Defence Research and Development Canada | Recherche et développement pour la défense Canada

DRDC | RDDC
technologysciencetechnologie

Urban Team Experimentation Requirements

David Tack and Andrew Morton
HumanSystems® Incorporated

Scott Arbuthnot
CogSim Technologies Inc

Prepared by:
HumanSystems® Incorporated, Guelph, ON N1H 3N4
CogSim Technologies Inc, Ottawa, ON

PSPC Contract Number: W7701-166107/001/QCL
Technical Authority: Justin Hollands, DRDC – Toronto Research Centre
Contractor's date of publication: March 2017

Defence Research and Development Canada

Contract Report
DRDC-RDDC-2018-C125
July 2018

CAN UNCLASSIFIED

Canadä

IMPORTANT INFORMATIVE STATEMENTS

This document was reviewed for Controlled Goods by Defence Research and Development Canada (DRDC) using the Schedule to the *Defence Production Act*.

URBAN TEAM EXPERIMENTATION REQUIREMENTS

by:

David Tack[1], Andrew Morton[1], and Scott Arbuthnot[2]

[1] Human*Systems*® Incorporated
Guelph, ON N1H 3N4

[2] CogSim Technologies Inc.
Ottawa, ON

HSI® Project Manager: David W. Tack
(519) 836-5911

PWGSC Contract No. W7701-166107/001/QCL
Task Authorization No. 001

On Behalf of

RDDC-R et D Défense Canada - Valcartier
DRDC-Defence R&D Canada - Valcartier
Bâtisee 53
24659 Route de la Bravoure
Quebec G3JX5

As represented by

DRDC – Toronto Research Centre
1133 Sheppard Avenue West
Toronto, ON, M3K 2C9

Project Scientific Authority:

Justin Hollands
(416) 635-2073

March 2017

Abstract

The development of digital battlefield networks and robust new computing, sensing, and communications technologies promise improved effectiveness on the battlefield. However, significant questions exist regarding the true effectiveness and value of these systems, methods of best employment, and the potential burden of overloading soldiers with information on the battlefield. Defence Research and Development Canada at Toronto (DRDC-T) needs to determine the requirements for a future Urban Team Experimentation (UTE) laboratory that can be used to evaluate the claims and the implications of these information systems for future soldier systems.

Through consultation with key scientific and engineering stakeholders at Defence Research and Development Canada's Toronto laboratory, goals and objectives were reviewed and capability requirements for the UTE were identified. Candidate virtual soldier software systems, and integrated hardware options, were evaluated against these capability requirements. A framework was also discussed for developing infantry scenarios to employ in UTE experimentation. The findings of these investigations are discussed and recommendations are provided.

Résumé

Le développement de réseaux de champs de bataille numériques et de nouvelles technologies robustes d'informatique, de détection et de communication promet une efficacité accrue sur le champ de bataille. Cependant, d'importantes questions subsistent quant à l'efficacité et à la valeur réelle de ces systèmes, aux meilleures méthodes d'utilisation et au fardeau potentiel que représente la surcharge d'information pour les soldats déployés sur le champ de bataille. Recherche et développement pour la défense Canada (RDDC) – Centre de recherches de Toronto doit déterminer les besoins d'un futur laboratoire d'expérimentation par une équipe urbaine (EEU) afin d'évaluer les prétentions et les implications de ces systèmes d'information pour les futurs systèmes de soldat.

En consultation avec les principaux intervenants scientifiques et techniques du laboratoire de RDDC à Toronto, nous avons examiné les buts et les objectifs recherchés et déterminé les besoins en matière de capacités EEU. Les systèmes logiciels de soldat virtuel en projet et les options matérielles intégrées ont été évalués en fonction de ces besoins en capacités. On a également discuté d'un cadre d'élaboration de scénarios d'infanterie EEU. Les conclusions de ces enquêtes font l'objet d'une discussion et des recommandations sont énoncées.

Executive Summary

URBAN TEAM EXPERIMENTATION REQUIREMENTS

David Tack, Andrew Morton, and Scott Arbuthnot, Human*Systems*® Incorporated; Defence R&D Canada – Toronto; March 2017.

The development of digital battlefield networks and robust new computing, sensing, and communications technologies has seen an explosion of new information products and capabilities for the future soldier. While these new systems and software promise to bring new levels of performance and effectiveness to the battlefield, significant concerns exist regarding the true effectiveness and value of these systems, methods of best employment, and the potential burden of overloading soldiers with information on the battlefield.

Defence Research and Development Canada at Toronto (DRDC-T) has employed virtual environments to support studies of soldier teams and information systems for many years now. There is currently a need to determine the requirements for a future Urban Team Experimentation (UTE) laboratory that can meet the projected goals and objectives of future soldier team studies, to determine which software and hardware changes are necessary or recommended, and to review how scenarios for soldier missions might be better framed for such future studies.

Through consultation with key scientific and engineering stakeholders at Defence Research and Development Canada's Toronto laboratory, goals and objectives were reviewed and capability requirements for the UTE were identified. Candidate virtual soldier software systems, and integrated hardware options, were evaluated against these capability requirements. As well, a framework was discussed for developing infantry scenarios to employ in UTE experimentation.

While no one software solution will currently meet all the requirements in their entirety most of the core UTE requirements can be achieved by the VBS2/3 software currently in use in DRDC-T. In the case of very specialized capability requirements, game engine software, which allows for a much deeper level of interfacing into the functionality of the software, could be used selectively as a more suitable alternative to VBS. Recommendations are provided for the development of higher level configuration, control and data capture systems to improve the speed and ease of software design and extraction of data.

A scenario framework was outlined for organizing participants, confederates, and bots in a scenario formation, and strategies were identified for balancing scenario characteristics according to the organizational level of interest in team experimentation. Perspectives on scenario complexity and methods for setting and controlling such complexity were discussed in the context of scenario development. Based on this investigation, further work was proposed to realize and operationalize this framework.

Table of Contents

List of Figures

List of Tables

1. Introduction

The development of digital battlefield networks and robust new computing, sensing, and communications technologies has seen an explosion of new information products and capabilities for the future soldier. While these new systems and software promise to bring new levels of performance and effectiveness to the battlefield, significant concerns exist regarding the true effectiveness and value of these systems, methods of best employment, and the potential burden of overloading soldiers with information on the battlefield.

Human factors studies of new information capabilities and products can be undertaken live in the field with soldiers and in virtual environments in the laboratory using computer-generated avatars, equipment, weapons effects, and terrain. Virtual environments offer some advantages over live field testing. Certain information products can be simulated on the virtual battlefield long before they are available for field testing. Information content and display interfaces can be manipulated in virtual space to evaluate alternative designs for usability faster, cheaper, and more easily than physical systems. While there is a loss of realism in a virtual battlespace it is easier to simulate a much broader range of terrain, lighting levels, and climate conditions then what is reasonably achievable in field testing and the circumstances of the battle are often easier to control. Virtual environments also enable the collection of certain types of data that can be more difficult or even impossible to collect in live field testing.

Defence Research and Development Canada at Toronto (DRDC-T) has employed virtual environments to support studies of soldier teams and information systems for many years now. The introduction of a new Individual Small Group Testing Facility, co-located near the current Soldier Team Laboratory, will expand the possibilities for studying soldier teams and new information systems. This presents an ideal opportunity to assess the needs and requirements of a future Urban Team Experimentation (UTE) laboratory to meet the projected goals and objectives of future soldier team studies, to determine which software and hardware changes are necessary or recommended, and to review how scenarios for soldier missions might be better framed for such future studies.

1.1 Aims

The aims of this project included:

a) Development requirements for the UTE through consultation with key stakeholders.

b) Evaluate current and candidate virtual soldier software for team studies.

c) Comment on suitability of different software and hardware options to meet UTE requirements.

d) Provide a framework for developing structured soldier mission scenarios.

1.2 Acronyms

AAR	After Action Review
AFRL	Air Force Research Laboratory
AHRS	Altitude Heading Reference System
AI	Artificial Intelligence
API	Application Programming Interface
CAF	Canadian Armed Forces
CGF	Computer-Generated Forces
C-PETS	Coalition Performance Evaluation Tracking System
DIS	Distributed Interactive Simulation
DAGR	Defense Advanced GPS Receiver
DND	Department of National Defence
DRDC	Defence Research and Development Canada
EXCON	Exercise Control
GFE	Government Furnished Equipment
GSM	Government Supplied Materials
HLA	High-Level Architecture
ISS	Integrated Soldier System
MET	Mission Event Timeline
NGO	Non-Governmental Organization
NVG	Night Vision Goggle
OPFOR	Opponent Force
PDU	Protocol Data Unit
PI	Principal Investigator
ROV	Remotely Operated Vehicle
S&T	Science & Technology
SA	Scientific Authority
SME	Subject Matter Expert
SRCL	Security Requirements Checklist
TA-BTS	Tactical Aviation Battle Task Standard
TCPS	Tri-Council Policy Statement
TTPs	Tactics Techniques and Procedures
UAV	Unmanned Aerial Vehicle
UGV	Unmanned Ground Vehicle
UTE	Urban Team Experimentation
VBS2/3	Virtual Battle Space 2/3
VOIP	Voice Over Internet Protocol
ZPD	Zone of Proximal Development

2. Approach

There were three major thrusts in this project:

1. UTE requirements development.
2. Assessment of current software/hardware to meet these requirements.
3. Scenario framework development.

2.1 UTE Requirements Development:

- The general goals and objectives of the UTE laboratory were identified.
- Meetings were held with key scientific and engineering stakeholders at DRDC-T to derive and discuss requirements for UTE experimentation.
- Requirements were drafted and reviewed/confirmed by DRDC-T stakeholders.

2.2 Software/Hardware Assessments

- Three categories of software (i.e. military training, dual use, and game engines) were reviewed for their suitability to meet UTE requirements.
- Opportunities to integrate input control hardware were reviewed and options identified.
- Integrating other soldier system information systems with the software simulation were also considered.

2.3 Scenario Framework Development

- Organizational Use Case were considered to derive platoon force structures of participants, confederates, and bots, according to the likely experimentation focus at each of three levels (i.e. soldier in a Section context, Section leaders in a platoon context, and Platoon commanders in a Company context).
- Given these three organizational levels, previous scenario development efforts in soldier systems were reviewed to determine which scenario characteristics are most likely to vary according to level of testing.
- Controlling scenario complexity is essential in soldier system information system experimentation. The literature in scenario complexity and past scenario development efforts in soldier systems were reviewed to propose a model for the scenario framework.

