FFI-RAPPORT

19/01622

webSAF

– an easy-to-use, web-based graphical user interface for controlling semi-automated forces

Per-Idar Evensen
Kristian Selvaag
Madeleine Josephine Selvig Hallén
Dan Helge Bentsen
Håvard Stien
Helene Rødal Holhjem

webSAF

– an easy-to-use, web-based graphical user interface for controlling semi-automated forces

Per-Idar Evensen
Kristian Selvaag
Madeleine Josephine Selvig Hallén
Dan Helge Bentsen
Håvard Stien
Helene Rødal Holhjem

Norwegian Defence Research Establishment (FFI) 16 September 2019

Keywords
Modellering og simulering
Datagenererte styrker
Grafisk brukergrensesnitt
Simuleringsverktøy
Stridssimuleringer
Landoperasjoner

FFI-rapport
19/01622

Prosjektnummer
1353

Elektronisk ISBN
978-82-464-3218-2

Approvers
Dan Helge Bentsen, *Research Manager*
Halvor Ajer, *Director of Research*

The document is electronically approved and therefore has no handwritten signature.

Summary

At the Norwegian Defence Research Establishment (FFI) we address complex issues related to warfare. One of the research questions we are investigating is how to increase combat effectiveness in land force operations. As part of this work, we conduct detailed, entity-level simulations of battalion to brigade level operations in order to assess and compare the performance of different land force structures, which may vary with regard to *composition of material and equipment*, *tactical organization*, or *operational concept*. Our simulation experiments are conducted as what may be described as *simulation-supported*, *two-sided* (Blue and Red) *wargames*, where military officers participate as players/operators on both sides.

For our use, traditional constructive simulation tools often lack the required level of resolution, are too complex and cumbersome to use, or are not flexible enough with respect to representation of new technologies such as new sensor systems, weapon systems, or protection systems. We are therefore developing *webSAF* – an easy-to-use, web-based graphical user interface (GUI) for controlling semi-automated entities in constructive simulations. This enables us to tailor the GUI and the simulation environment to our specific needs. The system has been named *webSAF* to reflect that it is a *web-based* system for controlling *semi-automated forces* (SAF).

webSAF is tailored for simulation-supported, two-sided wargaming and requires only a minimum number of operators on each side. It currently has functionality for controlling indirect fire and manoeuvre entities simulated in Virtual Battlespace (VBS) and air defence entities simulated in VR-Forces. However, webSAF is in principle independent of the simulation tools in use and can be used to control entities in a federation of different simulation tools. In the future we plan to extend webSAF with functionality for controlling combat support entities simulated in VBS, and air entities simulated in VR-Forces.

This report describes the functionality of webSAF version 1.0, which was completed in June 2019. First, we describe the background for this work. Next, we discuss our general requirements for simulation of land force operations for experimentation and analysis. Then we give an overall description of the simulation environment for webSAF, including the simulation tools in use. After this, we give a more detailed description of webSAF, including the functionality, design and implementation. Finally, we summarize our initial experiences with webSAF and outline our plans for further work.

Sammendrag

Ved Forsvarets forskningsinstitutt (FFI) ser vi på komplekse problemstillinger relatert til krigføring. Et av spørsmålene vi undersøker, er hvordan stridseffektiviteten i landoperasjoner kan økes. Som en del av dette arbeidet gjør vi detaljerte entitetsbaserte simuleringer av operasjoner på bataljon- til brigadenivå, for å vurdere og sammenlikne forskjellige landmaktstrukturer som kan variere med hensyn på sammensetning av materiell og utstyr, taktisk organisering eller operasjonskonsept. Våre simuleringseksperimenter gjennomføres som simuleringsstøttede, tosidige (Blå og Rød) krigsspill, hvor militære offiserer deltar som spillere/operatører.

Tradisjonelle simuleringsverktøy for “constructive”-simuleringer har ofte ikke tilstrekkelig oppløsing, er for komplekse og tungvinte å bruke, eller er ikke fleksible nok med hensyn til å representere nye teknologier som for eksempel nye sensorsystemer, våpensystemer eller beskyttelsessystemer. Vi utvikler derfor webSAF – et brukervennlig, web-basert grafisk brukergrensesnitt for å styre semiautomatiske entiteter i “constructive”-simuleringer. Dette gjør oss i stand til å skreddersy brukergrensesnittet og simuleringsmiljøet til våre spesifikke behov. Systemet har fått navnet webSAF for å reflektere at det er et web-basert system for å styre semiautomatiske styrker (semi-automated forces – SAF).

webSAF er skreddersydd for simuleringsstøttede, tosidige krigsspill og krever bare et minimum antall spillere/operatører på hver side. Det har funksjonalitet for å styre indirekte ild- og manøverentiteter simulert i Virtual Battlespace (VBS) og luftvernentiteter simulert i VR-Forces. webSAF er i prinsippet uavhengig av simuleringsverktøyene som anvendes og kan brukes til å styre entiteter i en føderasjon av forskjellige simuleringsverktøy. I framtiden planlegger vi å utvide webSAF med funksjonalitet for å styre kampstøtteelementer i VBS og luftentiteter i VR-Forces.

Denne rapporten beskriver funksjonaliteten i webSAF versjon 1.0, som var ferdig i juni 2019. Først beskriver vi bakgrunnen for dette arbeidet. Deretter diskuterer vi de generelle kravene vi har til simulering av landoperasjoner for eksperimentering og analyse. Videre gir vi en overordnet beskrivelse av simuleringsmiljøet for webSAF, inkludert simuleringsverktøyene som brukes. Etter dette gir vi en mer omfattende beskrivelse av webSAF, inkludert funksjonaliteten, designet og implementasjonen. Til slutt oppsummerer vi de foreløpige erfaringene med webSAF og skisserer planene våre for videre arbeid.

Contents

1 Introduction

At the Norwegian Defence Research Establishment (FFI) we address complex issues related to warfare. One of the research questions we are investigating is how to increase combat effectiveness in land force operations. As part of this work, we conduct detailed, entity-level simulations of battalion to brigade level operations in order to assess and compare the performance of different land force structures and operational concepts. Our simulation experiments are conducted as what may be described as s*imulation-supported, two-sided* (Blue and Red) *wargames*, where military officers participate as players/operators on both sides [1][2][3].

For our use, traditional constructive simulation tools often lack the required level of resolution, are too complex and cumbersome to use, or are not flexible enough with respect to representation of new technologies such as new sensor systems, weapon systems, or protection systems. We are therefore developing *webSAF* – an easy-to-use, web-based graphical user interface (GUI) for controlling semi-automated entities in constructive simulations [2][3]. This enables us to tailor the GUI and the simulation environment to our specific needs. The system has been named *webSAF* to reflect that it is a *web-based* system for controlling *semi-automated forces* (SAF).

webSAF is tailored for simulation-supported, two-sided wargaming and requires only a minimum number of players/operators on each side. It currently has functionality for controlling indirect fire and manoeuvre entities simulated in Virtual Battlespace (VBS) from Bohemia Interactive Simulations (BISim) and air defence entities simulated in VR-Forces from VT MAK. However, webSAF is in principle independent of the simulation tools in use and can be used to control entities in a federation of different simulation tools. In the future we plan to extend webSAF with functionality for controlling combat service support entities simulated in VBS, and air entities simulated in VR-Forces.

