Game play must be easy and entertaining or players will simply spend their time doing something else.
- Players must be able to easily modify scenarios to create new designs and configurations.
- Gleaning useful information from game play must be simple, or better yet, transparent to players.

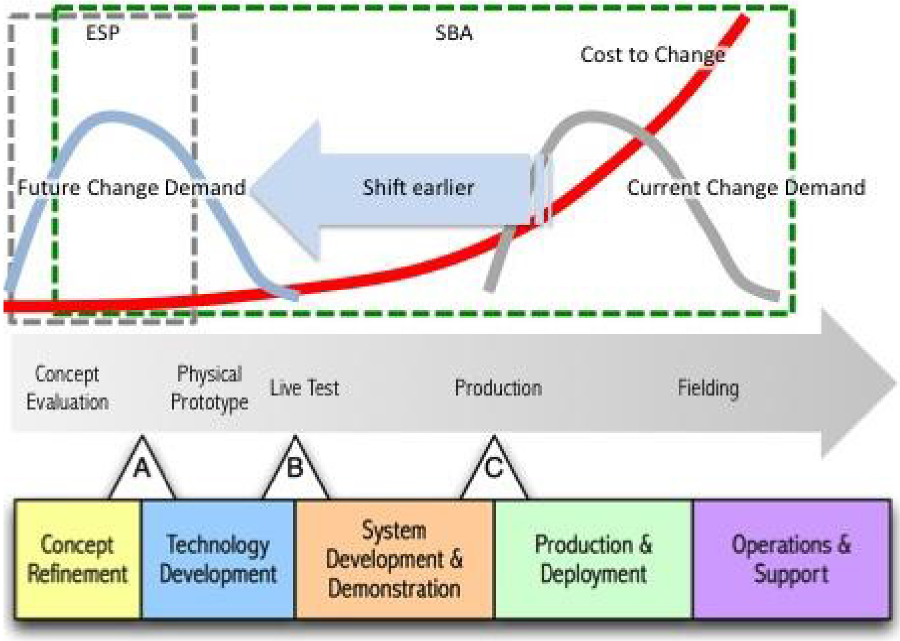


Figure 7.3 Early Synthetic Prototyping (ESP) shifts change demands earlier in the acquisition cycle, when they are less expensive to execute ([192]).

7.1.4 Procedural generation of environment data

In large, high-detailed virtual environments it is not feasible to store the exact positions of all natural objects like trees and bushes. Instead they can be *procedurally* or *algorithmically* generated by the simulation system at run-time, using a set of rules based upon terrain shape and land-cover material data. With this approach the placement of the vegetation will be completely deterministic [83]. If different simulation systems in a federation use the same rule set for generation of objects, and their terrain models are correlated, the objects will be positioned at the same places in all representations of the environment. Procedural techniques can also be used to generate micro-terrain features, land-cover material textures, and snow.

In computer games procedural techniques have been used to generate planets and entire universes. The upcoming computer game No Man's Sky (from Hello Games) will feature a procedurally generated universe with more than 18 trillion (10^{18}) planets [193].

7.2 Speculations about the future

In addition to the trends we have just described, there are several technologies we believe will evolve and have an impact on M&S for analysis and experimentation in the next five to ten years. In this chapter we will briefly describe some of these technologies.

7.2.1 Machine learning

Trainable AI for CGF that uses *machine-learning* techniques [137] will probably become available within the next ten years. Trainable automated forces (TAF) have the ability to learn from examples and experience.

TAF will potentially be very useful for experimenting with new and novel defence structures and operational concepts. Additionally, TAF will be useful for supporting course-of-action (COA) development in operations.

7.2.2 Real-time ray tracing

Real-time *ray tracing*[12] [194] will probably be feasible within the next ten years. This technology will potentially revolutionize how real-time computer graphics is generated in computer games, virtual simulation systems, and image generators. Figure 7.4 illustrates the difference between generating computer graphics by using traditional raster techniques (to the left), and by using ray tracing (to the right) [194].

Ray tracing techniques are not restricted to visual light, but may also be used for other areas of the electromagnetic domain, such as infrared light and radio waves using broadly the same algorithms. This will reduce the need for creating separate models and textures for different types of electromagnetic radiation [194].

---

[12] Ray tracing is a technique for generating an image by tracing rays through pixels on the screen and into the virtual scene.

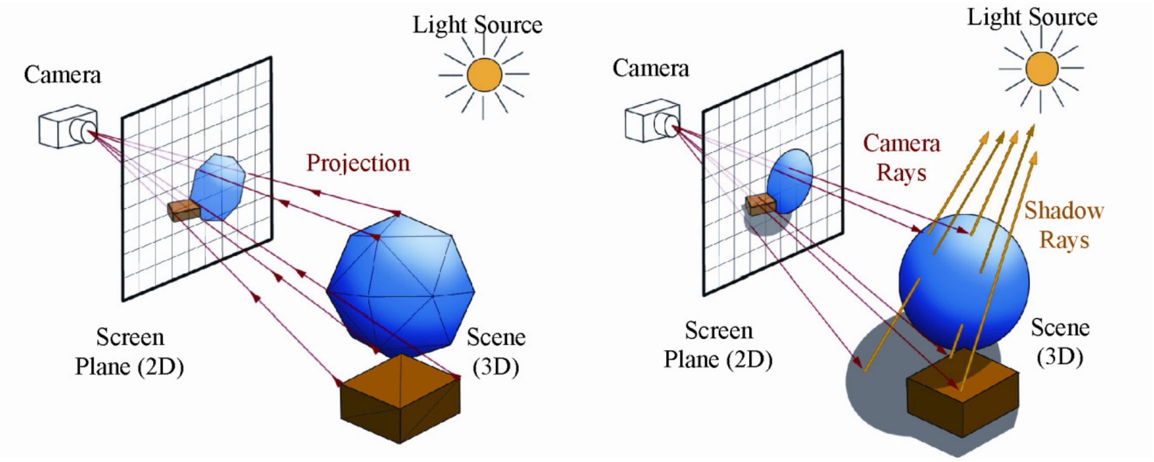


Figure 7.4 Generation of computer graphics with raster techniques (to the left), and ray tracing (to the right) ([194]).

7.2.3 More realistic real-time, physics-based material damage simulation

As we have mentioned several times in this report, real-time, physics-based material damage simulations are already feasible by employing methods for finite element analysis (FEA) and computational fluid dynamics (CFD) [66][90]. By taking advantage of the constantly increasing power of parallel computing, we expect these simulations to achieve much higher resolution, and become much more realistic in the next five to ten years.

8 Conclusion

In this report we have gone through different types of combat simulations (virtual and constructive), different levels of resolution for representing terrain and combat units (entity-level and aggregate-level), different techniques for modelling the four core activities of combat (moving, observing/sensing, shooting/engaging, and communicating), and different ways of representing human behaviour. We have also looked at some of the simulation tools that are most widely used today. Different simulation tools have different strengths and weaknesses, and often federations of multiple simulation tools and components have to be composed in order to meet specific simulation needs.