3. Results

The results of this project are organized into the following three Sections.

a. Goals and Requirements
b. Software Assessments
c. Options to fulfill Requirements
d. Scenario Development Framework

3.1 Goals and Requirements

Key stakeholders provided the goals and associated requirements for the Urban Team Experimentation (UTE) laboratory.

3.1.1 Able to Study New Information Capabilities

A key research goal for the UTE is the ability to study and investigate new and emerging information system capabilities. The following requirements expand on this goal.

3.1.1.1 Assess current and future Integrated Soldier System (ISS) capabilities

The UTE shall be capable of simulating current (Build 1) ISS capabilities, enhanced Build 1 capabilities (e.g. rangefinder, camera), likely Build 2 software upgrades, and possible future ISS capabilities. The current ISS Build 1 includes a soldier system hand-held Tactical User Interface device (i.e. wired smartphone), a Harris radio, batteries, wired and wireless Push-To-Talk (PTT), headset, and a hub connection for adding other components. Current enhanced capabilities include the connection of a digital camera, DAGR, USB memory stick, and/or a CORAL-CR-C thermal imager. Build 2 would provide updates and changes to the software, and Build 3 could introduce hardware and software changes. Ideally, the UTE would be capable of including or simulating these ISS systems as they exist now and in the future.

3.1.1.2 Integrate real hardware into the simulation

The UTE shall be capable of integrating computing hardware devices into the virtual “gaming” software such that these devices can exchange critical data and information. The objective requirement would see the integration of the actual ISS device into the UTE. The threshold requirement would see a tablet, or similar ISS simulation device, integrated into the UTE.

3.1.1.3 Enable the use of multi-modal display interfaces

The UTE shall be able to output game information to alternative multi-modal displays. Secondary visual displays (e.g. tablet) could be used to display other key information (e.g. situation awareness information, physiological monitoring data). Audio information shall be capable of being displayed to a common speaker or to head-mounted earphones. Tactile displays shall be capable of being integrated to display alert and alarms, wayfinding directions, etc.

3.1.1.4 Integrate alternative input interfaces

The UTE shall support external hardware interface devices. As an example, the game shall accept input controls from an instrumented assault rifle to enable the player to move and engage targets while handling the weapon alone.

3.1.2 Realistic Dismounted Infantry Simulation

One of the advantages of field testing is realism. A minimum level of realism is necessary in any software simulation to ensure the fidelity of the virtual experience is sufficiently immersive and "real" to elicit representative behaviours and actions from and between the study participants. The following requirements expand on this goal.

3.1.2.1 Realistic avatar appearance

The UTE shall provide the capability to represent real (actual human) and bot (computer controlled) avatar players in the software simulation with realistic clothing and equipment for the role they represent. Entities in the game will be able to be configured to represent realistic ethnicities, genders, occupations, head/facial hair, etc.

3.1.2.2 Generate recognizable avatars and gestures

Realistic recognition and interaction between real players in a game is a challenge. The UTE shall be capable of skinning the face of an avatar with the likeness of the real player to aid in player recognition while immersed in the game. The UTE shall also enable realistic gestures and motions within the game to enable non-verbal communication between real players.

3.1.2.3 Realistic bot behaviours and actions

The UTE shall enable the experimenter to assign behavioural responses and actions to bots in the game. For example, upon encountering blue-force members bots could be programmed to behave as a hostile (e.g. from avoidance to aggression to engagement with fire), or as a neutral (e.g. passive, observant, but non-interactive), or as a friendly (e.g. from welcoming to supportive).

3.1.2.4 Urban terrain simulation suitable to support team experimentation

Effective team experimentation requires sufficient terrain complexity to create informational challenges in positional location and situational awareness among elements in a larger team and battlespace.

Urban terrain shall be represented by multi-story buildings, roads, alleys, and open areas bounded by buildings. Terrain shall be able to be populated with context-specific items to dimensionalize the realism of the spaces (e.g. vehicles, trees, fuel drums, market stalls).

3.1.2.5 Representative night vision capability

For low-light and night operations, players equipped with night vision devices (e.g. NVGs, weapon optics) shall be provided with a simulated night vision view of the world. Ideally, this night vision view will be capable of simulating both image intensification and infra-red devices.

3.1.2.6 Represent remote-controlled devices

Remote devices are becoming more common on the modern battlefield (e.g. drones, UGV, remote cameras). The UTE shall be capable of representing these remote devices and their functionality, and be capable of being programmed to move through a pre-planned route, operated by a player in the game, or operated by an operator outside of the game play. Imagery from these devices shall be able to be remotely viewed by players in the game.

3.1.2.7 Integrate real player consequence system

To encourage more realistic shot avoidance behaviour the UTE shall be capable of triggering a shot consequence system (e.g. shock belt), worn by the person whose avatar is shot in the game.

3.1.3 Real-time Data Capture & Analyses

The UTE needs to support the data collection and dissemination of key information among participants for real/bot formations up to Platoon size. The following requirements expand on this goal.

3.1.3.1 Support Platoon-sized missions

The UTE shall support dismounted infantry operations up to and including Platoon-sized operations.

3.1.3.2 Update information to participants in play

As location and situation awareness information changes during the game, the UTE will update information displays to players in real time during game play.

3.1.3.3 Capture all communications with location and time

The UTE shall capture communications in real-time between blue-force players during game play. Voice communications will be recorded and stamped for time/date, player, role, and network.

3.1.4 Experimentation Configuration Dashboard

To support experimentation into select soldier system functionality, the UTE shall enable the experimenter to vary and control key aspects of the simulation. The following requirements expand on this goal.

3.1.4.1 Toggle experimentation states and conditions

The UTE shall enable the experimenter to vary game functionality and environmental conditions. Functional conditions could include error variability and data update delays (e.g. position location information displayed to players in real time). Environmental conditions could include time of day, precipitation, visual obscuration, etc.

3.1.4.2 Assignable capabilities to each individual player

The UTE shall provide the experimenter with the capability to vary the functional ISS capability of each player and be able to change a player's role and functionality during game play.

Radio nets shall be able to be assigned to participants according to an ALCON net, command net, Pl net, Section net, and Assault Group net.

3.1.4.3 Manage real and bot player roles and assignments

The UTE shall enable the experimenter to assign roles and functionality to both real and bot players. Real and bot players shall be able to be assigned to organizational groupings in both blue, red, other combatant, and non-combatant units.

3.2 General Software Assessment

The following section compares the baseline capabilities of five candidate software solutions for the UTE.

3.2.1 Introduction

Much of the current and future experimentation done by the UTE involves the use of software capable of representing soldiers, equipment and supporting capabilities such as weapons fire and effects. To date the UTE has made extensive use of Virtual Battle Space 2 (VBS2) from Bohemia Interactive. This software is available from the Department of National Defence (DND) who hold an enterprise license for the software and its development tools. VBS2 is a tried and proven tool which allows for a great deal of the desired capabilities out of the box. As such, VBS2 and its latest follow-on version VBS3 will very likely remain the core development software component for the lab however there are areas of specific interest where other software may provide a better platform. The following sections will look to address and highlight the pros and cons of the major common software platforms that could be considered.

3.2.2 Software Platforms

All the software platforms that are capable of supporting experimentation as described by the DRDC-T UTE for soldier representation fall generally into two categories, those being purpose built and game engines. Purpose-built software platforms such as VBS2/3 and Mak Engage have the benefit of covering a great deal of fundamental capabilities for soldier representation with little to no required development. Unfortunately, these systems do have limitations when looking at developing unique capabilities and/or hardware integration. On the other hand, general purpose game engines provide almost unlimited development capabilities at the price of having almost no prebuilt functionality (other than at the most general level) to represent soldiers and equipment.

Note as well that modifiable games such as America's Army and Arma3, etc. are not being considered as viable solutions. This is because they lack core API level access and true service support mechanisms. The software being reviewed in the category of game engine include the top-level systems available to date and as they all offer virtually identical capabilities they will be collectively referred to as "game engines".

There are some "middle ground" software platforms that somewhat span the gap between these two categories however none represents the perfect combination or solution. Although there are other software platforms not covered here, the following information represents the top choices with respect to professionally fielded and supported systems. The evaluated software by category are listed below:

Table 1: Software Platforms

MILITARY TRAINING SYSTEMS	
VBS2 / VBS3	
Mak Engage	

DUAL USE	
Unigine Sim2	
GAME ENGINES	
Unreal Engine	

Unity Engine	
Cry Engine	

It should be noted that an assessment of higher level support software in the computer-generated forces (CGF) category are not being assessed as these systems, although integral to many experiments, are not designed to represent the individual functionality required by the intended experimentation.

3.2.3 Feature Analysis

The table listed below provides a high-level overview of potential core software against requirements and other factors such as cost. The next section will break down each software product against the identified UTE requirements.

Table 2: Software Baseline Capability Comparison

FEATURE	VBS2/3	ENGAGE	UNIGINE SIM 2	GAME ENGINES
Cost	DND License	$5.6K/User	$10K/Developer $1.5K/Runtime	Free Non-Commercial Use
Ready to use	Yes	Yes	No	No
VR Capable	Yes (not tested)	In Development	Native	Native
Physics	Yes	Yes	Yes	Yes
Dynamic Sound	Yes	Yes	Yes	Yes
Rendering Quality	Medium	Medium	High	High
Terrain Size	Large	Large	Large	Small
Terrain Destruction	Some	Some	Some	Possible
Custom Terrain Tools	Included	Not Included	Included	Game Level Design Tools
Integration of Real Hardware	Possible	Possible	Possible	Possible
Custom Input Devices	Possible	Possible	Possible	Possible
Run on Tablets/Phones	No	No	Native	Native
Realistic Human Characters	Yes	Yes	Yes	Yes

FEATURE	VBS2/3	ENGAGE	UNIGINE	GAME ENGINES
Custom Face Avatars	Yes	Yes	Yes	Yes
Custom Animations	Yes	Yes	Yes	Yes
Fully Controllable Skeleton System	Partial	No	Yes	Yes
AI System	Yes	Yes	Scripted	Scripted
NVG/Thermal Support	Both	Both	Both	Development Required
Equipment Properties	Included	Some	Development Required	Development Required
Multiple Cameras	Yes	Yes	Yes	Yes
UAV/UGVs	Included	Included	Development Required	Development Required
Platoon Sized Units	Yes	Yes	Yes	Yes
Custom User Interfaces	Yes	Yes	Yes	Yes
Voice Comms	Yes	Yes	Yes	Yes
Radio Comms	Yes	Yes	Development Required	Development Required
Distributed Simulation	Native	Native	Native	Development Required
Vehicles	Yes	Yes	Yes	Yes
Multi-Player Control	Partial	Via CGF	Dev. Required	Dev. Required

3.3 Options to fulfill Requirements

Software and hardware options are reviewed and discussed in this section, in the context of key stakeholder goals and requirements from Section 3.1 (shown as italicized text at the start of each sub-section below). This assessment assumes a desktop or non-fully immersed participant interface system.