This report is organized as follows: First, in Chapter 2, we describe the background for this work. Next, in Chapter 3, we discuss our general requirements for simulation of land force operations for experimentation and analysis. In Chapter 4 we give an overall description of the simulation environment for webSAF, including the simulation tools we use. Chapter 5 provides a more detailed description of webSAF, including the functionality, design and implementation. In Chapter 6 we summarize our initial experiences with webSAF. Finally, in Chapter 7, we outline our plans for further work.

This report describes the functionality in webSAF version 1.0, which was completed in June 2019. The work with developing webSAF version 1.0 has mainly been carried out in FFI-project 1353 “Combat Effectiveness in Land Operations II”, FFI-project 1401 “Combat systems – manoeuvre II”, and FFI-project 1434 “Tactical and operational fires in the land domain”.

2 Background

The first time an interactive, constructive simulation system with semi-automated forces (SAF) was used to support analysis of land force structures at FFI was in 2010. In FFI-project 1143 "Future Land Forces" the performance of five fundamentally different land force structures was evaluated through a series of simulation experiments [4][5]. The goal was to rank these structures based on their relative performance. In addition, the experiments revealed several strengths and weaknesses of the evaluated structures.

The lightweight simulation tool mōsbē from BreakAway was used in the experiments. The main reasons for this choice were that mōsbē supports simulation of brigade-level operations and has a user interface that makes it easy to control large groups of entities. The experiments were conducted as simulation-supported, two-sided wargames, where military officers planned and controlled the operations, and the simulation tool kept track of the movement of units and calculated the results of duels and indirect fire attacks. Figure 2.1 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 was discontinued in 2008, but the tool was used at FFI until 2014.

Since then, GESI (GEfechts-SImulation System) from CAE (the command and staff training system at the Norwegian Army Land Warfare Centre) has been used a few times in a similar manner. Most recently, simulations in GESI was used to support the special review of the Norwegian land forces, which took place in 2017.

Both mōsbē and GESI have several significant weaknesses that can produce questionable simulation results. For example, they do not support representation of micro-terrain features, and this systematically favours long-range, direct fire weapon systems. In addition, the human behaviour models are very simple, and this entails that the operators have to spend a lot of time micromanaging the units. Furthermore, mōsbē and GESI do not have an application programming interface (API) for developing additional functionality or plug-ins. When used for experimentation and analysis purposes, it is also a disadvantage that GESI requires a large number of operators (since it is a system for training command and staff procedures). This limits the convenience and accessibility of GESI simulations for experimentation purposes.

Consequently, we needed to establish a new capability for conducting more detailed constructive simulations of battalion- to brigade-level operations at FFI. Based on our experiences with simulations in mōsbē and GESI, we identified two main factors that have the potential to improve the fidelity of our constructive simulations: (1) *increased terrain resolution*, and (2) *better tactical artificial intelligence* (AI) that can exploit this terrain [1]. For example, we expect that these two factors will result in more realistic detection and engagement distances.

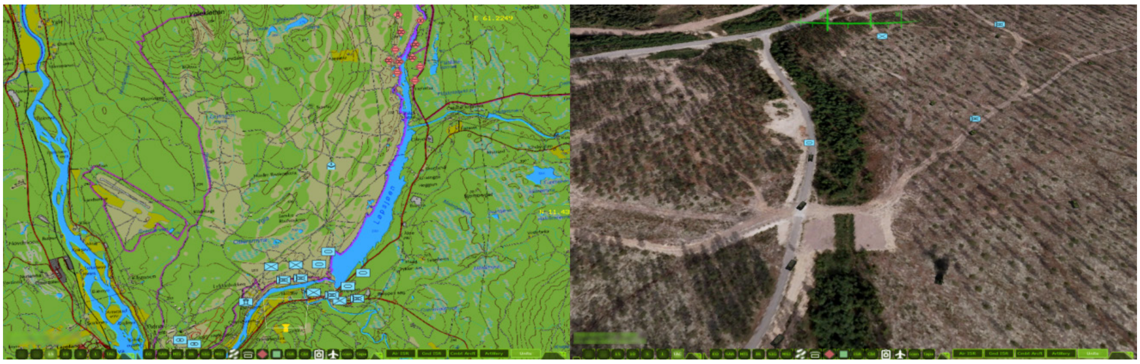


Figure 2.1 Two-dimensional theater view (to the left), and three-dimensional tactical view (to the right) in mōsbē ([4][5]).

At FFI we have used VBS as a tool for experimentation and analysis since 2008. For instance, VBS has been used in several detailed, smaller-sized (platoon to company) virtual simulation experiments, where the goal has been to evaluate the operational benefit of new technologies and new concepts by testing them out in a simulated environment [6][7]. Because we use VBS for virtual simulations, it is beneficial and cost-effective to use VBS for constructive simulations as well.

It is now possible to conduct simulations with more than 1,500 constructive, semi-automated entities in VBS. In addition, BISim is developing a new framework for more realistic human behaviour models in VBS, called VBS Control. Based on these improvements, it should be possible to conduct detailed simulations with multiple manoeuvre battalions in VBS, but VBS currently lacks an appropriate user interface for controlling constructive entities.

Over the last few years we have seen an increasing use of web technologies for modelling and simulation (M&S). The new version of the HyperText Markup Language (HTML), HTML5 (which was finalized in October 2014), especially provides new abilities for creating interactive web-interfaces for simulations and games [1]. The biggest advantage of using web technologies for M&S is increased *accessibility*.

The next generation distributed simulation environments are envisaged to rely heavily on open standards and service-based architectures [8]. M&S as a service (MSaaS) is an architectural and organizational approach that promotes abstraction, loose coupling, reusability, composability and discovery of M&S services [9][10]. In this paradigm, the vision is that instead of composing a simulation environment as a federation of different individual simulation tools, the users will be able to compose a simulation environment based on services. Web-based GUI systems that are independent from the simulation tools will of course fit perfectly into the MSaaS paradigm.

3 Simulation of land force operations for experimentation and analysis

We conduct simulations of land force operations for experimentation and analysis purposes, and as mentioned, one of the main research questions we are investigating is *how to increase combat effectiveness*. As part of this work we need to assess and compare the performance of different land force structures, which may vary with regard to *composition of material and equipment*, *tactical organization*, or *operational concept*.

Our focus is on simulation of the actual combat phases with engagements and skirmishes. These phases of combat are also the most complex and therefore the most challenging phases to simulate.

We have so far not found a single simulation tool that is satisfactory for our use, and it is of course not feasible for us to develop our own simulation tool from scratch. Our overall solution has therefore been to develop our own web-based GUI system (or front-end) and tailor available commercial off-the-shelf (COTS) simulation tools to suit our needs.

3.1 Simulation-supported wargames

As mentioned, our simulation experiments are conducted as what can be described as *simulation-supported*, *two-sided* (Blue and Red) *wargames*, where military officers participate as players/operators on both sides. A typical wargaming experiment consists of three major phases [11]:

1. *Preparation phase*
2. *Execution phase*, including a joint operational planning process
3. *Analysis phase*

The relationship between the different phases is illustrated in Figure 3.1, where the planning process and the game execution phase is the core of the experiments.