Combat is highly complex, especially in its human aspects. With the current methods and technologies for modelling and simulation of land force operations, it is possible to experiment with, and evaluate the potential operational benefit of, new technologies and new concepts. Furthermore, it is possible to experiment with different land force structures, reveal their strengths and weaknesses, and evaluate their relative operational performance by testing them against a fixed adversary. However, since combat is so complex, it will not be possible (at least not in the foreseeable future) to use simulations for determining the exact outcome of a battle.

The methods and technologies used for modelling combat are constantly evolving. From time to time it is useful to capture and document the state of the art of this domain. Not only for bringing this knowledge out to others, but also for us to get a better overall picture of where we stand today, and how we can improve the fidelity of our combat simulations in the future. It will therefore probably be a good idea to write an updated survey in about ten years, which will be around 2025.

References

[1] P-I. Evensen & M. Halsør, *Experimenting with Simulated Augmented Reality in Virtual Environments*, Interservice/Industry Training, Simulation and Education Conference (I/ITSEC) 2013, Paper No. 13028, 2013.

[2] P-I. Evensen & M. Halsør, *Using Virtual Environments to Evaluate the Operational Benefit of Augmented Reality*, NATO Modelling and Simulation Group (NMSG) Annual Symposium 2014 (STO-MP-MSG-126), Paper No. 5, 2014.

[3] E.Ø. Hoff, P-I. Evensen, H.R. Holhjem, I.B. Øyan & H.K. Nygård, *Simulation in Support of Army Structure Analysis*, Interservice/Industry Training, Simulation and Education Conference (I/ITSEC) 2012, Paper No. 12088, 2012.

[4] E.Ø. Hoff, P-I. Evensen, H.R. Holhjem, I.B. Øyan & H.K. Nygård, *Interactive Simulation to Support the Transition of Forces*, NATO Modelling and Simulation Group (NMSG) Annual Symposium 2013 (STO-MP-MSG-111), Paper No. 6, 2013.

[5] J. Storr, *The Human Face of War*, Continuum, 2009.

[6] United States Department of Defense (U.S. DoD) Modeling and Simulation Coordination Office (MSCO), *Department of Defense Modeling and Simulation (M&S) Glossary*, 2011.

[7] M. D. Petty & P. Gustavson, *Combat Modeling with the High Level Architecture and Base Object Model*, in *Engineering Principles of Combat Modeling and Distributed Simulation*, A. Tolk (edited by), John Wiley & Sons, 2012.

[8] J.A. Sokolowski & C.M. Banks (edited by), *Modeling and Simulation Fundamentals: Theoretical Underpinnings and Practical Domains*, John Wiley & Sons, 2010.

[9] J. Borah, *Simulation Concept Modeling – Theory and Application* (Tutorial Slides), Interservice/Industry Training, Simulation and Education Conference (I/ITSEC) 2012, Tutorial No. 1236, 2012.

[10] A. Tolk (edited by), *Engineering Principles of Combat Modeling and Distributed Simulation*, John Wiley & Sons, 2012.

[11] C.M. Banks, *Introduction to Modeling and Simulation*, in *Modeling and Simulation Fundamentals: Theoretical Underpinnings and Practical Domains*, J.A. Sokolowski & C.M. Banks (edited by), John Wiley & Sons, 2010.

[12] A.M. Law, *Simulaton Modeling and Analysis*, 4. Edition, McGraw-Hill, 2007.

[13] G.E.P. Box & N.R. Draper, *Empirical Model-Building and Response Surfaces*, John Wiley & Sons, 1987.

[14] M.D. Petty, *Verification, Validation, and Accreditation*, in *Modeling and Simulation Fundamentals: Theoretical Underpinnings and Practical Domains*, J.A. Sokolowski & C.M. Banks (edited by), John Wiley & Sons, 2010.

[15] A. Maria, *Introduction to Modeling and Simulation*, 1997 Winter Simulation Conference (WSC), 1997.

[16] L.G. Callahan & J.G Taylor, *Modeling and Simulation of Land Combat*, Georgia Tech Institute of Military Education, 1983.

[17] J.E. Hannay, O.M. Mevassvik, A. Skjeltorp & K. Bråthen, *LVC Simulation for Land Operations Training*, NATO Modelling and Simulation Group (NMSG) Annual Symposium 2014 (STO-MP-MSG-126), Paper No. 16, 2014.

[18] A. Ceranowicz, *Metasimulation*, Interservice/Industry Training, Simulation and Education Conference (I/ITSEC) 2014, Paper No. IF1401, 2014.

[19] Simulation Interoperability Standards Organization (SISO), http://www.sisostds.org.

[20] North Atlantic Treaty Organization (NATO) Standardization Office, *NATO Modelling and Simulation Standards Profile*, NATO Standard AMSP-01, Edition C, Version 1, 2015.

[21] Simulation Interoperability Standards Organization (SISO), *Guide for: DIS Plain and Simple*, SISO-REF-020-2007, 2009.

[22] D. Neyland, *Virtual Combat – A Guide to Distributed Interactive Simulation*, Stackpole Books, 1997.

[23] Institute of Electrical and Electronics Engineers (IEEE), *IEEE Standard for Distributed Interactive Simulation – Application Protocols*, IEEE Standard 1278.1-2012, 2012.

[24] Institute of Electrical and Electronics Engineers (IEEE), *IEEE Standard for Distributed Interactive Simulation – Communication Services and Profiles*, IEEE Standard 1278.2-2015, 2015.

[25] Institute of Electrical and Electronics Engineers (IEEE), *IEEE Recommended Practice for Distributed Interactive Simulation – Exercise Management and Feedback*, IEEE Standard 1278.3-1996, 1996.

[26] Institute of Electrical and Electronics Engineers (IEEE), *IEEE Recommended Practice for Distributed Interactive Simulation – Verification, Validation, and Accreditation*, IEEE Standard 1278.4-2003, 2003.

[27] F. Kuhl, R. Weatherly & J. Dahmann, *Creating Computer Simulation Systems – An Introdution to the High Level Architecture*, Prentice-Hall, 2000.

[28] Institute of Electrical and Electronics Engineers (IEEE), *IEEE Standard for Modeling and Simulation (M&S) High Level Architecture (HLA) – Framework and Rules*, IEEE Standard 1516-2010, 2010.

[29] Institute of Electrical and Electronics Engineers (IEEE), *IEEE Standard for Modeling and Simulation (M&S) High Level Architecture (HLA) – Federate Interface Specification*, IEEE Standard 1516.1-2010, 2010.