3.3.1 Able to Study New Information Capabilities

3.3.1.1 Assess Current and Future ISS Capabilities

"*The UTE shall be capable of simulating current (Build 1) ISS capabilities, enhanced Build 1 capabilities (e.g. rangefinder, camera), likely Build 2 software upgrades, and possible future ISS capabilities*."

Integration or simulation of ISS capabilities represents some unique challenges. Although integration of the real hardware would be ideal, this approach may not be practical unless the device includes a programmable API that can be used to override or emulate behavior. Integration to VBS2/3 location becomes an issue in two respects. The first issue is mapping. Either a special map must be created from VBS2/3 data and geocoded offline for use in the ISS or a VBS2/3 scenario map would need to be built based on a real-world location. Both issues however would not be usable with location data internal to the ISS. Assuming the ISS uses an internal GPS for location, entity locations would not correspond to the map data unless there is a software path capable of overriding or spoofing the GPS input. There would also need to be a path to allow locational information to be passed in for any computer-generated forces that form part of the scenario.

A likely more viable solution to ISS integration would be to emulate the device in software and hardware as a surrogate. This would allow proper handling of all locational information as well as a much more open methodology for modification or addition of capabilities. The issue of map integration with VBS2/3 as discussed above would still need to be addressed.

It may also be viable to create a simulated version of the ISS device within VBS2/3 or other selected software. Previous DRDC-T work to emulate the Blue Force Tracker within VBS2 could be expanded to fully emulate the ISS. Physical interaction with the device would not be possible in this solution however that may or may not be an issue depending on intended experimentation.

3.3.1.2 Integrate Real Hardware into Simulation

"*The UTE shall be capable of integrating computing hardware devices into the game such that these devices can exchange critical data and information. The objective requirement would see the integration of the actual ISS device into the UTE. The threshold requirement would see a tablet, or similar ISS simulation device, integrated into the UTE.*"

Hardware integration can be on many different levels and dependencies on software. Hardware that is independent of core soldier representation software such as VBS2/3 should be easily integrated. Inter-device communication can be handled via Wi-Fi, custom network communications or via override of the DIS data packet.

Hardware integration may also require input/output to the central simulation framework. This is normally accomplished either via the distributed network traffic (DIS or HLA) or directly via an API depending on information required and the level of interaction.

Hardware that requires a graphical representation of itself within the simulation framework will require custom software development via an API for the framework. Should the hardware require a graphical representation of the simulations environment it is essential that the representation be tightly correlated. As an example, if hardware depicts a 2D map of the scenario area and the scenario is based on non-real-world terrain or modified real world terrain, a method of creating or representing that terrain will be required.

3.3.1.3 Enable the use of multi-modal Display Interfaces

"The UTE shall be able to output game information to alternative multi-modal displays. Secondary visual displays (e.g. tablet) could be used to display other key information (e.g. situation awareness information, physiological monitoring data). Audio information shall be capable of being displayed to a common speaker or to head-mounted earphones. Tactile displays shall be capable of being integrated to display alert and alarms, wayfinding directions, etc."

All the potential software packages discussed have the capability to support this requirement. Game engines offer complete access to the source code and the others allow access either via network traffic for macro level items or through programming API for more discrete information.

Audio setup in all cases should be achievable using hardware only and not require code or script modifications.

3.3.1.4 Integrate Alternative Input Interfaces

"The UTE shall support external hardware interface devices. As an example, the game shall accept input controls from an instrumented assault rifle to enable the player to move and engage TARGETs while handling the weapon alone."

External hardware device interface can be accomplished in any of the proposed software technologies with some limitations. Static or semi-static equipment such as radios and laser designators are relatively easy to integrate however more dynamic items such as the weapon example can be quite complex. For any dynamic body the hardware must be tracked for both position and orientation either internally or externally to a resolution capable of aiming the weapon at near and far range target. This equipment tracking must also be synchronized to the character operator of the equipment.

Additionally, complex functionality of hardware equipment, such as stoppages or magazine changes for a weapon, may be difficult to represent in the software. Physical changes to the hardware equipment that are triggered from the software itself are possible but very difficult and expensive to achieve. Any integration at this level of fidelity should be achievable however it will require in-depth development.

Templeman (2014) noted that the validity of simulation depends on the behavioural realism of the User. Pointman was developed by the U.S. Naval Research Laboratory in a desktop application to improve the usability and physical realism of the User interface with VBS2 by providing a more realistic mapping of natural control actions to resulting avatar movement. Pointman uses sliding rudder pedals to simulate walking and crawling movement and to switch between standing and kneeling; a game controller is used to turn the body, side-step, pitch and roll the rifle; and a head tracker to control the avatar's head position and view, aim the rifle, and lean the torso to either side (see Figure 1).

Templeman (2014) suggests that this use of motor substitution achieved more realistic control of one's avatar in virtual space by using movement abstractions that avoid the mismatches in perceptions between reality and virtual actions in fully immersed motion capture systems. An assessment of the Pointman interface with U.S. Marines resulted in high rates of User acceptance over traditional computer interface methods alone. At an estimated $400 USD per Pointman suite it is a cost-effective means of improving the realism of action. More information is provided about Pointman in Annex A.

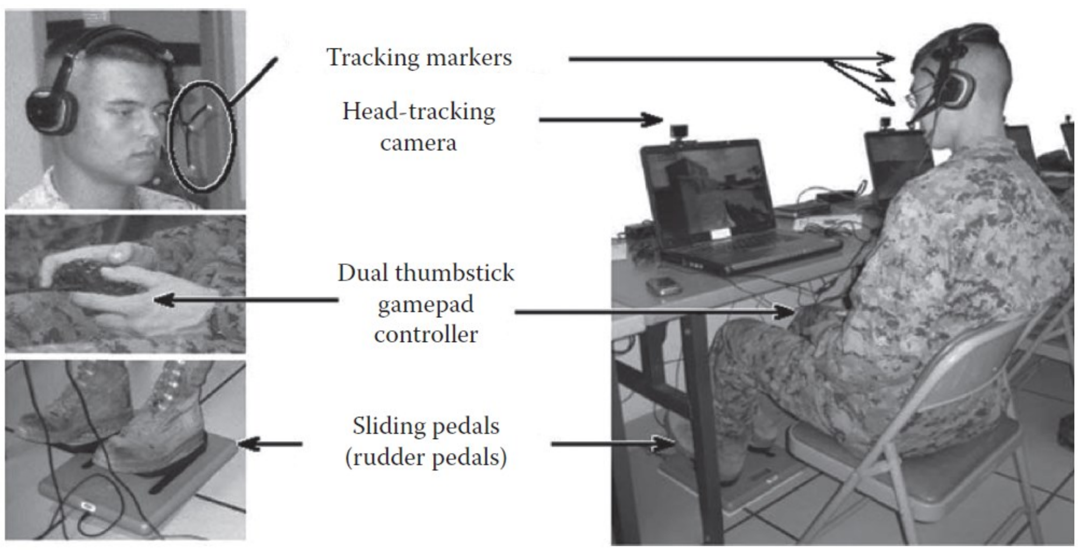

Figure 1: Pointman Input Devices (from Templeman, 2014)

3.3.2 Realistic Dismounted Infantry

3.3.2.1 Realistic Avatar Look

"The UTE shall provide the capability to represent players (real and bots) with realistic clothing and equipment for the role they represent. Entities in the game will be able to be configured to represent realistic ethnicities, genders, occupations, head/facial hair, etc."

Both Engage and VBS2/3 have a good variety of military and civilian characters represented as assets. Engage utilizes the DI-Guy character and animation system, a product of VT-MAK (https://www.mak.com/products/visualize/di-guy), to represent lifeforms while VBS2/3 has a native capability. While both systems offer the capability to create new lifeforms it is an in-depth process and has limitations on physical size of characters and animations.

Character representation and animation is a hallmark feature of all game engines and as such they offer extremely complete toolsets for both character creation and animation as well as supporting a wide variety of open source formats allowing access to a rich after-market asset pool. In both VBS2/3 and Engage, however, character asset creation is still an in-depth process likely requiring expert users. Additionally, utilizing assets in any derived software would require extensive programming and integration efforts.

3.3.2.2 Generate Recognizable Avatar Look and Gestures

"Realistic recognition and interaction between real players in a game is a challenge. The UTE shall be capable of skinning the face of an avatar with the likeness of the real player to aid in player recognition. The UTE shall also enable realistic gestures and motions within the game to enable non-verbal communication between real players."

The VBS2/3 system offers an avatar facial skinning system as part of the package (Figure 2). This software tool allows photographs of individuals to be adapted to 'players' in the system and will be loaded accordingly to the user logged in. It should be noted that the physical size and shape of the avatar are not altered, only the textures used for the face/head.

Figure 2: VBS3 Facial Skinning

Engage does not offer any integral tools for facial mapping of avatars.

All the game engines including Unigine offer complete custom character creation using third party 3D modelling tools. As with the case of urban terrain however, custom character creation would involve a significant amount of expert support.

While several of the software products can invoke hand gestures in the game through keyboard key combinations, custom gesturing is a feature that offers some challenges for all the discussed systems. The first challenge is tracking hands and fingers with enough fidelity to identify gestures. This will be especially challenging given the very likely scenario that the user will be working with physical equipment such as a weapon. This limitation would very likely limit any attempt at gesture tracking to a glove system. To complicate this situation most glove tracking systems only provide finger information but to not track the location of the hands to any frame of reference.

The rapid increase in virtual reality systems has however introduced innovation in tracking technologies. As such, there are several new technologies and products entering the market that could be candidates for gesture tracking solutions.

All the software technologies presented offer the ability to control player avatar appearance at the character skeleton level required to display gestures. For all the technologies, this would represent highly specialized software development to implement.

CASE STUDY EXAMPLE: VBS2/3 Real Time Character Tracking

Real time character tracking has been a simulation design goal for some years now and has overcome some but not all challenges. The US Army Pointman demonstrator is a good example of this. Because of some very extensive development work on this very topic using VBS2, the following information serve as a bit of a summary/lessons learned. The first item to discuss is the accessibility of character control within VBS2/3. This is handled via either the macro torso handlers or the full character skeleton system. Although not tested in VBS3, access via the full character skeleton system in VBS2 was extremely unpredictable as neither the bind pose information nor the skeletal traversal order was known. Additionally, any manipulation of the skeletal system seemed to fight against the software's internal kinematics and/or animation system, although this is only conjecture based on observation. The macro torso controls however did function quite well and can be used for most character manipulation.