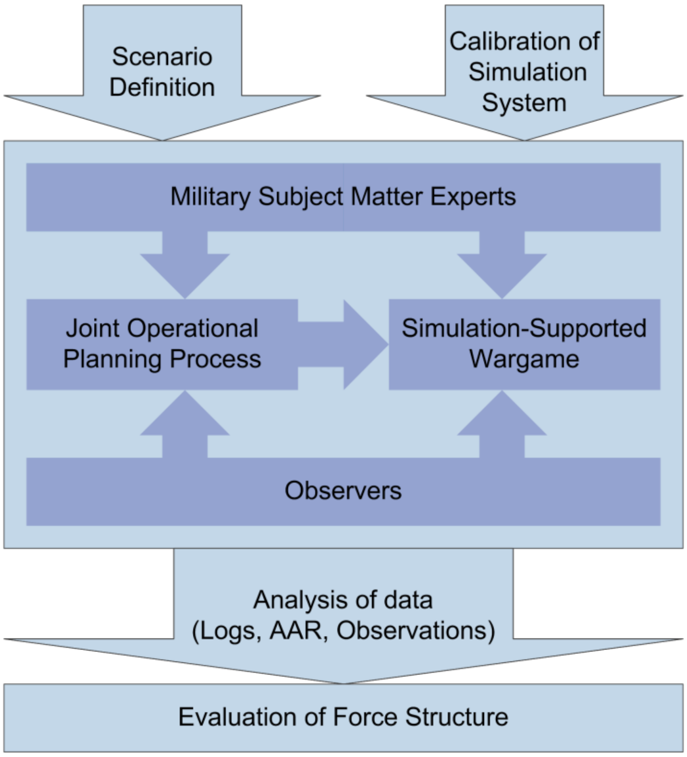


Figure 3.1 The different phases in a simulation-supported wargaming experiment.

3.2 Entity-level versus aggregate-level models

With today's computing capabilities it is possible to simulate operations up to brigade-level size using entity-level models. 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 makes them more accessible for face validation [1]. Nevertheless, it is also a known issue that current entity-level models tend to produce attrition levels that are higher than those observed historically [12][13]. “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” [12]. In other words, current constructive entity-level combat models lack good representations of the human aspects of combat and combat friction, resulting in that the simulated operations tend to run smoother than they would in the real world.

For our use, however, it would have been difficult to calibrate aggregate-level models to represent new combat systems or new concepts due to the lack of data from real operations for such systems or concepts. Furthermore, our need for higher resolution, and also the possibility of combining virtual and constructive simulations (e.g. having virtual air entities operating together with constructive ground entities), implies that we need a simulation environment based on entity-level models.

3.3 Modelling human behaviour

Modelling realistic human behaviour and cognition, including decision-making and creativity, is the hardest and most complex challenge in combat simulation [14]. Human behaviour modelling is challenging because "[h]uman behaviour is not generally yet thought to obey observable laws" [15]. Consequently, the current status for human behaviour simulation is that it can be used "to understand, [but] not necessarily predict, the aggregate behaviour of an inherently complex system for which we have no better model" [16]. When using human behaviour models "it is often possible to perform sensitivity analysis and identify broad trends as opposed to exact predictions" [16]. For example, a constructive simulation 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 a battle with a certain probability [16].

3.4 Summary of simulation capability requirements

The most important requirements for our new simulation capability for experimentation and analysis can be summarized as follows:

- It must support entity-level simulations of battalion- to brigade-level land force operations, and use simulation models with high resolution. Typically, we need to simulate operations that include between one and four manoeuvre battalions (each with 50–60 combat vehicles and about 200 soldiers), one or two artillery battalions, and one air defence battalion on each side.

- It must represent entities from the following capabilities: manoeuvre (MBTs, infantry fighting vehicles (IFVs), unmanned ground vehicles (UGVs), infantry, etc.), indirect fire (artillery, missile launchers, close air support, etc.), air defence (missile launchers, radars, etc.), aviation (fixed wing aircrafts, helicopters, unmanned aerial vehicles (UAVs), etc.), combat engineering, and intelligence, surveillance, target acquisition, and reconnaissance (ISTAR) (sensors and facilities).

- It must support high terrain resolution (10 meters between the elevation points, or better) and representation of micro-terrain features.

- It must have a GUI that is easy to use (military officers should be able to control the entities with minimal instruction).

- It must have an API for developing additional functionality (e.g. customized simulation models).

- It should have a tactical AI where the operators are able to issue high-level orders at the company level and more detailed orders at the platoon level for vehicles and squad level for infantry.
- It should have a tactical AI where the entities are able to intelligently take advantage of the terrain.

4 Simulation environment

Based on the requirements summarized in Chapter 3.4, we have composed a simulation environment where the ground-to-ground combat entities are simulated in VBS, and the air and air defence entities are simulated in VR-Forces. The entities simulated in VBS use behaviour models developed in VBS Control. All the constructive, semi-automated entities are controlled from web clients (webSAF Clients). Figure 4.1 shows an overview of the components in a webSAF simulation. The simulation tools in use are briefly described in Chapter 4.1, and webSAF is described in more detail in Chapter 5.

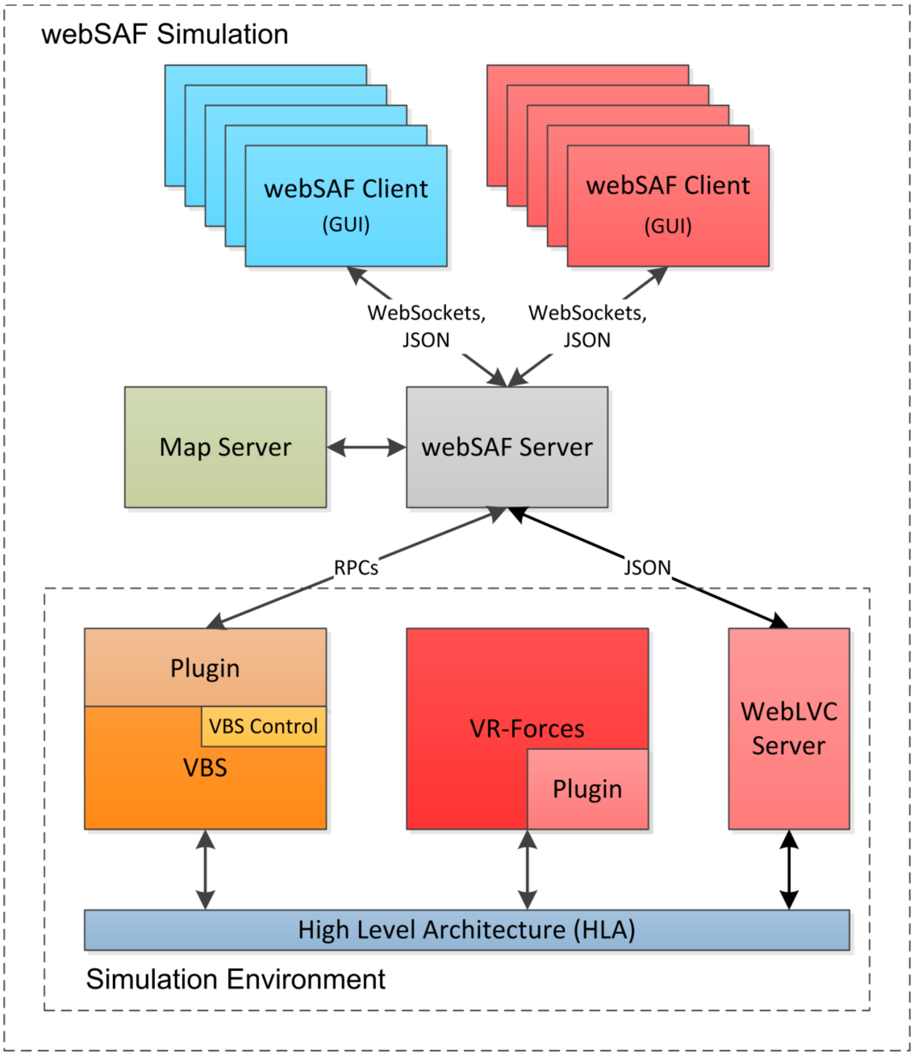


Figure 4.1 Overview of the components in a webSAF simulation.