[30] Institute of Electrical and Electronics Engineers (IEEE), *IEEE Standard for Modeling and Simulation (M&S) High Level Architecture (HLA) – Object Model Template (OMT) Specification*, IEEE Standard 1516.2-2010, 2010.

[31] North Atlantic Treaty Organization (NATO) Standardization Office, *STANAG 4603 – Modelling and Simulation Architecture Standards for Technical Interoperability: High Level Architecture (HLA)*, NATO STANAG 4603, Edition 2, 2015.

[32] Simulation Interoperability Standards Organization (SISO), *Standard for Real-time Platform Reference Federation Object Model*, SISO-STD-001.1-2015, 2015.

[33] Object Management Group (OMG), *Data Distribution Service for Real-time System – Version 1.2*, 2007.

[34] Object Management Group (OMG), http://www.omg.org.

[35] R. Joshi & G-P. Castellote, *A Comparison and Mapping of Data Distribution Service and High-Level Architecture*, Real-Time Innovations Whitepaper, 2006.

[36] Simulation Interoperability Standards Organization (SISO), *Standard for Military Scenario Definition Language (MSDL)*, SISO-STD-007-2008, 2008.

[37] Simulation Interoperability Standards Organization (SISO), *Standard for Coalition Battle Management Language (C-BML) Phase 1*, SISO-STD-011-2014, 2013.

[38] J.M. Pullen, D. Corner, R. Wittman, A. Brook, O.M. Mevassvik & A. Alstad, *MSDL and C-BML Working Together for NATO MSG-085*, 2012 Spring Simulation Interoperability Workshop (SIW), Paper No. 045, 2012.

[39] K. Heffner, B. Gautreau, O.M. Mevassvik & R. De Reus, *A Proposed Process and Toolset for Developing Standardized C2-to-Simulation Interoperability Solutions*, NATO Modelling and Simulation Group (NMSG) Annual Symposium 2014 (STO-MP-MSG-126), Paper No. 2, 2014.

[40] R. Garrett, *To Believe or Not to Believe, Fidelity is the Question*, Interservice/Industry Training, Simulation and Education Conference (I/ITSEC) 2012, Paper No. 12425, 2012.

[41] R.G. Sargent, *Verification, Validation and Accreditation of Simulation Models*, 2000 Winter Simulation Conference (WSC), 2000.

[42] C. Dunn, *Face Validation: From Concept To Concrete Process*, Interservice/Industry Training, Simulation and Education Conference (I/ITSEC) 2012, Paper No. 12218, 2012.

[43] North Atlantic Treaty Organization (NATO) Science and Technology Organization (STO), *Generic Methodology for Verification and Validation to Support Acceptance of Models, Simulations, and Data*, STO Technical Report, STO-TR-MSG-073, 2015.

[44] Simulation Interoperability Standards Organization (SISO), *Guide for Generic Methodology for Verification and Validation (GM-VV) to Support Acceptance of Models, Simulations, and Data – GM-VV Volume 1: Introduction and Overview*, SISO-Guide-001.1-2012, 2012.

[45] Simulation Interoperability Standards Organization (SISO), *Guide for Generic Methodology for Verification and Validation (GM-VV) to Support Acceptance of Models, Simulations, and Data – GM-VV Volume 2: Implementation Guide*, SISO-Guide-001.2-2013, 2013.

[46] Simulation Interoperability Standards Organization (SISO), *Guide for Generic Methodology for Verification and Validation (GM-VV) to Support Acceptance of Models, Simulations, and Data – GM-VV Volume 3: Reference Manual*, SISO-Guide-039-2013, 2013.

[47] P.A. Birkel, *SNE Conceptual Reference Model*, 1998 Fall Simulation Interoperability Workshop (SIW), Paper No. 018, 1998.

[48] P. Muller, *The Future Immersive Training Environment (FITE) JCTD: Improving Readiness Through Innovation*, Interservice/Industry Training, Simulation and Education Conference (I/ITSEC) 2010, Paper No. 10148, 2010.

[49] P. Muller, R. Schaffer & J. Carswell, *The Future Immersive Training Environment JCTD: Technical Challenges in Demonstrating Virtual Reality for Infantry Training*, Interservice/Industry Training, Simulation and Education Conference (I/ITSEC) 2010, Paper No. 10334, 2010.

[50] C. Cruz-Neira, D. Reiners, J.P. Springer, C. Neumann, C.N.S. Odom & K. Kehring, *An Integrated Immersive Simulator for the Dismounted Soldier*, Interservice/Industry Training, Simulation and Education Conference (I/ITSEC) 2011, Paper No. 11320, 2011.

[51] T. Roberts, J. Saffold & P. Garrity, *Virtual Locomotion Concepts and Metrics Study: Experimentation and Results*, Interservice/Industry Training, Simulation and Education Conference (I/ITSEC) 2012, Paper No. 12236, 2012.

[52] M. Bink, V. Ingurgio & D. James, *Training Effectiveness of Dismounted Infantry Simulation*, Interservice/Industry Training, Simulation and Education Conference (I/ITSEC) 2013, Paper No. 13166, 2013.

[53] B.M. Williamson, C. Wingrave, J.J. LaViola, T. Roberts & P. Garrity, *Natural Full Body Interaction for Navigation in Dismounted Soldier Training*, Interservice/Industry Training, Simulation and Education Conference (I/ITSEC) 2011, Paper No. 11120, 2011.

[54] B.M. Williamson, J.J. LaViola, T. Roberts & P. Garrity, *Multi-Kinect Tracking for Dismounted Soldier Training*, Interservice/Industry Training, Simulation and Education Conference (I/ITSEC) 2012, Paper No. 12378, 2012.

[55] Z. Cheng, A. Ligouri, T. Webb & H. Cheng, *Human Motion Capture in Natural Environments*, Interservice/Industry Training, Simulation and Education Conference (I/ITSEC) 2014, Paper No. 14116, 2014.

[56] T.J. Michael, Y. Minkov & R. Rockah, *Integration of Low-Cost HMD Devices in Existing Simulation Infrastructure*, Interservice/Industry Training, Simulation and Education Conference (I/ITSEC) 2014, Paper No. 14190, 2014.

[57] K. Bland, S. Lopez-Couto & J. Vassall, *Considerations on Utilizing a Game Engine as an Image Generator*, Interservice/Industry Training, Simulation and Education Conference (I/ITSEC) 2014, Paper No. 14274, 2014.

[58] H. Liu, M. Bowman, R. Adams, J. Hurliman & D. Lake, *Scaling Virtual Worlds: Simulation Requirements and Challenges*, 2010 Winter Simulation Conference, 2010.

[59] J. Thorpe, *Trends in Modeling, Simulation, & Gaming: Personal Observations About the Past Thirty Years and Speculation About the Next Ten*, Interservice/Industry Training, Simulation and Education Conference (I/ITSEC) 2010, Paper No. IF1001, 2010.