The macro controller system allows for three main inputs: weapon controller, upper body controller, lower body controller and head controller. These controllers along with scripting commands work in harmony with the internal animation system and allow very good character representation. It should be noted that as of VBS2 version 2.8 that custom character reflection across the network was only partially implemented. To implement such a system, the following needs to be tracked in real time: head, chest and weapon. Kneeling and prone conditions can be derived in software and the appropriate controllers manipulated. It should also be noted that very fine tracking control is required for accurate weapons aiming. This type of real time input has been created using three wireless Altitude Heading Reference System (AHRS) devices and could be demonstrated if required. Other tracking systems have been tested to varying degrees of success. The Microsoft Kinect system has proven to be unreliable and unstable in tracking unless used with many constraints. Optical tracking systems offer good tracking however are quite expensive and require a large physical footprint. Our development efforts have shown that micro AHRS devices provide the best solution.

The last element of real time tracking is locomotion. If we assume that each participant will have a physically restricted area in which to move, some alternative form of walking/running, etc. will need to be employed. As seen in the Pointman demonstrator a foot pedal system was used for "walking in place" Unfortunately this does not lend itself well to turning while walking/running/crawling. Alternatively, miniature joysticks have been mounted to weapons as a form of locomotion input and work reasonably well however does not impose any physical strain to the participant which can be a pro or con depending on the desired effect. In the previously mentioned AHRS implementation we experimented with two additional devices attached to the feet and using foot orientation and acceleration could implement locomotion for walking, running and crawling in place. Combined with torso orientation this allows direction to be controlled as well. Although this system was not fully developed, it showed good promise and imposed physical stress to the participant which was important to the DND research sponsor.

Lastly the weapon needs to be tracked. Like the body this can be accomplished using any of the different technical methods previously discussed, however as mentioned if the weapon is to be used for aiming then the tracking needs to be of very high quality. In the previous research development mentioned above we implemented a micro AHRS tracked weapon system and the tracking was fine enough to observe the participants' breathing on the aim reticule. One other consideration in this type of implementation is the ability to sense when the weapon is 'at the ready' or when it is shouldered and ready to fire as VBS2/3 offers both shouldered and non-shouldered firing depictions. The weapon we developed for this used internal capacitive sensors in the fore stock, pistol grip (AR-15 weapon) and butt plate to allow detection of weapon firing position. The custom electronics also tracked shots (semi or full auto) and detected magazine changes. The weapon did not simulate stoppages or misfires.

Wireless micro AHRS systems used in this development cost ~$300 per unit. The specialized AR-15 weapon based on a high-end Airsoft system cost ~$5000 per unit. For the custom weapon, all sensors and electronics were internalized to the weapon and connected wirelessly to the host computer. The figure below depicts the final customized weapon (top) next to an actual AR-15 (bottom).

Figure 3: Instrumented AR-15 (CogSim)

The final technical detail worth mentioning is that the different tracking systems need to operate in different frames of reference either by default or by software. As an example, torso tracking needs to be in a frame of reference that can be tied to orientation (north) to orient the character in software. The head and weapon orientations will then need to be measured or converted to the characters' frame of reference.

3.3.2.3 Realistic Bot Behaviors and Actions

"The UTE shall enable the experimenter to assign behavioral responses and actions to bots in the game. For example, upon encountering blue-force member's bots could be programmed to behave as a hostile (e.g. from avoidance to aggression to engagement with fire), or as a neutral (e.g. passive, observant, but non-interactive), or as a friendly (e.g. from welcoming to supportive)."

This functionality is currently implemented in VBS2/3 and Engage. General responses can be influenced or controlled by setting conditions that influence the internal AI of the system. As an example, how two entities respond to encountering each other is dependent on their designations as enemy, friendly or neutral combined with general military conditions such as "weapons free" and "weapons hold". Below this level of initial conditioned response, unique behaviors and responses would need to be controlled via the scripting interface within VBS2/3. Scripts allow a customized response to given input criteria and can be enabled or disabled by linking that script (or not) to a given scenario. It should be noted however that in the absence of scripting logic the system will default to its base AI. This functionality dictates that critical responses to developed scenarios will need to be thoroughly tested and validated to isolate and/or eliminate unwanted behaviors.

3.3.2.4 Urban Terrain Simulation Suitable to Support Team Experimentation

"Effective team experimentation requires sufficient terrain complexity to create informational challenges in positional location and situational awareness among elements in a larger team.

Urban terrain shall be represented by multi-story buildings, roads, alleys, and open areas bounded by buildings. Terrain shall be able to be populated with context-specific items to add dimension to the realism of the spaces (e.g. vehicles, trees, fuel drums, market stalls)."

All the software solutions previously mentioned allow for the creation and/or modification of highly detailed urban environments. It should be noted that not simply the 3D representation of buildings and other objects is required, but also the 'physics' representation so that objects can move and react with other objects in a natural way. Additionally, the ability to climb stairs and ladders as well as blocking explosion and bullets should be handled by the software. VBS2/3 offers this functionality natively. All the software solutions offer the ability to model physics for terrain and object creation however it will require custom development.

It is also highly desirable that the urban terrain allow for object damage and destruction. Both Engage and VBS2/3 have basic native capability for this and are likely sufficient for normal operations. However, should the destruction of buildings and objects be prominent in the experiment, the game engines will be much more capable of a more sophisticated solution at the cost of more development effort.

3.3.2.5 Representative Night Vision Capability

"For low-light and night operations, players equipped with night vision devices (e.g. NVGs, weapon optics) shall be provided with a simulated night vision view of the world. Ideally, this night vision view will be capable of simulating both image intensification and infra-red devices."

Integral sensor representations are available from VBS2/3, Engage and Unigine. All three of these software systems offer realistic materials-based implementations of sensor views. The remaining game engine software all have the capability to create NVG and/or thermal views through software development.

3.3.2.6 Represent Remote Controlled Devices

"Remote devices are becoming more common on the modern battlefield (e.g. drones, UGV, remote cameras). The UTE shall be capable of representing these remote devices and their functionality, and be capable of being programmed to move through a pre-planned route, operated by a player in the game, or operated by an operator outside of the game play. Imagery from these devices shall be able to be remotely viewed by players in the game."

The ability to create and exercise remote vehicles is inherent in both VBS2/3 and Engage. All the other software engines are capable of this as well however they would require significant software prototyping and development.

Remote viewing of imagery is not integral to any of the software under consideration. To achieve this capability, technology would need to be adopted to handle video compression, streaming and decompression over a network. It should be noted that this very technology, along with embedded Key-Length-Value (KLV) data was integrated for the DRDC-T TIGER UAV simulator and could provide a good starting point for any development effort. The software tool sets used for this work was purchased from ImpleoTV and has been used internationally by several large defense industry companies.

3.3.2.7 Integrate Real Player Consequence System

"To encourage more realistic shot avoidance behavior, the UTE shall be capable of triggering a shot consequence system (e.g. shock belt), worn by the person whose avatar is shot in the game."

Shot consequence systems have existed for some time now however they have not been widely utilized. These devices have not been widely adapted due to complex APIs and bulky physical equipment. Additionally, there has been an ethical/legal trade-off between the amount of discomfort these systems impose on the user compared to the perceived increase in realism. As an example, if the intent is to simply let the user know that they have been hit or damaged there are visual, audio and other means to convey this information. On the other hand, if the intent is to increase caution in the user by introducing physical discomfort as a form of consequence, then there arise moral and ethical issues. In the military training environment, these thresholds may be much higher but this issue has very much limited the commercial development of such devices.

On a positive note, with the rapid expansion of the new virtual reality game market there are several new start-up companies looking again at haptic feedback devices. Even though there may be new commercial product offerings soon they will likely be constrained in the physical impact to the user for the above-mentioned reasons.

One should also note that some thought would need to be given to any implementation of this type of system to understand how the software architecture determines results of events. Given the standard military distributed training scenario it needs to be well understood how the event-consequence evaluation system works. A good example is the scenario where one player fires a shot at another player. In this example the shot taken is represented over the network as an information packet containing the munition used, the ID of the shooter and usually the ID of the intended target along with shot vector with muzzle velocity magnitude. The firing party however does not normally decide whether the intended target was hit. This logic is normally handled by the target and includes determination of the damage done should a hit occur.

3.3.3 Real-Time Data Capture and Analysis

3.3.3.1 Support Platoon Sized Missions

"The UTE shall support dismounted infantry operations up to and including Platoon-sized operations."

Both VBS2/3 and Engage are capable of handling platoon and larger sized missions. As individuals within the larger formation are normally represented by multiple computers and software, scaling is not an issue.

Game engine technology has a virtually unlimited capacity to handle large groups of players as well given sound software development.

3.3.3.2 Update Information to Participants in Play

"As location and situation awareness information changes during the game, the UTE will update information displays to players in real time during game play."

For any software application that contains multiple similar or dissimilar player representations, care and planning must be given to the supporting network communications infrastructure and configuration. Local groups can be networked trivially however geographically separated setups will provide configuration challenges depending on connectivity between sites. Any reliance on the internet for connectivity will need careful planning considerations for both the low-level information exchange as technical items such as User Datagram Protocol (UDP) not being supported will need to be overcome. Additionally, without careful design architecture performance of the network will very likely be severely impacted.

3.3.3.3 Capture all Communications with Location and Time

"The UTE shall capture real-time communications between blue-force players during game play. Voice communications will be recorded and stamped for time/date, player, role, and network."

Any voice communications that are radio based and handled through Distributed Interactive Simulation (DIS) or High-Level Architecture (HLA) can easily be logged and recorded. Date/time, player and network information can be collected from network traffic and correlated as well. There are any number of 3rd party tools capable of this functionality. Both VBS2 and VBS3 include a native capability to log and playback voice communications via DIS.

Non-radio voice communications are not normally captured and recorded in any of the software solutions. Straight voice over IP (VOIP) communications functionality is native to game engines and

Unigine; however recording and playback capability would need to be developed. Neither VBS2 or Engage offer VOIP natively however developing this functionality natively is not a difficult task. VBS3 does seem to offer this capability natively although it has not been tested and confirmed.

VBS3 has removed dependence on CNR-Radio and incorporated a new internal radio system. VBS Radio now offers user-friendly setup of global, side, group and direct speech communication channels. The standard VBS Radio version, included with VBS3 for free, features five channels, 3D positional audio, automatic access to vehicle intercom systems and squad radio nets, and an audio tone to signal when a transmission begins and ends. The VBS Radio Pro version is available for purchase as an add-on to VBS3 and offers unlimited channels and allows configuration of radio network pre-sets, enabling users to customize channel names, colors, icons and enable saving and loading of networks per mission. VBS Radio Pro also features distance-based degradation that can also be influenced by weather effects in the simulation and provides support for side tone playback.