4.1 Simulation tools

4.1.1 Virtual Battlespace (VBS)

VBS is an interactive, three-dimensional synthetic environment, for use in military training and experimentation. It is developed by Bohemia Interactive Simulations (BISim) and is based on game technology from the Armed Assault (ARMA) series. VBS is delivered with a comprehensive content library funded by different nations over the years, and in addition it has its own scripting language for creating new functionality. VBS is used by many military organizations worldwide (including the Norwegian Armed Forces), and has become an industry standard in game-based military simulation.

VBS Simulation Software Development Kit (SDK) is a new developer toolkit for VBS. It includes a library of APIs and source code allowing developers to customize virtually every aspect of VBS and produce custom applications.

4.1.2 VBS Control

VBS Control is a new framework for AI in VBS. The behaviour models used by VBS Control are based on behaviour trees (BTs), and users can create customized behaviour models through the VBS Control Editor. The BTs can be visually debugged in real time, and VBS can visualize the path planning and navigation mesh for debugging purposes.

BTs are a relatively new and increasingly popular approach for developing behaviour models [17][18]. The approach has become especially popular for creating behaviours for non-player characters (NPCs) in computer games, robots, and autonomous vehicles. The first high-profile computer game that used BTs was Halo 2 from Bungie Software [19], which was released in 2004.

BTs are represented as directed trees with a hierarchy of control flow nodes and task nodes that control the behaviour of an agent. The control flow nodes contain some decision logic and have at least one child node. The task nodes are leaf nodes (nodes without children) and contain conditional tasks which test some property in the simulated environment (or the real world in the case of robots and autonomous vehicles) or action tasks which alter the state of the simulation (or the real world) in some way [1].

What makes BTs so powerful is their composability and modularity. Task nodes and control flow nodes are composed into sub-trees which represent more complex actions, and these actions can be composed into higher-level behaviours [20]. Task nodes and action sub-trees can be reused, and different sub-trees can be developed independently of each other. BT editors enable users (e.g. military simulation users) to create modular behaviour models without needing programming skills. For our simulation environment we are developing BTs for combat drills corresponding to each manoeuvre order that can be issued from webSAF [17][18].

4.1.3 VR-Forces

VR-Forces is a framework for computer-generated forces (CGF) developed by VT MAK. It includes simulation models for hundreds of battlefield units and systems in all domains (land, naval, air, and space), and can be used in both aggregate- and entity-level mode. The VR-Forces framework is customizable, and provides several C++-based APIs for different development tasks.

5 webSAF

webSAF consists of a *server*, a number of *web clients*, and a *map server*. The web clients connect to the server through WebSockets, and send and receive data in the form of JavaScript Object Notation (JSON) packets. Map tiles used by the GUI system are streamed from a map server. The ground-to-ground combat entities are simulated in VBS, and the webSAF server communicates with VBS through remote procedure calls (RPCs) using gRPC [21]. The air defence sensors and weapons are simulated in VR-Forces, and the webSAF server communicates with VR-Forces using JSON packets. The communication with VR-Forces goes through a WebLVC Server [22], which wraps the JSON packets inside High Level Architecture (HLA) interactions. The HLA interactions are unwrapped by plug-ins in VR-Forces. The typical data transferred between the webSAF server and the simulation tools are entity status data and orders.

5.1 Components

5.1.1 webSAF server

The webSAF server is the management hub of the system. It is the connection link between the web clients and the simulation environment. All user interaction between a user and a simulation tool goes through the webSAF server. The webSAF server makes sure all clients get the information they should have, and no more. It also ensures that only valid commands are sent from the clients to the underlying simulation environment.

The webSAF server also hosts the scenarios used in a simulation. It uses the scenario files to initializes the simulation environment and receives updates from the different simulation tools used in the simulation environment. The webSAF server composes this information and maintains an overview of the current state of the simulation.

5.1.2 webSAF client

Operators use the webSAF client to control own semi-automated forces. The operators connect to the client through a web browser, and the client connects to the webSAF server. Each operator logs in to the webSAF client with a user that has a specific role, and thus controls a predefined portion of the forces, such as a battalion of mechanised infantry.

5.1.3 Map server

The map server is a shared map service serving map tiles in different formats. webSAF uses HyperText Transfer Protocol (HTTP) to request map tile images in Portable Network Graphics (PNG) format. Tiles are organized in x-, y-, and z-indexes, where the x- and y-indexes represents longitude and latitude, while the z-index represents zoom level.

5.2 Operator roles

webSAF supports three different types of operator or player roles: *manoeuvre commander*, *joint fires commander*, and *air defence commander*. The operators play against each other on either Blue or Red side. In addition, an *administrator* operator can spectate and control both sides. Functionality for more operator roles (e.g. aviation, naval, and ISTAR) will be implemented in the future.

5.2.1 Manoeuvre commander

The manoeuvre commander controls combat vehicles and infantry in VBS. Orders are currently issued at the platoon level for vehicles and at the squad level for infantry. In addition, the manoeuvre commander can request indirect fire from the joint fires commander. In the future, we also plan to add functionality for issuing orders at more levels. The aim is that each manoeuvre commander will be able to control an entire manoeuvre battalion, including organic indirect fire support entities like mortars.

5.2.2 Joint fires commander

The joint fires commander receives fire requests from the manoeuvre commanders, and prioritizes and forwards fire mission orders to the indirect fire entities in VBS. The joint fires commander can adjust how much ammunition to spend and select which units will execute the fire mission. The support for joint fires is currently limited to land-based indirect fire capabilities like tube and rocket artillery. However, we intend to implement fire support from air and sea capabilities in the future.

5.2.3 Air defence commander

The air defence commander receives air detections from the air defence sensors in VR-Forces. VR-Forces sends tracks to the webSAF server, which are then presented to the air defence commander through the GUI. For each track, VR-Forces also sends a list of launchers which can engage on that track. The air defence commander decides which track to engage, which launcher to use, and when to fire.

5.3 GUI functionality

The main component of the GUI is the *map area*. The *order of battle* (OOB) is located in the top left corner, while *notifications* are in the top right corner. The OOB and notifications are easy to hide or show by using the buttons in the upper left and right corners respectively, or by using the underlined hotkeys. The top bar shows the operator's role and the current simulation time. Selecting a unit, either in the map or in the OOB, will display information about that unit's entities at the bottom left corner and information about the unit's location, its operational status and *order queue* at the bottom right corner. Figure 5.1 shows an example of the web-based GUI for a Blue manoeuvre commander. The GUI functionality has been developed in close collaboration with military subject matter experts (SMEs).