[60] M.D. Petty, R.W. Franceschini & J. Panagos, *Multi-Resolution Combat Modeling*, in *Engineering Principles of Combat Modeling and Distributed Simulation*, A. Tolk (edited by), John Wiley & Sons, 2012.

[61] F.W. Lanchester, *Aircraft in Warfare: The Dawn of the Fourth Arm*, London: Constable and Company Limited, 1916.

[62] M.D. Petty & J. Panagos, *A Unit-level Combat Resolution Algorithm Based on Entity-level Data*, Interservice/Industry Training, Simulation and Education Conference (I/ITSEC) 2008, Paper No. 8359, 2008.

[63] J.W. Graham, R.J. Andrade, *Developing Interoperable Simulations through Conceptual Modeling and Ontological Analysis*, Interservice/Industry Training, Simulation and Education Conference (I/ITSEC) 2012, Paper No. 12110, 2012.

[64] E. Cayirci, *Multinational Computer Assisted Exercise*, in *Engineering Principles of Combat Modeling and Distributed Simulation*, A. Tolk (edited by), John Wiley & Sons, 2012.

[65] K.M. Fauske, *Fysikkmotorer for militære syntetiske miljø*, FFI-notat 2009/01438, 2009.

[66] J. Mann, J. Lyons & J. de la Cruz, *Using Real-time Physics to Enhance Game Based Effects*, Interservice/Industry Training, Simulation and Education Conference (I/ITSEC) 2008, Paper No. 8033, 2008.

[67] D. Browning, J. Adkison, S. Borkman & J. De La Cruz, *PhysX Extended: An Integrated Real-Time Physics and Terrain Solution*, Interservice/Industry Training, Simulation and Education Conference (I/ITSEC) 2009, Paper No. 9391, 2009.

[68] North Atlantic Treaty Organization (NATO) Standardization Office, *STANAG 4662 – SEDRIS – Environmental Data Coding Specification (EDCS)*, NATO STANAG 4662, 2012.

[69] North Atlantic Treaty Organization (NATO) Standardization Office, *STANAG 4663 – SEDRIS – Spatial Reference Model (SRM)*, NATO STANAG 4663, 2012.

[70] North Atlantic Treaty Organization (NATO) Standardization Office, *STANAG 4664 – SEDRIS – Functional Specification and Transmittal Format*, NATO STANAG 4664, 2012.

[71] SEDRIS, http://www.sedris.org.

[72] D. Lashlee, J. Bricio, R. Holcomb & W.T. Richards, *GIS Data for Combat Modeling*, in *Engineering Principles of Combat Modeling and Distributed Simulation*, A. Tolk (edited by), John Wiley & Sons, 2012.

[73] A. Lemmers, A. Gerretsen, S. Skinner & E. Jones, *Missionland: The Creation of a Virtual Continent for Mission Simulation*, Interservice/Industry Training, Simulation and Education Conference (I/ITSEC) 2012, Paper No. 12087, 2012.

[74] United States National Imagery and Mapping Agency, *Performance Specification: Digital Terrain Elevation Data*, MIL-PRF-89020B, 2000.

[75] D.J. Lowe & M.A. Cosman, *Psuedo-Specific High-Resolution Data Boundary Techniques*, Interservice/Industry Training, Simulation and Education Conference (I/ITSEC) 2014, Paper No. 14317, 2014.

[76] P.L. Isaacson, *A Method of Generating Whole-Earth High-Resolution Terrain Texture*, Interservice/Industry Training, Simulation and Education Conference (I/ITSEC) 2007, Paper No. 7050, 2007.

[77] G. Dukstein, J. Watkins, K. Le & H. Gonzalez, *Extending Construction Simulators through Commonality & Innovative Research*, Interservice/Industry Training, Simulation and Education Conference (I/ITSEC) 2013, Paper No. 13244, 2013.

[78] K.J. Hebert & D. Sexton, *Dynamic Synthetic Environments Through Run-Time Modification of Source Data*, Interservice/Industry Training, Simulation and Education Conference (I/ITSEC) 2012, Paper No. 12050, 2012.

[79] M. Schwarz, S. Arbuthnot & D. Kemper, *Realistic Water Simulation for Training of Amphibious Vehicle Crews*, Interservice/Industry Training, Simulation and Education Conference (I/ITSEC) 2012, Paper No. 12167, 2012.

[80] Environmental Systems Research Institute (ESRI), *ESRI Shapefile Technical Description – An ESRI Whitepaper*, 1998.

[81] M. Rahmes, M. Akbari & M. McGonagle, *Optimizing Supervised Learning for Pixel Labeling and Material Classification*, Interservice/Industry Training, Simulation and Education Conference (I/ITSEC) 2014, Paper No. 14016, 2014.

[82] C. Fink, *Improving Material Classification Quality with Elevation-derived Metrics*, Interservice/Industry Training, Simulation and Education Conference (I/ITSEC) 2014, Paper No. 14380, 2014.

[83] M.D. Woodman & P. Morrison, *Implementation of Real-Time Snow Layers in Game-Based Simulation*, Interservice/Industry Training, Simulation and Education Conference (I/ITSEC) 2014, Paper No. 14361, 2014.

[84] B. Blank, G. Pasternak, A. Crane & A. Broadbent, *SmartObjects and Agents for Autonomous High Population Simulated Environments*, Interservice/Industry Training, Simulation and Education Conference (I/ITSEC) 2011, Paper No. 11098, 2011.

[85] P. Kerbusch, R. Smelik, S. Smit & F. Kuijper, *Easy Pattern-Of-Life Generation using Physical and Human Terrain*, Interservice/Industry Training, Simulation and Education Conference (I/ITSEC) 2012, Paper No. 12180, 2012.

[86] J. Watkins, T. Tamash, J. Moerk, R. Overman & J. de la Cruz, *Simulation of Civilian Traffic Rules in Computer Generated Forces*, Interservice/Industry Training, Simulation and Education Conference (I/ITSEC) 2013, Paper No. 13246, 2013.

[87] K. Bentley, J. Pivonka & M. Johnson, *Improving the Fidelity of Transportation Features in Simulation*, Interservice/Industry Training, Simulation and Education Conference (I/ITSEC) 2013, Paper No. 13059, 2013.

[88] J. Mann & L. Eifert, *Easing the Pain of Urban Modeling*, Interservice/Industry Training, Simulation and Education Conference (I/ITSEC) 2006, Paper No. 2811, 2006.

[89] T. Walker, M. Janisz, C. Porter & T. Duggan, *A Hybrid Approach for Automating Building Interiors*, Interservice/Industry Training, Simulation and Education Conference (I/ITSEC) 2008, Paper No. 8020, 2008.

[90] S.L. Griffith, *Realistic material damage simulation using real-time Finite Element Analysis*, Interservice/Industry Training, Simulation and Education Conference (I/ITSEC) 2009, Paper No. 9409, 2009.