3.3.4 Experimentation Configuration Dashboard

3.3.4.1 Toggle Experimentation States and Conditions

"The UTE shall enable the experimenter to vary game functionality and environmental conditions. Functional conditions could include error variability and data update delays (e.g. position location information displayed to players in real time). Environmental conditions could include time of day, precipitation, visual obscuration, etc."

For any non-local network design as discussed in the previous section, careful design and configuration of the supporting network will be required to minimize data update delays. It should also be noted that network stress-testing should include voice and radio communications as they can impose a significant increase in network data volume. It has been experienced on many occasions with military distributed mission training that the network grinds to a halt upon first contact when radio communications spike.

To intentionally implement an operator controlled delay or error in information reporting custom software would need to be developed in all cases. It should be noted that information transfers internal to the software cannot and should not be altered as this data flow represents the "real time" element of the system. For devices either internal or external to the system for which a delay or error is required to be introduced the following high-level methodology will need to be implemented. The true data will need to be polled from the system network via software and have the error or delay introduced. The equipment that is being simulated will need to accept only this external data source as its input. For normal operation of the equipment there would be no data manipulation in the external loop.

The following information pertains to environmental conditions that can be set either per scenario or dynamically during execution. The first is time of day and is available in both VBS2/3 and Engage. The rendering system of both software packages allow for proper representation of time of day with respect to sun and shadow positioning as well as direct and ambient light including night.

In VBS2/3 the weather conditions are also selectable as follows:

Table 3: VBS2/3 Environment Control Options

Overcast	Selectable between 0 and 100%. More than 70% cloud cover brings a chance of rain, increases wind conditions, and may affect sea conditions.
Wanted Overcast	Selectable between 0 and 100%. The scenario changes the cloud cover to this setting over a period of about 30 minutes.
Current Fog	Selectable to set the fog density at the Base altitude.
Fog Base / Ceiling	Selectable altitude for the base fog density and ceiling limit for the fog. Fog extends from ground level to the ceiling limit where the fog dissipates. The density of the fog increases exponentially from the ceiling to the ground. Increasing the base altitude with the same ceiling increases the density of the fog beneath the base altitude.
Wanted Fog	Selectable to set a desired fog density at the base altitude. The scenario changes the fog density to this setting over a period of about 30 minutes.
Wanted Fog Base / Ceiling	Set desired base and ceiling limits for the fog. The scenario changes the base and ceiling to these settings over a period of about 30 minutes.
Rain	Selectable amount of rainfall, increasing rain impacts the Overcast level.
Surface Moisture	Select to increase or decrease the amount of water on the ground. This setting primarily affects mud surfaces with an impact on vehicle performance.
Snow	Select to increase or decrease the amount of snowfall. Snow functions correctly only on maps created using tools from VBS3 3.5 or later.
Environment Modify	Select a setting to apply a ground layer of snow in the scenario.
Wind Speed	Selects the general wind speed in m/s. Gusts of various speeds may still exist. Set wind speed to "Dynamic" to use variable wind speeds and directions during the scenario.
Wind Blowing From	Select a direction that the winds blow from.

3.3.4.2 Assignable Capabilities to Individual Players

"The UTE shall provide the experimenter with the capability to vary the functional ISS capability of each player and be able to change a player's role and functionality during game play."

This capability is resident within VBS2/3 and is somewhat resident within Engage. Utilization of game engine software is certainly capable of implementing this capability however as is the case once again this will require a software development effort.

"Radio nets shall be able to be assigned to participants according to an ALCON net, command net, Pl net, Section net, and Assault Group net."

Simulated radio nets are normally their own subsystem of any training software. All but VBS2/3 would require a separate network of radio emulation software and as such can be set up as desired to represent any given communications plan. The software radio capabilities of VBS2/3 can also be similarly set up however the overall number of radio networks that can be supported simultaneously may be limited by the standard versus pro implementations in software. Note that this does not limit the number of simulated radio networks that can be utilized but only the number of radio representations a given player can have assigned. Also, a radio set to transmit and receive on a given frequency but monitor another frequency would constitute two radios.

3.3.4.3 Manage Real and Bot Player Roles and Assignments

"The UTE shall enable the experimenter to assign roles and functionality to both real and bot players. Real and bot players shall be able to be assigned to organizational groupings in both blue, red, other combatant, and non-combatant units."

This functionality is currently implemented in VBS2/3. With respect to AI players, VBS2/3 has an internal logic set based on enemy/friendly/neutral interactions as well as organization group logic. As an example, members of a squad/platoon will automatically follow the leader and act/react to situations. VBS2/3 also allows the modification of all action/reactions via scripting. It should be noted however that in the absence of scripting logic the system will default to its base AI. This functionality dictates that critical responses to developed scenarios will need to be thoroughly tested and validated to isolate and/or eliminate unwanted behaviors.

All other potential software implementations can implement organizational groups and behaviors however would require this to be fully developed and implemented in the code.

3.3.5 Other Considerations

DRDC-T has been using Bohemia Interactive's VBS software product in their soldier team experimentation for many years. In discussions with DRDC-T engineers it was apparent that there were important, pragmatic issues to consider regarding the choice of software platform for the UTE. A key factor was the time and effort necessary to learn and adapt any new software product to be able to support the needs of specific experiments. It was estimated that a full 2 years of significant effort were required to learn the current software code, parameter setting, and custom programming to be able to develop the necessary terrain, bot behaviours and actions, software tools, and scripting requirements for the unique experiments they support.

Most of the experiments that are run at DRDC-T involve a significant amount of custom programming and development work by the engineering staff. Over time they have developed a sizable library of VBS coding techniques, work-arounds, data capture coding, compiled scenarios, and insight into getting the most experimentation value out of the VBS software. Despite the unique nature of the experiments they support there is some scope for re-using past coding work or re-purposing it with coding modifications, offering some efficiencies when preparing for future VBS experiments. As well, the VBS product is currently employed as a simulation training product in the Army's Infantry School so soldier participant in DRDC experiments may already be familiar with the software and common software platform increases opportunities for collaboration between both organizations. In their opinion, there has been a significant amount of technical inertia built up with the VBS product to date which needs to be considered in the cost/benefit decision of any change to a different software platform.

Future development opportunities suggested by DRDC-T engineers included the development of a data parsing capability to expedite the conversion of VBS data to results; a longer-range outlook with planned future experiments to identify opportunities for developing some common modules to improve efficiency and re-use of developed code across multiple experiments; and a universal tablet solution as a common secondary display option to simulate other information products.

3.4 Scenario Development Framework

The UTE would also benefit from a framework for developing scenarios for team experimentation in dismounted infantry operations.

The following section describes considerations, guidelines, and criteria for the development of scenarios for use in team experimentation in the UTE laboratory. The first sub-section details the scenario contexts, or Use Cases, across which the scenarios may be implemented in the UTE lab and the considerations of force structure to maximize the experimental outcomes for given lab resources. The second sub-section elaborates guidelines in the development of scenarios using lessons learned from prior scenario development efforts, Use Case considerations, and development strategies. The third sub-section explores definitions and measures of scenario complexity, towards the goal of controlling and intentionally manipulating complexity across multiple scenarios and different experimental Use Cases in a standardized and repeatable manner.

3.4.1 Scenario Use Cases

Three scenario Use Cases were considered for the UTE lab, reflecting guidance of the Scientific Authority (SA) and other stakeholders. In developing the Use Cases, the ratio of participant players to supporting confederate and computerized (bot) players was maximized. Participant players are insulated from bot players where possible, such that the need for and complexity of bot scripting is minimized while confederates can react appropriately to unanticipated participant behaviour. While the needs of specific experimentation may vary, the following figures illustrate the generalized relative mix of participant players to confederate players to bot players. A limited number of additional players representing the opponent force (OPFOR) and experiment control (EXCON) would also be required. EXCON would be responsible for scenario injects (e.g. weather, external events), controlling civilian behaviour, and blue (coalition) and green (local) force interactions.

The most basic Use Case of team experimentation considers individual soldiers within a section context. In Figure 4, an eight-soldier section would likely require five confederates and 12 bots to situate the soldiers in a Section and Platoon context. These five confederates include the Pl Commander, two Section Commanders, and two Section Second-In-Command (2ic) soldiers. Confederates are used at the command levels of the platoon and other two sections to ensure realistic behaviours in these teams. Bots will follow their confederate Section Commanders and 2ICs and make up the numbers in the platoon. Six bots would be included in each confederate Section to fill out the numbers to a total of eight members per Section. For the soldiers in a section context, the number of OPFOR would be strictly limited by the section capability.

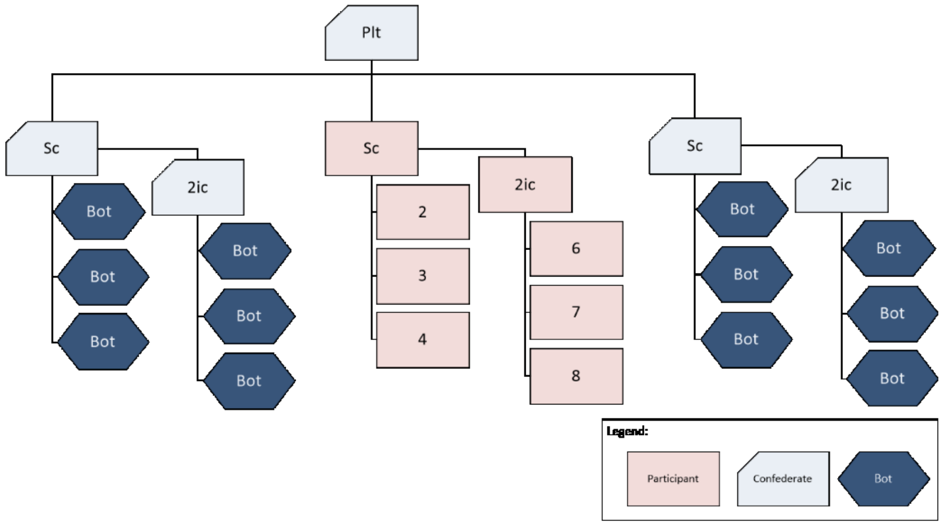

Figure 4: Soldiers in a Section

The second Use Case considered were Section Commanders and Assault Group leaders in a platoon context. In this Use Case shown in Figure 5, the six participant players would likely require the support of eight confederate players (FT or Fire Team leaders, a Platoon Commander and Warrant Officer) and 13 bot players to act as fire team soldiers. These number are approximations based on hypothetical experimental goals, while the goals of specific experimentation would drive the number of confederate and bot players required; however, these generalized numbers are useful approximations for the development of laboratory capabilities and understanding experimentation time and resource drivers. Again, OPFOR and EXCON would be additional beyond the force structure show here.