Figure 5.1 GUI for a Blue manoeuvre commander.

5.3.1 Map area

Map navigation is similar to most so called "sliding maps" often found online, like OpenStreetMap [23] and Google Maps [24]. The map can be panned by dragging the map with the mouse. When the mouse button is released, the map continues to slide before slowing to a halt. The sliding action enables the user to pan great distances with minimal hand movement. The map view can be zoomed in or out by scrolling the mouse wheel. The bottom bar shows a map scale bar for the current zoom level, a button for accessing the *Settings menu*, and the mouse cursor coordinates in Military Grid Reference System (MGRS). The coordinate system can be changed by the operator to a latitude/longitude coordinate system or a georeferencing coordinate system using the *Settings*.

Overlaid on the map are units, spot reports, tracks, and other visual information. Map icons for friendly units are aggregated based on the OOB and zoom level. When completely zoomed in, each icon represents a platoon for vehicles and a squad for infantry.

5.3.2 Order of battle

The default OOB has a collapsible tree view, similar to the file explorer in many operating systems. The tree represents the command hierarchy of the operator's units. Through the Settings menu at the bottom bar, the operator can choose how to display the OOB. He or she can keep the default hierarchical view, or choose only to display units that can perform tasks.

To the right of the unit name is a MIL-STD-2525C symbol, indicating the type and size of the unit. Double clicking the unit name zooms and pans the map to bring that unit to the centre. Hovering over the unit with the cursor highlights that unit on the map. Conversely, hovering over a unit icon in the map highlights that unit in the OOB.

5.3.3 Notifications

The operator receives a notification when an event of interest occurs, for example when units encounter enemy forces, or artillery units are ready in a firing position. Notifications will stack on top of each other if several notifications are received. New notifications appear at the top of the list. What type of notifications the user receives depends on the operator role. The exclamation mark on the notifications button has a number that indicates the total number of active notifications. Each notification has useful information, for example "N1.343P in contact! 7 o'clock, 73 meters. 7 inf. Taking fire! Pulling back into cover", as seen from Figure 5.2. Clicking the notification does different things depending on the type of notification. For instance, clicking the contact notification in Figure 5.2 will centre the map on the specific unit that has encountered enemy forces and automatically select it. Other notifications centre the map on a position in the map where something is happening, like "Fire order underway". The time passed since the notification was received is indicated below the information. This makes it easy for the user to see which notifications are old, and which are new.

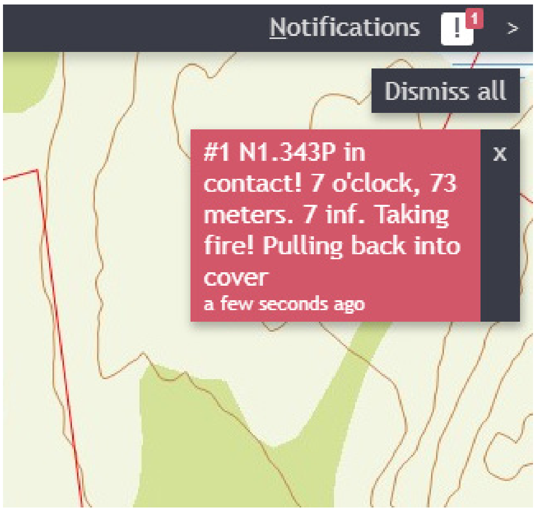


Figure 5.2 Contact notification.

5.3.4 Issuing orders and the order queue

Units have to be selected in order to be able to issue orders. Once they are selected, their icons become highlighted in the map. Right clicking the map will bring up a context menu with different options for interaction depending on the role of the operator, and which unit is selected. A manoeuvre commander will for example have the option of moving a selected infantry squad to the clicked position, assault that location or defend a position. Other operators have other options for interaction. If no units are selected, the context menu displays interactions for requesting support from other operators.

Orders are stacked in a unit's order queue as soon as an order is issued. This is seen in Figure 5.3. The unit will then execute the orders from top to bottom. An order can be deleted from the queue by clicking the red "x", and the unit will then proceed to execute the next order in the queue.

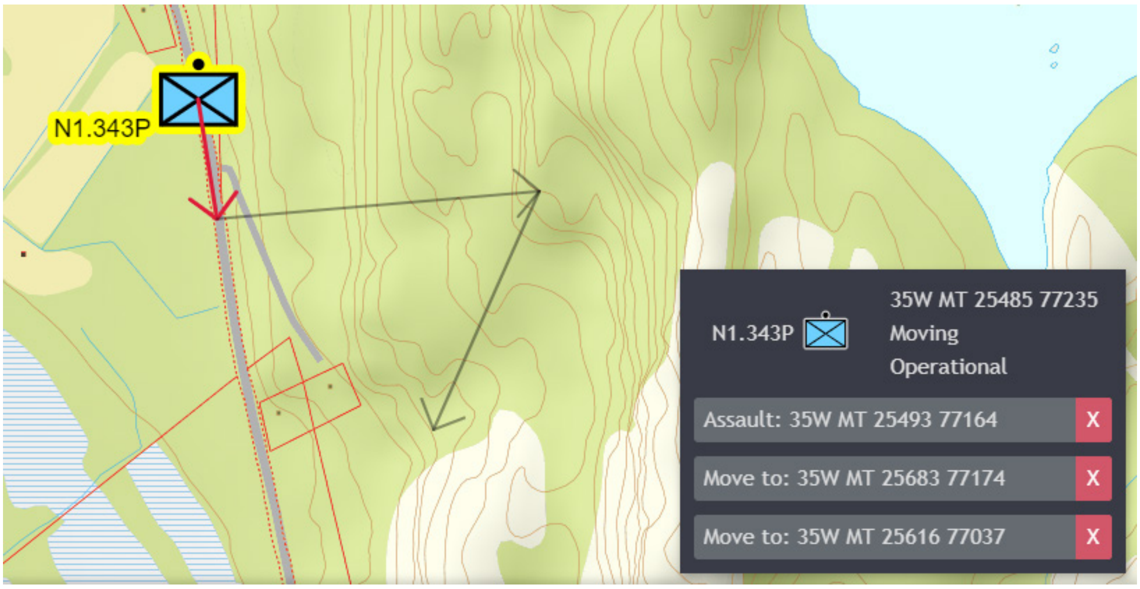


Figure 5.3 Infantry squad N1.343P is selected and has three queued orders: assault, move and move.

5.3.5 Other unit interactions

Some interactions are available by right clicking the units either in the map or in the OOB instead of clicking a position in the map. These interactions currently include mounting and unmounting infantry, and requesting logistics information. The logistics request current only displays information about ammunition, but we are considering adding support for fuel consumption in the future. Figure 5.4 shows an example of logistics information. The information is presented in an expandable/collapsible table. A bar shows how much ammunition is left. Details on the exact amount of each ammunition type is shown when expanding an item in the table.