[91] J. Lyons, P. Sowden, J. de la Cruz & H. Gonzalez, *Overcoming Static Structures: Dynamic Buildings for Training*, Interservice/Industry Training, Simulation and Education Conference (I/ITSEC) 2013, Paper No. 13292, 2013.

[92] K.D. Pfeiffer & T. Tamash, *Measuring the impact of natural environment representation on combat simulation outcomes*, Interservice/Industry Training, Simulation and Education Conference (I/ITSEC) 2014, Paper No. 14305, 2014.

[93] M. Petronzio, *11 Fascinating Facts About Google Maps*, Mashable, 2012, Retrieved October 26, 2015, from http://mashable.com/2012/08/22/google-maps-facts.

[94] B. Wiesner, D. Brockway & T. Stanzione, *Open Streaming Terrain for Modeling and Simulation*, Interservice/Industry Training, Simulation and Education Conference (I/ITSEC) 2011, Paper No. 11102, 2011.

[95] T. Palmer & R. Boyd, *Leveraging Graphics Processor Capability to Uncover Hidden Terrain Database Errors*, Interservice/Industry Training, Simulation and Education Conference (I/ITSEC) 2011, Paper No. 11070, 2011.

[96] F. Santiago, M. Verdesca, J. Watkins & J. de la Cruz, *Geospatial Correlation Testing Framework and Toolset*, Interservice/Industry Training, Simulation and Education Conference (I/ITSEC) 2012, Paper No. 12107, 2012.

[97] R.F. Richbourg, A. Ceranowicz & R.R. Lutz, *My Simulation is from Mars; Yours is from Venus*, Interservice/Industry Training, Simulation and Education Conference (I/ITSEC) 2008, Paper No. 8097, 2008.

[98] R. Siegfried, M. Bertschik, M. Hahn, G. Herrmann, J. Lüthi & M. Rother, *Effective and Efficient Training Capabilities through Next Generation Distributed Simulation Environments*, NATO Modelling and Simulation Group (NMSG) Annual Symposium 2013 (STO-MP-MSG-111), Paper No. 7, 2013.

[99] United States Department of Defense, *MIL-STD-2525C, Department of Defense Interface Standard: Common Warfighting Symbology*, 2008.

[100] J.M. Williams, E. Pollak & S.R. Schwalm, *The Application of COTS Physics Engines to CGF Vehicle Dynamics*, Interservice/Industry Training, Simulation and Education Conference (I/ITSEC) 2009, Paper No. 9324, 2009.

[101] S. Bruvoll, *Situation dependent path planning for computer generated forces*, FFI-rapport 2014/01222, 2014.

[102] Department of the Army (U.S.), *The Infantry Rifle Platoon and Squad*, Field Manual No. 3-21.8, 2007.

[103] P.E. Hart, N.J Nilsson & B. Raphael, *A Formal Basis for the Heuristic Determination of Minimum Cost Paths*, IEEE Transactions on Systems Science and Cybernetics, Volume 4, Issue 2, pages 100-107, 1968.

[104] J.J. Tran, K-T. Yao, G. Wagenbreth, D.M. Davis, R.F. Lucas & D.J. Brakeman, *A High Performance Route-Planning Technique for Dense Urban Simulations*, Interservice/Industry Training, Simulation and Education Conference (I/ITSEC) 2008, Paper No. 8293, 2008.

[105] M.R. Endsley, *Toward a Theory of Situation Awareness in Dynamic Systems*, Human Factors: The Journal of the Human Factors and Ergonomics Society, Volume 37(1), pages 32-64, 1995.

[106] P. Colon, J. Tran, K.T. Yao, M. Anhalt & J.M. Curiel, *Modeling Human Perception of Situation Awareness During Constructive Experimentation*, Interservice/Industry Training, Simulation and Education Conference (I/ITSEC) 2007, Paper No. 7448, 2007.

[107] C. von Clausewitz, *On War*, Princeton University Press, 1989.

[108] C. Lauterbach, M.C. Lin, D. Manocha, S. Borkman, E. LaFave, G. Peele & M. Bauer, *Accelerating Line-of-Sight Computations in Large OneSAF Terrains with Dynamic Events*, Interservice/Industry Training, Simulation and Education Conference (I/ITSEC) 2008, Paper No. 8273, 2008.

[109] J.R. Moulton & C. Fink, *Recent Simulation Advancements for more Credible Sensor Representations and Training*, Interservice/Industry Training, Simulation and Education Conference (I/ITSEC) 2013, Paper No. 13145, 2013.

[110] North Atlantic Treaty Organization (NATO) Standardization Agency, *AAP-06 NATO Glossary of Terms and Definitions*, Edition 2014, 2014.

[111] S.J. Anderson, *Target Classification, Recognition and Identification with HF Radar*, NATO Sensors & Electronics Technology (SET) Panel Annual Symposium 2004 (STO-MP-SET-080), Paper No. 25, 2004.

[112] R.R. Brooks, P. Ramanathan & A.M. Sayeed, *Distributed Target Classification and Tracking in Sensor Networks*, Proceedings of the IEEE, Volume 91, Issue 8, pages 1163-1171, 2003.

[113] J. Lyons, A. Turek & H. Mall, *Physics-based Damage in a Combat Medic Training Simulation*, Interservice/Industry Training, Simulation and Education Conference (I/ITSEC) 2010, Paper No. 10109, 2010.

[114] C. Eidman, G. Boyle, M. Lay & E. Pollak, *Physics Based Modeling of a Warhead Detonation using Graphics Processors*, Interservice/Industry Training, Simulation and Education Conference (I/ITSEC) 2010, Paper No. 10341, 2010.

[115] C.P. Rahaim, P.A. Salvini & L. Moore, *Explosive Fragment Engineering Visualization and Virtual Reality Demonstration Project*, Interservice/Industry Training, Simulation and Education Conference (I/ITSEC) 2011, Paper No. 11139, 2011.

[116] E. Neugebauer, D. Nitsch & O. Henne, *Architecture for a Distributed Test Bed*, NATO Modelling and Simulation Group (NMSG) Annual Symposium 2009 (RTO-MP-MSG-069), Paper No. 19, 2009.

[117] R. Siegfried, T. van den Berg, A. Cramp & W. Huiskamp, *M&S as a Service: Expectations and challenges*, 2014 Fall Simulation Interoperability Workshop (SIW), Paper No. 040, 2014.

[118] North Atlantic Treaty Organization (NATO) Science and Technology Organization (STO), *Modelling and Simulation as a Service: New Concepts and Service-Oriented Architectures*, STO Technical Report, STO-TR-MSG-131, 2015.

[119] A. Dickens, B. Holcomb, L. Wihl & R. Aplin, *Interfacing a Communications Effect Model to Provide Accurate Modeling of Communications in Computer Generated Forces*, Interservice/Industry Training, Simulation and Education Conference (I/ITSEC) 2009, Paper No. 9221, 2009.