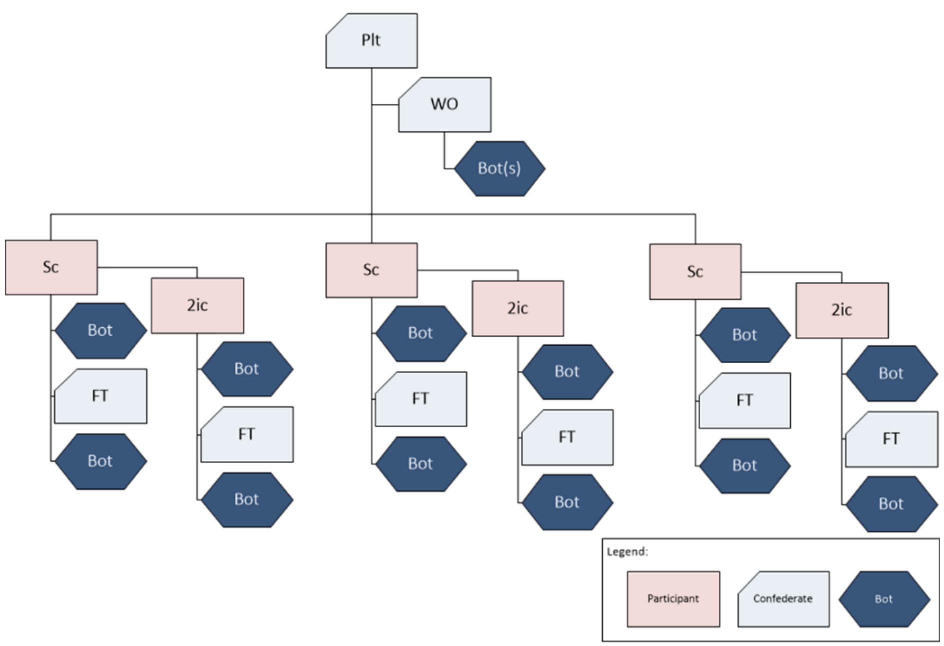

Figure 5: Section Commanders & Assault Group Leaders in a Platoon

The final Use Case considered for the UTE lab is Platoon Commanders in a company context. In this situation, three participant players would require the support of 13 confederate players (Company Commander, 3x Platoon Warrant Officers, and 9x Section Commanders) and 12 bot players to act as soldiers in each platoon – see Figure 6. While efficiencies may be possible with confederate players filling multiple roles and bot players not being fully simulated, the increased requirement for supporting players in this Use Case is clearly illustrated. While the scale of the scenario would be much larger at the platoon commander in a company context, EXCON and OPFOR requirements would not increase proportionally.

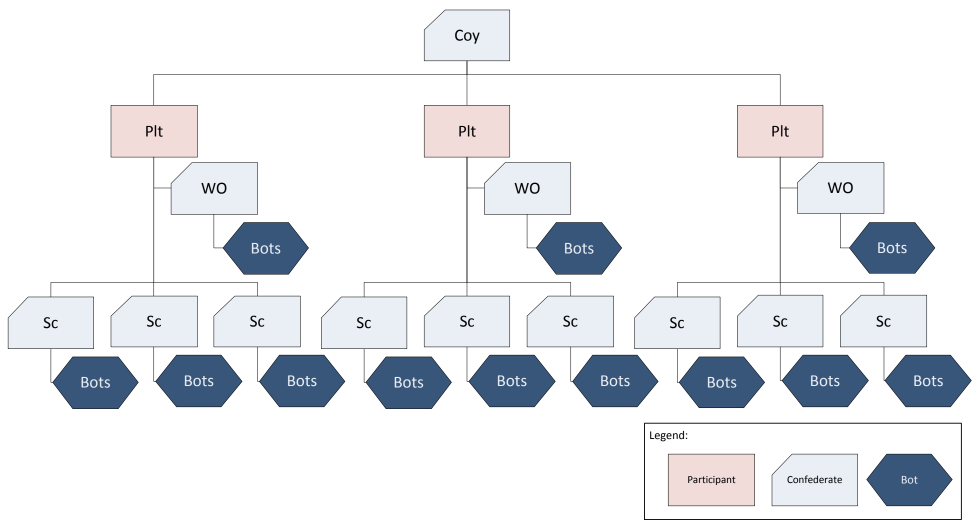

Figure 6: Platoon Commanders in a Company

3.4.2 Scenario Design

3.4.2.1 Trade Space

Through the definition of scenario Use Cases and analysis of experimentation scenario requirements, trade-offs in scenario requirements become apparent. These trade-offs are mapped to a trade space in Figure 7, with the three scenario contexts on the x-axis in the space and a generic low, medium, and high ascending the y-axis space. While the total number of players is relatively constant across Use Case contexts, other considerations show ascending or descending requirements. In general, as the level of experimentation moves from the soldier to the Section Commander to the Platoon Commander the number of participant players feasible to test at one time will decrease; whereas at the same time the number of confederate players and enemy bots will increase. There are higher terrain realism and civilian activity requirements at the smaller unit size decreased to the soldier level, reflecting the immediacy of the simulation to the individual soldier and the increasing level of abstraction and distance with higher levels of command. On the other hand, terrain scale required for individual soldier level simulation is much less than for the simulation for higher levels of command.

This trade space illustrates the resource requirements, including development time and effort, of experimentation across the different levels of scenario Use Cases. For example, in general testing with soldiers in a section will necessitate greater terrain realism; however, efficiencies will be realized in greater throughput of participant players with fewer confederate requirements. Conversely, while less terrain realism is required when testing platoon commanders, more enemy bots and confederates will be required in order to collect data from fewer participants. The trade space is a useful depiction of the considerations in developing the UTE lab capabilities and designing scenarios for different levels of team experimentation.

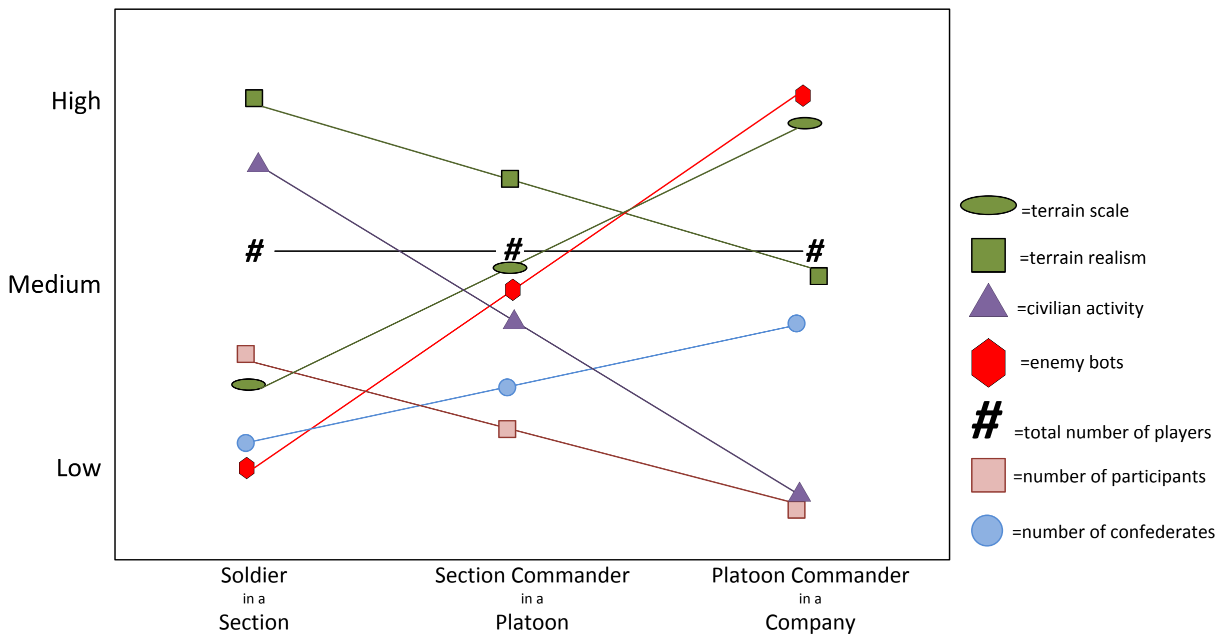

Figure 7: Trade Space of Scenario Considerations

3.4.3 Strategies

From the analysis of team scenario needs and a review of scenarios from prior studies, general strategies in scenario development are described. Forty-eight small unit (section level) scenarios were generated for team experimentation using the Virtual Battlespace Simulation platform (Tack, Bruyn Martin, Palmer, & Elderhorst, 2012). However, detailed inspection of these scenarios revealed four core missions, varied in instantiation to form 12 vignettes which were then iterated across four terrain models to give 48 scenarios. The scenario developers varied six key parameters to create 12 unique vignettes. These parameters manipulated the number, locations, and behaviour of blue force, enemy force, criminals, coalition forces, civilians, and an NGO organization. Combining these 12 vignettes with four different terrain models, each with different engagement distances, building density, and building characteristics results in 48 unique scenarios. This parameter combination strategy for scenario design is illustrated in Figure 8.

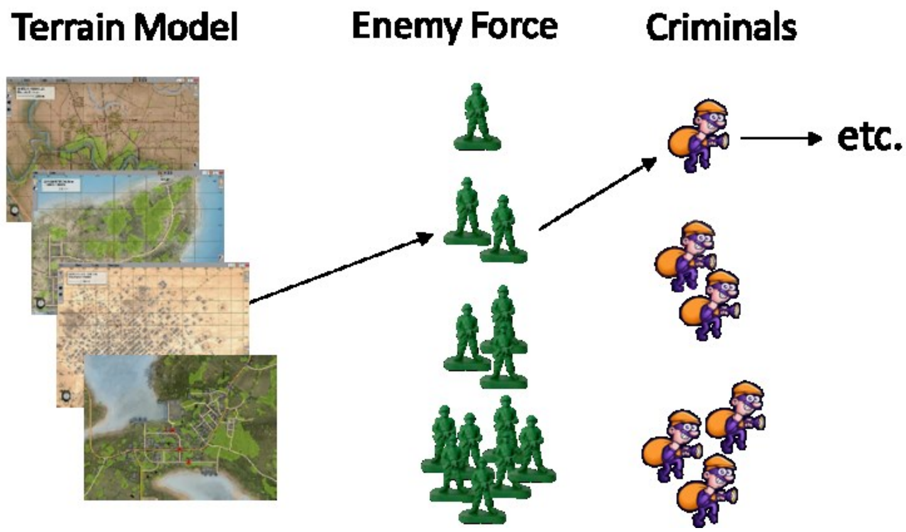

Figure 8: Parameter Combination Scenario Design Strategy

Another recent scenario development project took a different strategy to scenario design (Tack and Nakaza, 2016). A single broad scenario context was first developed and this scenario context was reused through a series of vignettes exercising different components of the overall scenario. This is depicted visually in Figure 9. In this way, the individual vignettes were nested within a larger scenario context, with the context of adjacent units and higher units re-used in multiple vignettes. By reusing the overall scenario context, efficiencies in development of individual vignettes could be realized. This project used the attack of a platoon house as the overall scenario context, with different phases of the attack and types of attack as different vignettes for experimentation. Individual vignettes can exercise the same unit size, thereby providing multiple tests for a single Use Case (e.g. soldiers in a section), or across different unit sizes. This type of approach could have advantages in a repeated measures experimental design by providing continuity in narrative as the participant is tested across multiple conditions.