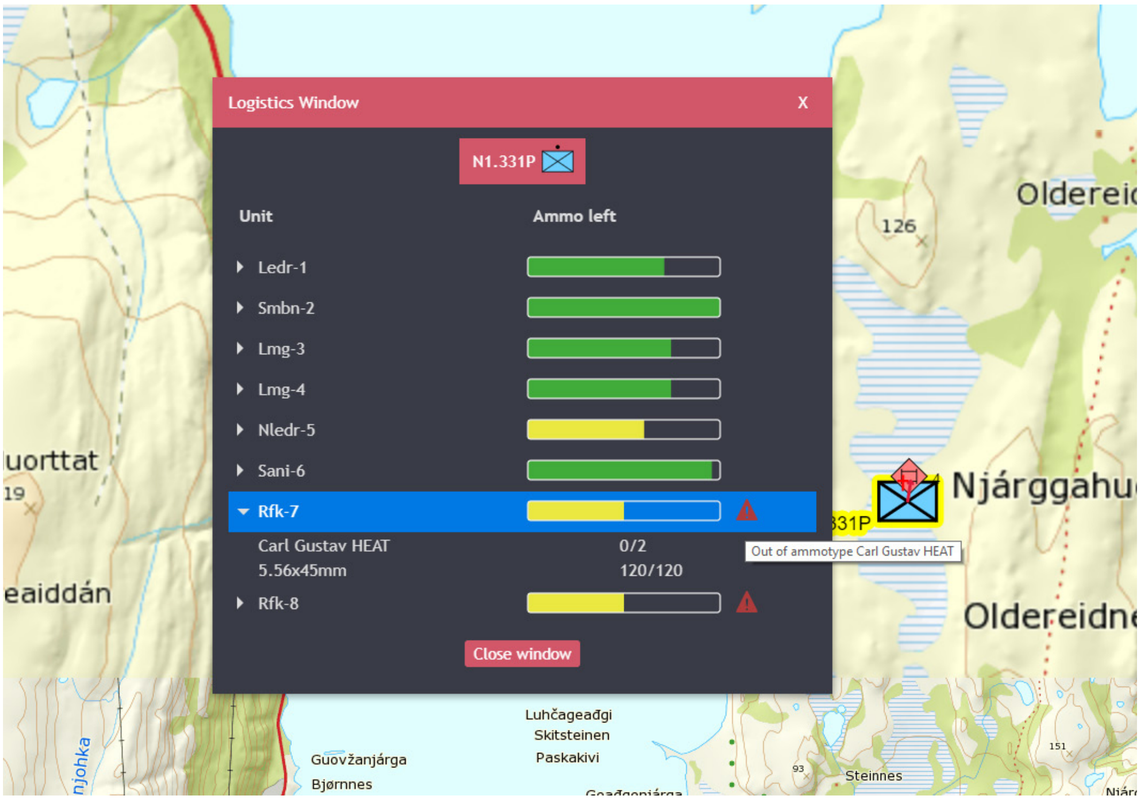


Figure 5.4 Logistics information window.

5.3.6 Fog of war

Operators cannot see entities on the opposing side, unless they are spotted by own units. When a unit spots an enemy entity, it is presented in the GUI as a MIL-STD-2525C symbol. In addition, the entity's approximate speed and direction is displayed as a labelled arrow protruding from the icon's base. If multiple enemies are spotted in close proximity to each other, the map icons will stack as illustrated in Figure 5.5. The number to the right of each icon indicates the number of that type of entity.

As long as an enemy entity is visible to friendly units, the spot report information (type, position, speed, and direction) is continuously updated in the map view. When an entity is no longer detectable by own forces, the spot report will fade and eventually disappear. The operator can set the time before they disappear in the settings.

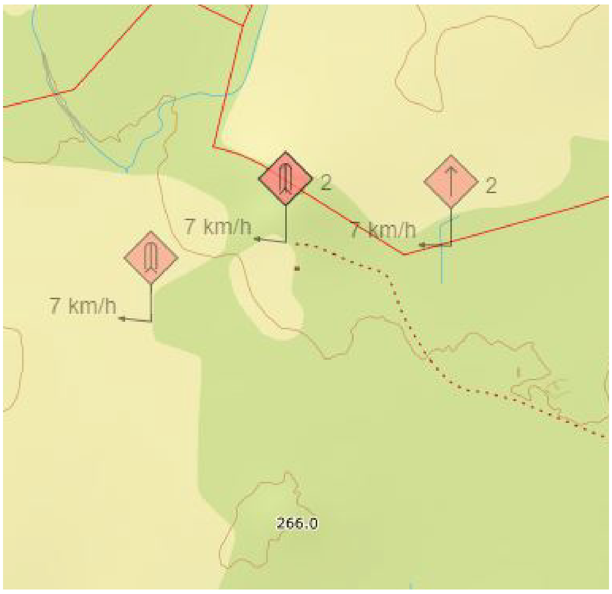


Figure 5.5 Stacked spot report icons for Blue operator.

5.4 GUI design philosophy

The GUI design philosophy for webSAF is based on balancing the following aspects, roughly prioritized in descending order:

1. *Time efficient user interaction*. The user should be able to do a lot, fast.

2. *Familiar user interaction*. The system should be easy to use.

3. *Ease of implementation*. The work needs to be completed within time and budget constraints.

4. *Ease of maintenance and extension*. The project is anticipated to expand in the future. The software architecture therefore needs to be amenable to changes and extensions.

5. *Aesthetics*. Visual style should not hinder adoption by users.

One of the main drivers behind this work was to save operator time during constructive simulation, in order to reduce the number of operators required. The GUI was therefore designed to help the operator reach his or her goals with the least amount of manual effort. Operator efficiency takes precedence over other design aspects.

Familiarity was the driving aspect for other design decisions. The “sliding maps” way of panning and zooming in and out in the map view, for example, is very similar to widely used online map services. The right-click context menu is also a way of minimizing UI clutter, while still keeping useful interactions just a click away. The context menu is complemented by a few static menus, mainly the OOB and the notifications. The user has the option to hide these, and when not hidden, they are kept to the sides as to not clutter the map. Apart from the thin bars at the top and bottom of the screen, the map fills the entire GUI area. Menus and available interactions appear on-screen when an action fires, for instance the logistics menu upon requesting logistics information. The disadvantage for a new user is that it is not immediately obvious what interactions are available. On the other hand, it can be overwhelming to see all the available options simultaneously.

Another aspect that might discourage new users is poor aesthetics. However, aesthetics is rarely prioritized in internal software projects, and slow adoption of new software and routines is a persistent problem across organizations [25]. Thus, the visual design of the GUI was given some consideration. Perceived performance is closely linked to aesthetics: Slow user interfaces can fatigue users. Performance tuning has not been given significant development time this far, but it was considered during software architecture design and when selecting external software libraries.

5.5 GUI implementation

The GUI was implemented in the TypeScript programming language [26]. TypeScript provides code type safety which helps catch errors early in development. It compiles to JavaScript that runs in modern web browsers. The web framework React was employed in order to ease the development of GUI components. React is maintained by Facebook, and provides an easy way to create scalable and responsive single page applications [27]. React also has an open source ecosystem of ready-made GUI components with permissive licenses. Some of these components were used, and others were developed in-house. The component that contains the map is rendered using OpenLayers [28], an open source web mapping library that provides similar functionality to other online maps.