[120] D.S. Alberts, *Information Age Transformation: Getting to a 21st Century Military*, Command and Control Research Program (CCRP) Publication Series, 2002.

[121] J.R. Potts, T. Griffith, K. Roth & J. Snyder, *Customizable Speech Centers for Automated Entities within Simulation*, Interservice/Industry Training, Simulation and Education Conference (I/ITSEC) 2012, Paper No. 12094, 2012.

[122] J. Pagan, K. Kaste, J. Reni, B. Atkinson, L. Sciarini & M. Walwanis, *Next-Generation Semi-Automated Forces (SAFs): A Guidance Model*, Interservice/Industry Training, Simulation and Education Conference (I/ITSEC) 2013, Paper No. 13081, 2013.

[123] Y. Papelis & P. Madhavan, *Modeling Human Behavior*, in *Modeling and Simulation Fundamentals: Theoretical Underpinnings and Practical Domains*, J.A. Sokolowski & C.M. Banks (edited by), John Wiley & Sons, 2010.

[124] G. Gunzelmann, C. Gaughan, T. Alexander, W. Huiskamp, K. van den Bosch, S. de Jong, A.G. Bruzzone & A. Tremori, *In Search of Interoperability Standards for Human Behaviour Representations*, Interservice/Industry Training, Simulation and Education Conference (I/ITSEC) 2014, Paper No. 14027, 2014.

[125] L.J. Moya & A. Tolk, *Towards a Taxonomy of Agents and Multi-Agent Systems*, 2007 Spring Simulation Multi-Conference (SpringSim), 2007.

[126] G.K. Bharathy, L. Yilmaz & A. Tolk, *Agent Direkted Simulation for Combat Modeling and Distributed Simulation,* in *Engineering Principles of Combat Modeling and Distributed Simulation*, A. Tolk (edited by), John Wiley & Sons, 2012.

[127] M.J. Doyle & A.M. Portrey, *Are Current Modeling Architectures Viable for Rapid Human Behavior Modeling?*, Interservice/Industry Training, Simulation and Education Conference (I/ITSEC) 2011, Paper No. 11129, 2011.

[128] D. Buede, B. DeBlois, D. Maxwell & B. McCarter, *Filling the Need for Intelligent, Adaptive Non-Player Characters*, Interservice/Industry Training, Simulation and Education Conference (I/ITSEC) 2013, Paper No. 13204, 2013.

[129] B.D. Hamilton, R. Long, N. Coeyman, V. Gottesman & J. Williams, *The Application of Commercial Gaming Technology to Adaptive Adversarial Behaviors*, Interservice/Industry Training, Simulation and Education Conference (I/ITSEC) 2008, Paper No. 8237, 2008.

[130] P. Kerbusch, W. van Doesburg, R. Jansen & P. Langeslag, *Pluggable Brains: Separating the Intelligent Behavior from the Simulated Player*, Interservice/Industry Training, Simulation and Education Conference (I/ITSEC) 2011, Paper No. 11148, 2011.

[131] K. Dill, E.R. Pursel, P. Garrity & G. Fragomeni, *Design Patterns for the Configuration of Utility-Based AI*, Interservice/Industry Training, Simulation and Education Conference (I/ITSEC) 2012, Paper No. 12146, 2012.

[132] B. Presnell, R. Houlette & D. Fu, *Making Behavior Modeling Accessible to Non-Programmers: Challenges and Solutions*, Interservice/Industry Training, Simulation and Education Conference (I/ITSEC) 2007, Paper No. 7258, 2007.

[133] A.M. Turing, *Computing Machinery and Intelligence*, Mind, New Series, Volume 59, No. 236, pages 433-460, 1950.

[134] S.K. Numrich & P.M. Picucci, *New Challenges: Human, Social, Cultural, and Behavioral Modeling*, in *Engineering Principles of Combat Modeling and Distributed Simulation*, A. Tolk (edited by), John Wiley & Sons, 2012.

[135] A.H. Dekker, *Human Behaviour Modelling as an Emerging Disruptive M&S Technology*, NATO Modelling and Simulation Group (NMSG) Annual Symposium 2013 (STO-MP-MSG-111), Paper No. 22, 2013.

[136] R.M. Jones, *Introduction to Cognitive Architectures for Modeling and Simulation* (Tutorial Slides), Interservice/Industry Training, Simulation and Education Conference (I/ITSEC) 2011, Tutorial No. 1144, 2011.

[137] R.G. Abbott, J.D. Basilico, M.R. Glickman & J. Whetzel, *Trainable Automated Forces*, Interservice/Industry Training, Simulation and Education Conference (I/ITSEC) 2010, Paper No. 10441, 2010.

[138] A.J. Gonzalez & R. Ahlers, *Context-based Representation of Intelligent Behavior in Training Simulations*, Transactions of the Society for Computer Simulation International, Volume 15, Issue 4, pages 153-166, 1998.

[139] A.J. Gonzalez, B.S. Stensrud & G. Barrett, *Formalizing Context-Based Reasoning: A Modeling Paradigm for Representing Tactical Human Behavior*, International Journal of Intelligent Systems, Volume 23, Issue 7, pages 822-847, 2008.

[140] A.J. Gonzalez & P. Brézillon, *Comparing Two Context-driven Approaches for Representation of Human tactical Behavior*, The Knowledge Engineering Review, Volume 23, Issue 3, pages 295-315, 2008.

[141] R.A. Løvlid, A. Alstad, G. Skogsrud, S. Bruvoll, O.M. Mevassvik & K. Bråthen, *Modelling battle command with context-based Reasoning*, FFI-rapport 2013/00861, 2013.

[142] A. Marzinotto, M. Colledanchise, C. Smith & P. Ögren, *Towards a Unified Behavior Trees Framework for Robot Control*, 2014 IEEE International Conference on Robotics and Automation (ICRA), 2014.

[143] J.E. Laird, *The Soar Cognitive Architecture*, The Massachusetts Institute of Technology Press, 2012.

[144] J.R. Anderson, *Rules of the Mind*, Psychology Press, 1993.

[145] E.L. DuBois, W.P. Hughes & L.J. Low, *A Concise Theory of Combat*, Naval Postgraduate School, 1997.

[146] Institute of Electrical and Electronics Engineers (IEEE), *IEEE Recommended Practice for Distributed Simulation Engineering and Execution Process (DSEEP)*, IEEE Standard 1730-2010, 2011.

[147] IHS Jane's, http://www.janes.com.

[148] GlobalSecurity.org, http://www.globalsecurity.org.

[149] Department of the Army (U.S.), *Worldwide Equipment Guide – Volume 1: Ground Systems*, 2011.