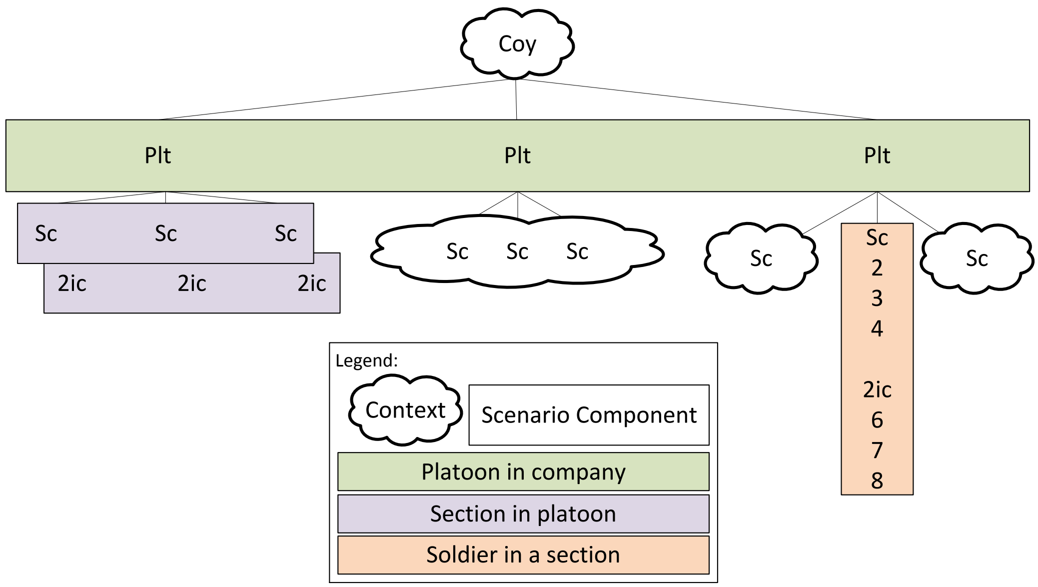

Figure 9: Nested Scenario Design Strategy

3.4.4 Scenario Complexity

3.4.4.1 Prior Scoring System

The scenarios generated for the prior study also included a coarse scoring mechanism (Tack et al., 2012). Relative ratings of the scenarios in six key parameters (e.g. civilians, enemy force, NGO, etc.) were conducted by the scenario developers in terms of number, locations, and behaviour on a scale from 0 indicating not applicable to 3 indicating the most complex of the parameter options (e.g. very crowded with civilian activity, enemy in multiple advantageous locations, NGO interacting with civilian population). Additional geographic complexity ratings were based on opening engagement range, building density, and building heights. The total complexity score of a vignette was the summation of all parameter ratings and the geographic complexity ratings. This score provided a rough index of the relative complexity in comparing the vignettes generated through the parameter combination strategy of scenario design.

3.4.4.2 Theoretical Perspectives

The scientific literature of training has developed perspectives on scenario complexity that could be useful in the current application. An optimal progression of challenges, or content difficulty, with increasing skill of the trainee has been proposed as a Zone of Proximal Development (ZPD) or Flow Channel (Ferdinandus, 2012) – see Figure 10. This ZPD necessitates quantification of difficulty to enable calibration to the trainee skill level. This has resulted in theoretical perspectives on the quantification of difficulty, or complexity, of a training task. Note that complexity is the preferred term, as opposed to difficulty which is viewed as a subjective assessment that may not hold across individuals (Martin, 2012).

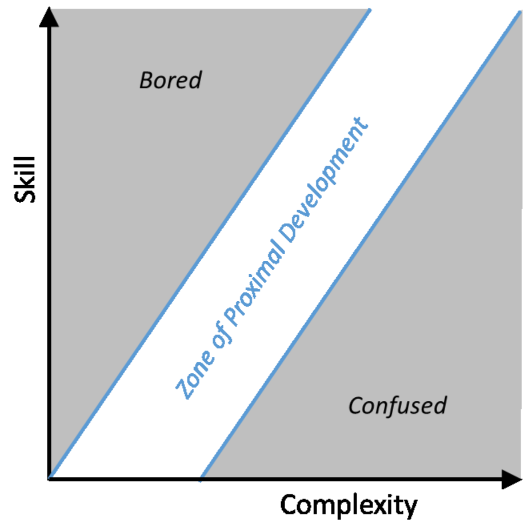

Figure 10: Zone of Proximal Development (Ferdinandus, 2012)

Tasks are said to contain three essential components as building blocks of complexity: products, acts, and information cues (Wood, 1986). Products are the observable outcome of behaviours, with a set of attributes, in the form of an object or event. Acts are the behaviours required to produce a desired product, with both direction and context as attributes. Information cues are the pieces of processed data which form the stimulus of the task. Using this general theory of tasks, a model of task complexity is described by Wood (1986).

Component complexity is a simple overall scalar metric of the number of task components involved in a task (Wood, 1986). Component complexity is indexed by the summation of distinct acts possible, information cues considered, and subtask products required. In this way scenarios of larger scale are considered more complex. Automaticity and redundancy may substantially reduce the component complexity level of a scenario, in both cognitive and psychomotor tasks.

Coordinative complexity accounts for the form, strength, and sequencing of relationships between task inputs and task products of a scenario (Wood, 1986). Interdependencies in timing, frequency, intensity, and location requirements of acts, information cues, and subtask products will increase the coordinative complexity of a scenario. The summation of the number of precedence relations between acts and information cues has been suggested as a simple metric of coordinative complexity.

Dynamic complexity accounts for the changes to the relationships between task inputs and products (Wood, 1986). As the situation changes, the cause-effect chain may be altered thereby shifting task requirements and forcing adaptation of previous strategies and actions. Dynamic complexity can be thought of as component and coordinative complexity over time. If component and coordinative complexity are constant over the course of the scenario, dynamic complexity will be nil; however, changes in the component or coordinative complexity are themselves complexity and are capture in dynamic complexity.

Total task complexity is then implied through the weighted summation of component, coordinative, and dynamic complexity (Wood, 1986).

An alternative perspective of complexity as objective task characteristics that increase information load, information diversity or rate of information change was posited by Campbell (1988). Four characteristics were proposed as (1) multiple potential pathways to the desired end-state, (2) multiple desired outcomes, (3) conflicting interdependencies, and (4) uncertainty or probabilistic links between pathways and end-states. Multiple potential pathways to a desired end state can serve to decrease complexity in allowing for many possible means to a solution; however, if only one pathway results in the desired end-state or if efficiency is a criterion then complexity will be increased. Complexity is increased with multiple desired outcomes as more goals must be satisfied, although this effect is moderated when the outcomes are positively related. Conversely conflicting interdependencies, or negative relationships, between end-states will require trade-offs and optimization among desired outcomes and thereby increase complexity. Finally, if the connection between pathways and end-states are uncertain or probabilistic, complexity will be greater. Total complexity is then determined by the number of complexity characteristics present and the degree to which the characteristics influence complexity.

A typology of complexity is created with all possible combinations of complexity characteristics as a coarse relative ordering of total task complexity (Campbell, 1988). This typology of task complexity and task type heading labels are shown in Table 4. Simple tasks show none of the sources of complexity. Decision tasks emphasize the choice or discovery of an optimal outcome to achieve multiple end-states. Judgment tasks emphasize deciphering conflicting interdependence and probabilistic linkages. Problem tasks emphasize multiple pathways to the desired outcome, often with optimization criteria. Fuzzy tasks are distinguished by the presence of both multiple desired end-states and multiple pathways to achieving them.

Table 4: Task Typology & Classifications (Campbell, 1988)

Task Index	Sources of Complexity				Task Type
	Multiple Pathways	Multiple Desired End-states	Conflicting Interdependence	Probabilistic Linkages	
1	-	-	-	-	Simple
2	X	-	-	-	Problem
3	-	X	-	-	Decision
4	-	-	X	-	Judgment
5	-	-	-	X	Judgment
6	-	-	X	X	Judgment
7	-	X	-	X	Decision
8	X	-	X	-	Problem
9	-	X	X	-	Decision
10	X	-	-	X	Problem
11	X	X	-	-	Fuzzy
12	-	X	X	X	Decision
13	X	-	X	X	Problem
14	X	X	-	X	Fuzzy
15	X	X	X	-	Fuzzy
16	X	X	X	X	Fuzzy

Scenario complexity is then the combination of task complexity, measured in terms of component, coordinative and dynamic complexity, and task structure, as described through the task typology (Martin, 2012). Another interpretation is that scenario complexity is the product of task complexity and cognitive context moderators plus the task framework structure (Dunne, Schatz, Fiore, Nicholson, & Fowlkes, 2010). Cognitive context moderators are external stimuli that increase the cognitive load of the trainee or reduce the resources available for the task.

Another project conducted for DRDC examined the underlying scenario structure to determine the elements that contributed to difficulty in a collaborative problem-solving research platform (Morton & Adams, 2011). Structural elements that were found to increase complexity included size and scale of the scenario, interaction and interdependence of scenario components, non-exclusive solutions, ambiguity of information cues, as well as level and coherence of distractor noise.

3.4.5 Consolidated Perspective

Consolidating the past scoring systems and theoretical perspectives of scenario complexity, a model of scenario complexity is presented in Figure 11. This model reflects the perspectives of Wood (1986), Campbell (1988), Dunne et al (2010), and Martin (2012) while also being consistent with the findings of Morton and Adams (2011). Scenario complexity is defined by components of individual task complexity, task structure (or framework), and context moderators. Simple ratings of parameters within any of the components of the model could enable comparisons of the relative complexity between analogous scenarios developed through a parameter combination or nested design strategy. For example, component complexity would be influenced by the enemy force size and the presence of civilian population would increment the context moderators. Past scoring systems have allowed for relative ratings of complexity between similar scenarios developed for the same project but have not been validated. Methods of summating the relative contributions of the different parts of the complexity model to total scenario complexity would need to be developed and validated.