An event system was used to manage the state in the beginning. As the complexity of the application grew, the state became difficult to maintain, and Redux [29] was introduced. Redux restricts how and when updates can happen by having the complete state of an application in a single store that is only updated through so-called actions. These changes are made through pure functions that take the previous state and an action, and return the new state [29].

Communication between the GUI and the webSAF server is managed through a simple event system that sends updates (in JSON format) between GUI components and the webSAF server via a WebSockets client. Figure 5.6 shows the overall architecture of the webSAF client.

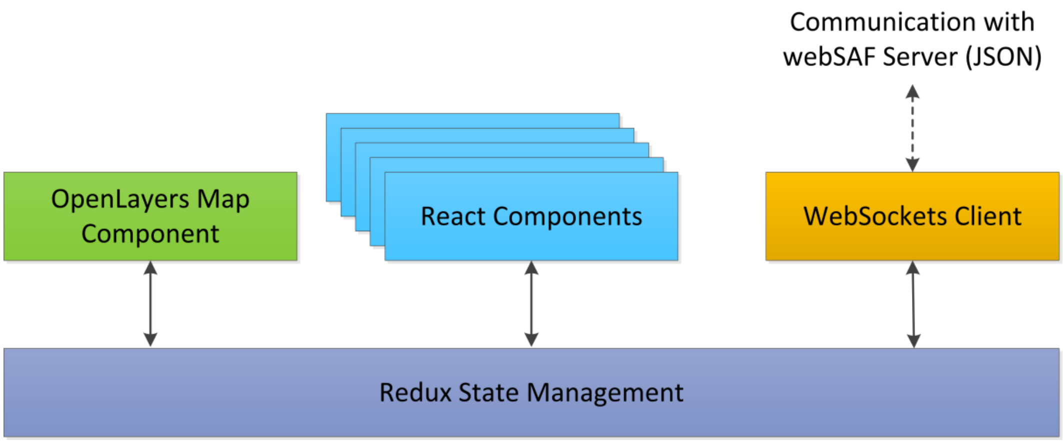


Figure 5.6 Software architecture of the webSAF client.

6 Initial experiences

The initial tests with webSAF have been successful, and we have not yet observed any significant performance issues. webSAF has been tested in several small experiments with up to four operators on each side, controlling a total of about one thousand semi-automated entities.

6.1 Overall feedback

The general feedback after internal testing at FFI is that webSAF is easy to use. It is intuitive, uncluttered and behaves as expected by the users. Testers appreciate that the webSAF client does not require anything from them at start-up, and that a crash may be fixed by a simple refresh in the browser.

The comments we have received from military officers (typical users of command and staff training systems and C2 systems) are that webSAF appears to be better than other GUI systems they have encountered before, especially with regards to accessibility, ease of use, and visual feedback. Senior military officers have also expressed that they want their command and staff training system, and likewise their C2 systems, to have GUIs with similar functionality. Because webSAF is being developed in collaboration with military SMEs and officers, feedback is received frequently and changes are implemented continuously.

6.2 Advantages to developing a web-based GUI system

There are several advantages to developing our own web-based GUI system for controlling entities in constructive simulations:

- It requires minimal hardware for the operator clients.
- No simulation software, and thus no licenses, needs to be installed on the operator client computers.
- It can be tailored to a specific use, e.g. two-sided wargaming or command and staff training.
- It is in principle independent of the simulation tools in use, and can be used to control entities in a federation of different simulation tools.
- There are a lot of tools and libraries available, many of which are open source, for developing web-based GUIs and applications.
- In principle, the web-based GUI system allows military participants to join a simulation experiment from any location, even on a classified network.

Due to its benefits, we expect to see more web-based GUIs for simulation systems in the near future, and we also expect more web-based GUIs for C2 systems. In principle, the same web-based GUI system could be used for wargaming, command and staff training, and C2, for example by defining different sets of operator roles.

7 Further work

To get a more realistic tactical AI in our simulation environment, much of the work ahead will focus on developing more, and more realistic, behaviour models. We are in the process of building a BT-based library of behaviour models for the most important battle drills for mechanized infantry platoons [17][18]. In this regard, it is also worth mentioning that the composability and modularity of BT-based behaviour models open up opportunities for collaboration on development, and sharing of behaviour models between nations using VBS.

In addition, the further development of webSAF will mainly be focused on the following areas:

- Increased automation to reduce the number of unnecessary players/operators. This will reduce the resources needed for, and thereby also lower the threshold for, conducting simulation-supported wargames with webSAF.

- Development of simulation models for more capabilities and effects to get a more complete representation of land force operations, including joint operational support functions. Examples of capabilities that need to be represented are air entities and short range air defence (SHORAD), ISTAR, combat service support (engineer, logistics and medical) and communication effects. Suitable GUI functionality and additional roles for controlling these capabilities must also be developed.

- Support for larger operations (up to brigade level).

- Development of a system for logging data from the simulations, including examining the possibility for a more sophisticated approach for logging data. In addition to logging the positions of entities, firing of weapons, and hits, it might be useful to log the behaviour of the entities in more detail, for example what orders they are following and which combat drills they use.

The simulation environment also requires further calibration and validation, and the whole system requires more extensive testing.

So far the development of webSAF has been focused on creating a working solution. In the future we also plan to look at the possibility of making the GUI system more interoperable by using the Coalition Battle Management Language (C-BML) [30] standard for sending orders to the simulated units. In addition, we plan to look at the possibility of employing the WebLVC protocol [22] for transferring entity status data from the whole simulation system (and not just VR-Forces).

In the future, it will also be interesting to compare the results from the more detailed simulations in webSAF with the results from our previous simulations, and investigate whether the results from the more detailed simulations are more useful.

8 Summary and conclusion

This report has given a description of webSAF version 1.0. webSAF is an easy-to-use, web-based GUI for controlling semi-automated entities in constructive simulations. Currently, webSAF has functionality for controlling indirect fire and manoeuvre entities simulated in VBS and air defence entities simulated in VR-Forces.

webSAF is especially tailored for simulation-supported, two-sided wargaming. The GUI functionality has been developed in close collaboration with military SMEs and officers, and is designed to be easy to use, in order to reduce the number of players/operators required.

webSAF has so far been tested in small experiments with up to four operators on each side, controlling a total of about one thousand semi-automated entities. The general feedback after internal testing at FFI is that webSAF is easy to use and does not require much practice because it is intuitive and behaves as expected by the users. Military officers have expressed that webSAF appears to be better than other GUI systems they have encountered before, and that they want their command and staff training system, and their C2 systems, to have GUIs with similar functionality

Developing a web-based GUI system for controlling entities in constructive simulations has several advantages. It requires only minimal hardware, and no simulation software, for the operator clients. In addition, it can be tailored to a specific use (e.g. two-sided wargaming or command and staff training), and is in principle simulation system agonistic. We believe the web-based GUI approach is the way ahead to make constructive simulations and also C2 systems more accessible and easier to use.

References

[1] P-I. Evensen & D.H. Bentsen, *Simulation of land force operations – a survey of methods and tools*, FFI-rapport 2015/01579, 2016.