[150] Department of the Army (U.S.), *Worldwide Equipment Guide – Volume 2: Airspace and Air Defence Systems*, 2011.

[151] Department of the Army (U.S.), *Worldwide Equipment Guide – Volume 3: Naval and Littoral Systems*, 2011.

[152] D. Shine, O. Mazonka, L. Holden, D. Hemming & T. Frank, *A Methodology for the Generation, Storage, Verification and Validation of Performance Data for Modelling and Simulation*, NATO Modelling and Simulation Group (NMSG) Annual Symposium 2013 (STO-MP-MSG-111), Paper No. 15, 2013.

[153] North Atlantic Treaty Organization (NATO), *NATO Code of Best Practice for C2 Assessment*, Command and Control Research Program (CCRP), 2002.

[154] D.B. Maxwell, J. Geil, W.A. Rivera & H. Liu, *A Distributed Scene Graph Approach to Scaled Simulation-Based Training Applications*, Interservice/Industry Training, Simulation and Education Conference (I/ITSEC) 2014, Paper No. 14033, 2014.

[155] K-T. Yao, R.F. Lucas, C.E. Ward, G. Wagenbreth & T.D. Gottschalk, *Data Analysis for Massively Distributed Simulations*, Interservice/Industry Training, Simulation and Education Conference (I/ITSEC) 2009, Paper No. 9350, 2009.

[156] O. Balci, *Verification, Validation and Testing*, in *Handbook of Simulation: Principles, Methodology, Advances, Applications, and Practice*, J. Banks (edited by), John Wiley & Sons, 1998.

[157] P.A. Roman & J.Surdu, *Simulation Support to Operations – Bringing Simulation to the Fight*, Interservice/Industry Training, Simulation and Education Conference (I/ITSEC) 2009, Paper No. 9021, 2009.

[158] P. Wikberg, M. Thorstensson, P. Hammar & G. Tolt, *Mission Integrated Simulation – A Case Study*, Interservice/Industry Training, Simulation and Education Conference (I/ITSEC) 2014, Paper No. 14085, 2014.

[159] J.I. Hyndoy, O.M. Mevassvik & K. Brathen, *Simulation in Support of Course of Action Development in Operations*, Interservice/Industry Training, Simulation and Education Conference (I/ITSEC) 2014, Paper No. 14119, 2014.

[160] D. O'Connor, *Dynamic Decision Support – A War Winning Edge*, NATO Modelling and Simulation Group (NMSG) Annual Symposium 2013 (STO-MP-MSG-111), Paper No. 13, 2013.

[161] M. Hieb & D.T. Maxwell, *Converging Simulation and C2: Improving Foundation Data Consistency and Affordability*, Interservice/Industry Training, Simulation and Education Conference (I/ITSEC) 2012, Paper No. 12326, 2012.

[162] CAE GESI, http://www.cae.com/defence-and-security.

[163] CAE, *CAE GESI Command & Staff Training* (Brochure), Retrieved October 26, 2015, from http://www.cae.com/uploadedfiles/content/businessunit/defence_and_security/media_centre/document/brochure.gesi.command.staff.training.pdf.

[164] Department of Defence (U.S.) Joint Staff J7, *JTLS Executive Overview*, Joint Theater Level Simulation (JTLS 4.1.6.0), 2014.

[165] ROLANDS & ASSOCIATES Corporation, http://www.rolands.com.

[166] MASA Group, http://masa-group.biz.

[167] MASA Group, *MASA SWORD – Automated, Aggregated Constructive Simulation for Efficient Training & Analysis* (Brochure).

[168] U.S. Army PEO STRI – OneSAF, http://www.peostri.army.mil/PRODUCTS/ONESAF.

[169] J.J. Padilla, *Military Simulation Systems*, in *Engineering Principles of Combat Modeling and Distributed Simulation*, A. Tolk (edited by), John Wiley & Sons, 2012.

[170] eSim Games, https://www.esimgames.com.

[171] N. Hinrichsen, *CV90/35 Infantry Fighting Vehicle Simulator for the Danish Army*, eSim Games LLC, 2011.

[172] RE-liON, http://www.re-lion.com.

[173] TerraSim Inc., http://www.terrasim.com.

[174] Bohemia Interactive Simulations (BISim), https://bisimulations.com.

[175] Bohemia Interactive Simulations (BISim) – Virtual Battlespace 3 (VBS3), https://bisimulations.com/virtual-battlespace-3.

[176] Bohemia Interactive Simulations (BISim) – VBS IG, https://bisimulations.com/vbs-ig.

[177] Bohemia Interactive Simulations (BISim) – VBS Tactics, https://bisimulations.com/vbs-tactics.

[178] P-I. Evensen, *Design and Implementation of an Augmented Reality System for VBS2*, FFI-rapport 2012/00490, 2012.

[179] Bohemia Interactive Simulations (BISim), *Dispatches from BISim at ITEC: Day 2*, 2015, Retrieved October 26, 2015, from https://bisimulations.com/content/wed-04292015-1059/dispatches-bisim-itec-day-2.

[180] VT MÄK, http://www.mak.com/products.html.

[181] VT MÄK – VR-Forces: Comuter Generated Forces and Simulator Development, http://www.mak.com/products/simulate/vr-forces.html.

[182] S. Martinussen, I. Tansem, P-I. Evensen, H. Rødal & I. Bore, *Simulation In Support of Defence Structure Analysis*, RTO-MP-MSG-060, Paper No. 22, 2008.

[183] United States Department of Defense, *MIL-STD-2525C, Department of Defense Interface Standard: Common Warfighting Symbology*, 2008.

[184] J.F. Geiner, I. Johansen & E. Daltveit, *Alternative Strukturer for Landmakten*, FFI-rapport 2011/00863, 2011.

[185] R.R. Leonhard, *Fighting by Minutes: Time and the Art of War*, Westport: Praeger, 1994.

[186] J. Kelly & M. Brennan, *Distributed Manoeuvre – 21st Century Offensive Tactics*, Land Warfare Studies Centre, Working Paper No. 134, 2009.

[187] L. Granowetter, *The WebLVC Protocol: Design and Rationale*, Interservice/Industry Training, Simulation and Education Conference (I/ITSEC) 2013, Paper No. 13163, 2013.

[188] J-R. Martinez-Salio & J-M. Lopez-Rodriguez, *Future of LVC Simulation: Evolving Towards the MSaaS Concept*, Interservice/Industry Training, Simulation and Education Conference (I/ITSEC) 2014, Paper No. 14072, 2014.

[189] P. Mell & T. Grance, *The NIST Definition of Cloud Computing*, National Institute of Standards and Technology (NIST) Special Publication 800-145, 2011.

[190] M.R. Macedonia, C. Bouwens & J. Shiflett, *Cloud Simulation Infrastructure – Delivering Simulation From the Cloud*, Interservice/Industry Training, Simulation and Education Conference (I/ITSEC) 2012, Paper No. 12343, 2012.