[2] P-I. Evensen, K. Selvaag, D.H. Bentsen & H. Stien, "Web-Based GUI System for Controlling Entities in Constructive Simulations", *Proceedings of the Interservice/Industry Training, Simulation and Education Conference (I/ITSEC) 2017*, Paper No. 17043, 2017.

[3] P-I. Evensen, K. Selvaag, D.H. Bentsen, H.R. Holhjem & H. Stien, "Easy-to-Use, Web-Based Graphical User Interface for Controlling Entities in Constructive Simulations", *Proceedings of the NATO Modelling and Simulation Group (NMSG) Annual Symposium 2018 (STO-MP-MSG-159)*, Paper No. 7, 2018.

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

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

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

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

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

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

[10] 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.

[11] P-I. Evensen, S.E. Martinussen, M. Halsør & D.H. Bentsen, "Wargaming Evolved: Methodology and Best Practices for Simulation-Supported Wargaming", *Proceedings of the Interservice/Industry Training, Simulation and Education Conference (I/ITSEC) 2019*, Paper No. 19182, 2019.

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

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

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

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

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

[17] P-I. Evensen, H. Stien & D.H. Bentsen, "Modeling Battle Drills for Computer-Generated Forces using Behavior Trees", *Proceedings of the Interservice/Industry Training, Simulation and Education Conference (I/ITSEC) 2018*, Paper No. 18081, 2018.

[18] P-I. Evensen, H. Stien & D.H. Bentsen, *Modelling Human Behaviour using Behaviour Trees*, FFI-rapport 18/01651, 2018.

[19] D. Isla, "Handling Complexity in the Halo 2 AI", *Proceedings of the Game Developers Conference (GDC) 2005*, 2005.

[20] I. Millington & J. Funge, *Artificial Intelligence for Games*, 2. Edition, Morgan Kaufmann, 2009.

[21] gRPC, https://grpc.io.

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

[23] OpenStreetMap, https://www.openstreetmap.org.

[24] Google Maps, https://www.google.com/maps.

[25] B.H. Hall & B. Khan, *Adoption of New Technology*, National Bureau of Economic Research, Cambridge, Massachusetts, 2003.

[26] TypeScript – JavaScript that Scales, https://www.typescriptlang.org.

[27] React – A JavaScript Library for Building User Interfaces, https://reactjs.org.

[28] OpenLayers, from https://openlayers.org.

[29] Redux – A Predictable State Container for JavaScript Apps, https://redux.js.org.

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

Abbreviations

AAR	After Action Review
AI	Artificial Intelligence
API	Application Programming Interface
AR	Augmented Reality
ARMA	Armed Assault
BISim	Bohemia Interactive Simulations
BT	Behaviour Tree
C2	Command and Control
C-BML	Coalition Battle Management Language
CGF	Computer-Generated Forces
COTS	Commercial Off-The-Shelf
DLL	Dynamic Link Library
GESI	GEfechts-SImulation System
GUI	Graphical User Interface
HLA	High Level Architecture
HTML	HyperText Markup Language
HTTP	HyperText Transfer Protocol
IFV	Infantry Fighting Vehicle
ISTAR	Intelligence, Surveillance, Target Acquisition, and Reconnaissance
JSON	JavaScript Object Notation
M&S	Modelling and Simulation
MBT	Main Battle Tank
MGRS	Military Grid Reference System
MSaaS	Modelling and Simulation as a Service
NATO	North Atlantic Treaty Organization
NPC	Non-Player Characters
OOB	Order of Battle
PNG	Portable Network Graphics
RPC	Remote Procedure Call
SAF	Semi-Automated Forces
SDK	Software Development Kit
SHORAD	Short Range Air Defence
SME	Subject Matter Expert
UAV	Unmanned Aerial Vehicle
UGV	Unmanned Ground Vehicle
VBS	Virtual Battlespace

About FFI

The Norwegian Defence Research Establishment (FFI) was founded 11th of April 1946. It is organised as an administrative agency subordinate to the Ministry of Defence.

FFI's MISSION

FFI is the prime institution responsible for defence related research in Norway. Its principal mission is to carry out research and development to meet the requirements of the Armed Forces. FFI has the role of chief adviser to the political and military leadership. In particular, the institute shall focus on aspects of the development in science and technology that can influence our security policy or defence planning.

FFI's VISION

FFI turns knowledge and ideas into an efficient defence.

FFI's CHARACTERISTICS

Creative, daring, broad-minded and responsible.

Om FFI

Forsvarets forskningsinstitutt ble etablert 11. april 1946. Instituttet er organisert som et forvaltningsorgan med særskilte fullmakter underlagt Forsvarsdepartementet.

FFIs FORMÅL

Forsvarets forskningsinstitutt er Forsvarets sentrale forskningsinstitusjon og har som formål å drive forskning og utvikling for Forsvarets behov. Videre er FFI rådgiver overfor Forsvarets strategiske ledelse. Spesielt skal instituttet følge opp trekk ved vitenskapelig og militærteknisk utvikling som kan påvirke forutsetningene for sikkerhetspolitikken eller forsvarsplanleggingen.

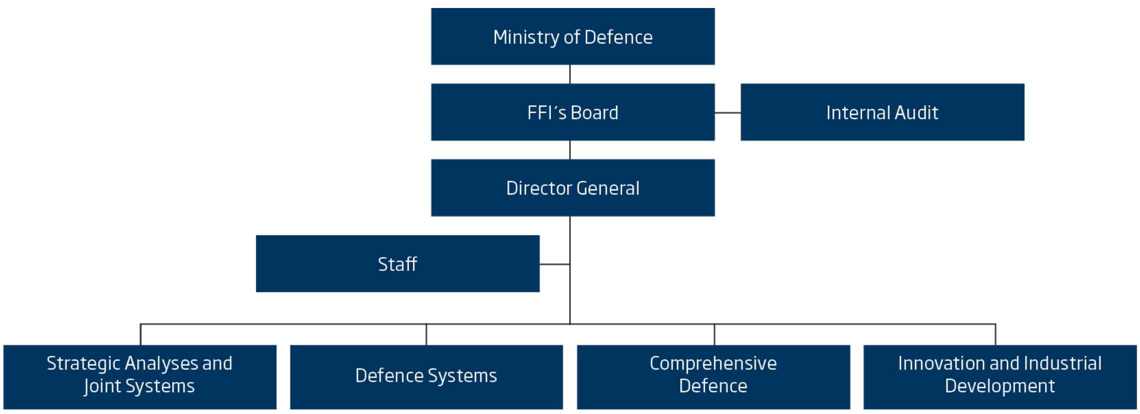
FFIs VISJON

FFI gjør kunnskap og ideer til et effektivt forsvar.

FFIs VERDIER

Skapende, drivende, vidsynt og ansvarlig.

FFI's organisation



Forsvarets forskningsinstitutt
Postboks 25
2027 Kjeller

Besøksadresse:
Instituttveien 20
2007 Kjeller

Telefon: 63 80 70 00
Telefaks: 63 80 71 15
Epost: ffi@ffi.no

Norwegian Defence Research Establishment (FFI)
P.O. Box 25
NO-2027 Kjeller

Office address:
Instituttveien 20
N-2007 Kjeller

Telephone: +47 63 80 70 00
Telefax: +47 63 80 71 15
Email: ffi@ffi.no

FFI Forsvarets forskningsinstitutt
Norwegian Defence Research Establishment