[191] R. Siegfried, A. Diehl, S.Y. Diallo, M. Bertschik, G. Herrmann & M. Rother, *Outline of a Service-Based Reference Architecture for Effective and Efficient Use of Modelling and Simulation*, NATO Modelling and Simulation Group (NMSG) Annual Symposium 2014 (STO-MP-MSG-126), Paper No. 18, 2014.

[192] K. Murray, R. Darken, B. Vogt & S.R. Goerger, *Early Synthetic Prototyping: Exploring New Designs and Concepts Within Games*, Interservice/Industry Training, Simulation and Education Conference (I/ITSEC) 2014, Paper No. 14133, 2014.

[193] C. Higgins, *No Man's Sky would take 5 billion years to explore*, Wired UK, 2014, Retrieved October 26, 2015, from http://www.wired.co.uk/news/archive/2014-08/18/no-mans-sky-planets.

[194] S.G. Skinner & C.J. Hutchinson, *Real Time Ray – Tracing for Interactive Visual Simulation*, Interservice/Industry Training, Simulation and Education Conference (I/ITSEC) 2010, Paper No. 10069, 2010.

Abbreviations

AAR	After-Action Review
AI	Artificial Intelligence
AMCS	Army Mission Command Systems
ANN	Artificial Neural Network
APC	Armoured Personnel Carrier
API	Application Programming Interface
APS	Active Protection Systems
AR	Augmented Reality
ASCII	American Standard Code for Information Interchange
ASR	Automatic Speech Recognition
AVCATT	Aviation Combined Arms Tactical Trainer
BFT	Blue-Force Tracking
BISim	Bohemia Interactive Simulations
BMS	Battlefield Management System
BT	Behaviour Tree
C2	Command and Control
C2SIM	C2-to-Simulation
C4I	Command, Control, Communications, Computers, and Intelligence
CADRG	Compressed ARC Digitized Raster Graphics
CAS	Close Air Support
CAX	Computer-Assisted Exercise
C-BML	Coalition Battle Management Language
CBRN	Chemical, Biological, Radiological, and Nuclear
CCRP	Command and Control Research Program
CD&E	Concept Development & Experimentation
CES	Communication Effects Service
CFD	Computational Fluid Dynamics
CGF	Computer-Generated Forces
COA	Course of Action
COIN	Counter-Insurgency
COTS	Commercial Off-The-Shelf
CT	Counter-Terrorism
CxBR	Context-Based Reasoning
DCRI	Detection, Classification, Recognition, and Identification
DDS	Data Distribution Service
DEM	Digital Elevation Model
DIS	Distributed Interactive Simulation
DoD	Department of Defence
DSEEP	Distributed Simulation Engineering and Execution Process
DTED	Digital Terrain Elevation Data
DVTE	Deployable Virtual Training Environment
EBS	Exterior Ballistics Service

EDCS	Environmental Data Coding Specification
EO	Electro-Optical
ESRI	Environmental Systems Research Institute
EW	Electronic Warfare
FEA	Finite Element Analysis
FOM	Federation Object Model
FPS	First-Person Shooter
FSM	Finite-State Machine
GeoTIFF	Geographic Tagged Image File Format
GESI	GEfechts-SImulation System
GM-VV	Generic Methodology for Verification and Validation
GPS	Global Positioning System
GPU	Graphics Processing Unit
GUI	Graphical User Interface
HE	High Explosive
HITL	Human-In-The-Loop
HLA	High Level Architecture
HMD	Head-Mounted Display
HMM	Hidden Markov Model
HTML	HyperText Markup Language
I/ITSEC	Interservice/Industry Training, Simulation and Education Conference
IEEE	Institute of Electrical and Electronics Engineers
IFV	Infantry Fighting Vehicle
IG	Image Generator
IR	Infrared
ISTAR	Intelligence, Surveillance, Target Acquisition, and Reconnaissance
ITEC	International Training and Education Conference
IW	Irregular Warfare
JC3IEDM	Joint Command, Control, Consultation, and Information Exchange Data Model
JCATS	Joint Conflict and Tactical Simulation
JSON	JavaScript Object Notation
JTLS	Joint Theater Level Simulation
KE	Kinetic Energy
KML	Keyhole Markup Language
LOS	Line of Sight
LRF	Laser Range Finder
LVC	Live, Virtual, and Constructive
M&S	Modelling and Simulation
MAS	Multi-Agent System
MBT	Main Battle Tank
MIP	Multi-Lateral Interoperability Program
MOUT	Military Operations on Urban Terrain
MRM	Multi-Resolution Modelling

MSaaS	Modelling and Simulation as a Service
MSCO	Modeling and Simulation Coordination Office
MSDL	Military Scenario Definition Language
MTBF	Mean Time Between Failures
NATO	North Atlantic Treaty Organization
NCW	Network-Centric Warfare
NIST	National Institute of Standards and Technology
NLP	Natural Language Processing
NMSG	NATO Modelling and Simulation Group
NPC	Non-Player Character
NTF	NATO Training Federation
NWDC	Naval Warfare Development Command
ODT	Omnidirectional Treadmill
OMG	Object Management Group
OMT	Object Model Template
OneSAF	One Semi-Automated Forces
OOB	Order of Battle
OOTW	Operations Other Than War
PDU	Protocol Data Unit
PEO STRI	Program Executive Office for Simulation, Training and Instrumentation
PKO	Peacekeeping Operations
QoS	Quality of Service
R&D	Research and Development
RBS	Rule-Based System
ROE	Rules of Engagement
RPR FOM	Real-time Platform Reference Federation Object Model
RTI	Run-Time Infrastructure
SA	Situational Awareness
SAF	Semi-Automated Forces
SAR	Search and Rescue
SDK	Software Development Kit
SDS	Synthetic Dynamic Environment Service
SET	Sensors & Electronics Technology
SISO	Simulation Interoperability Standards Organization
SIW	Simulation Interoperability Workshop
SME	Subject-Matter Expert
SNE	Synthetic Natural Environment
SO	Stability Operations
SOF	Special Operations Force
SRM	Spatial Reference Model
STANAG	Standardization Agreement
STO	Science and Technology Organization
STRICOM	Simulation, Training and Instrumentation Command

TAF	Trainable Automated Forces
TIN	Triangular Irregular Network
TTP	Tactics, Techniques, and Procedures
U.S.	United States
UAS	Unmanned Aerial Sytem
UAV	Unmanned Aerial Vehicle
UGV	Unmanned Ground Vehicle
USJFCOM	United States Joint Federated Forces Command
V&V	Verification and Validation
VBS	Virtual Battlespace
VR	Virtual Reality
webGL	Web Graphics Library
WES	Weapons Effects Service
WSC	Winter Simulation Conference
XML	Extensible Markup Language