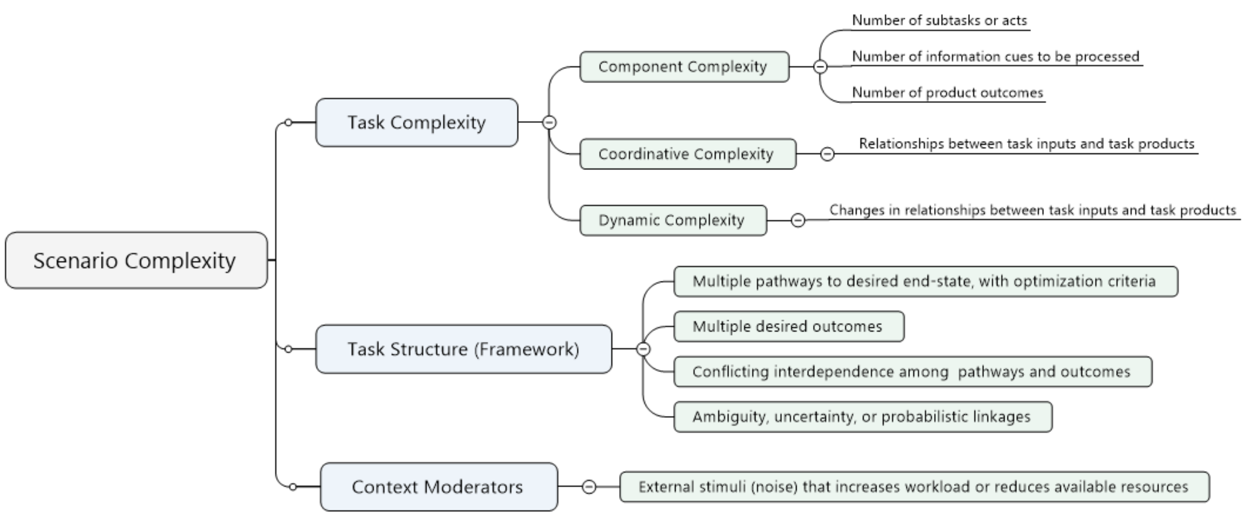

Figure 11: Scenario Complexity Model

4. Discussion

The results of this investigation are discussed below for both UTE software and hardware, and the scenario framework, including recommendations for the way forward.

4.1 UTE Software and Hardware

Based on the requirements developed with key scientific and engineering stakeholders at DRDC-T, candidate hardware and software options were reviewed and assessed to determine the best options for fulfilling these requirements, in the context of infantry team experimentation.

Considering all the UTE requirements together it is evident that no one software solution will meet all aspects of every identified requirement. However, a large majority of the core requirements can be achieved by continuing to use the VBS2/3 software. While many of the requirements can be achieved using VBS2/3, some requirements will necessitate coordination with expert development, especially in cases of interfacing to specialized hardware and third-party software. The current level of organizational and technical engineering investment made to date in VBS2 and the extensive time and effort necessary to become proficient with any new software product argues against a change from VBS2/3 unless revolutionary benefits were possible with any new software. Our analysis suggests that the other software products reviewed do not come close to this threshold requirement.

In the case of very specialized areas of interest, however, VBS2/3 has significant limitations. In those instances, game engine software, which allows for a much deeper level of interfacing into the functionality of the software, will be much more suitable to achieve the desired results. These specialized vignettes will need to be evaluated on a case by case basis to decide the most valid engineering development path.

In addition, it seems evident that emphasis should be placed on developing higher level configuration, control and data capture systems. Development in these areas should allow for more reuse of equipment and software as well as higher level experimentation that could combine and integrate lower level systems.

In summary, we recommend that you:

4.1.1 UTE Software

1. Continue to use VBS software to provide the baseline soldier level simulation framework. Development of specific add-ons, modifications and hardware integration will likely be required in some cases.
2. Upgrade to VBS3. There are enough advancements in capability with VBS3 that upgrading from VBS2 is highly recommended. However, the upgrade to VBS3 may require careful testing and reconfiguring of software previously developed for use with VBS2, with a well-planned, phased integration should be undertaken.
3. Consider the selection and adoption of a single game engine technology to fulfill the requirements for some select, focused vignettes and tasks. This technology will enable the full leveraging of software capabilities such as virtual reality, integration of unique hardware and smaller scale form computing platforms such as tablets and cell phones.

4. Adopt a single supporting system for computer generated forces (CGF) scenario design and reuse across multiple experiments. It is possible that VBS2/3 could provide this functionality in lieu of a more sophisticated tool with some limitations (discussed further in 4.1.3).

5. Investigate the development of a common instructor dashboard and fully capable data capture and after-action review system, as comprehensive data capture and playback is critical to experiment conduct and analysis. Section 4.1.4 expands on some approach options for an integrated AAR capability.

4.1.2 UTE Hardware

1. Acquire and develop a common laptop solution that can emulate the ISS Builds and simulate other secondary information displays. The laptop solution shall be capable of interfacing with the VBS gaming environment (e.g. common maps, entity positions, map orientation) and shall be able to be networked with common laptops used by other soldier participants in an experiment, and with desktop controller workstations to emulate different soldier system architectures.
2. Acquire and investigate the suitability of using Pointman suites to improve the physical realism of being immersed in a virtual soldier environment.
3. Investigate the suitability of using an instrumented Assault Rifle (AR) for use as an instrumented control interface.

4.1.3 Computer Generated Forces Options

There are several commercially available options for DIS/HLA based CGF systems that could be used to plan and execute team experimentation. The Mak VR Forces product is in common use within DND and offers almost unlimited capability to create and control entities through its scripting and programming APIs. Full product documentation for VR Forces is available on-line.

Bohemia has also recently introduced their new VBS Tactics product. VBS Tactics is an intuitive, web-based 2D software interface that allows users to conduct real-time tactical exercises up to the company level. The interface enables users to configure doctrine-based orders of battle, plan a mission, run it in real-time, and review the results in after-action review. We have no ‘hands on’ experience with this product however so all information is company provided.

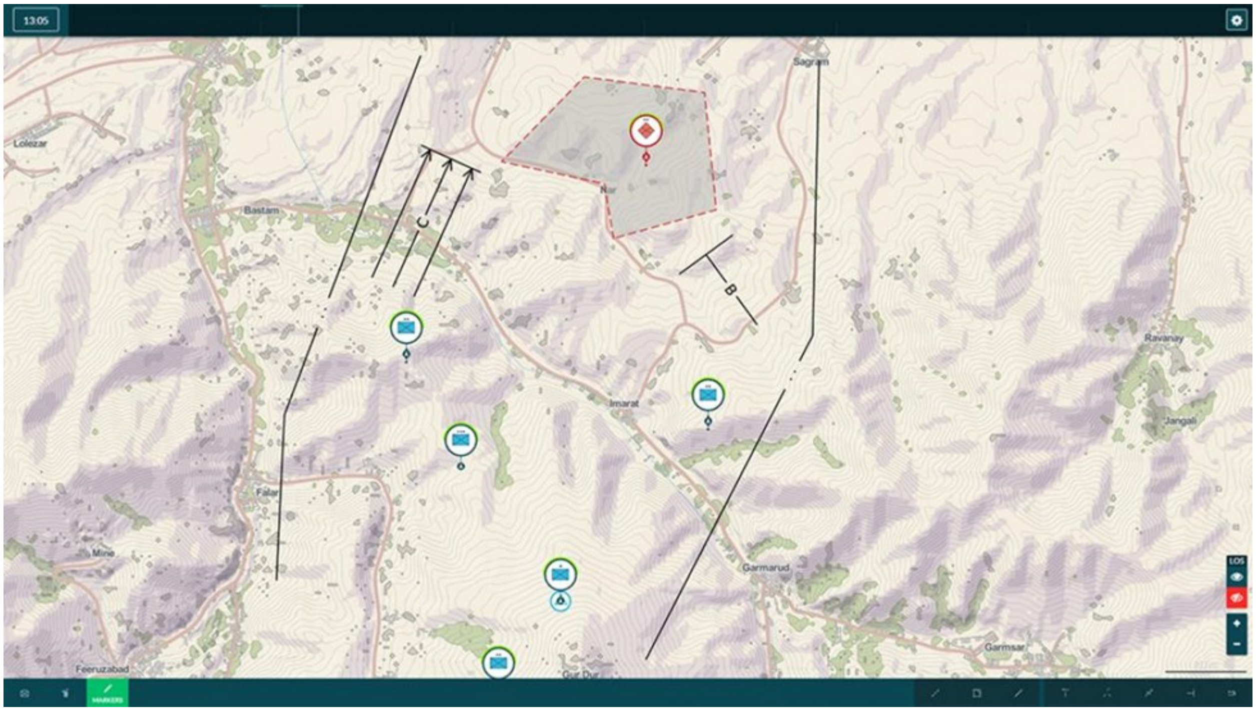

Figure 12: VBS Tactics

4.1.4 After Action Review Options

Assuming VBS2/3 will continue to provide the core framework for experimentation, the internal system AAR capabilities could be exploited. The AAR system integral to VBS2/3 has many strengths including tight coupling to the game engine and good line-of-sight and view shed analysis capabilities (i.e. the terrain area visible from a specific location) (Figures 13 and 14). Basic functionality such as record and playback are included along with rudimentary Mission Event Time-line (MET) and statistical reporting.

Figure 13: VBS2/3 After Action Review Plan View

It should be noted that voice communications recording and playback within the internal AAR system is only available in VBS3. Additionally, VBS2/3 uses its internal communications network between VBS instances for most information exchange including AAR record and playback. Capture and playback of non-VBS systems in any exercise would be limited to the information distributed via DIS or HLA network traffic.

However, the VBS AAR system does not have all the capabilities one would want in a fully integrated mission and data visualization tool. The VBS AAR has the capability of recording or playing mission video but the playback is not controllable within a master event timeline that is synchronized, and capable of displaying, other data types recorded at the time of mission execution.

Figure 14: VBS2/3 After Action Review LOS and View Shed

Alternatively, a Crown-owned system such as CH-146 Mission Rehearsal Tactics Trainer (MRTT) AAR system depicted below has been developed and could be adopted for use within the UTE environment. The MRTT AAR software, while originally designed for the Mak product, is open source and could be adapted for use with VBS3. This AAR system is fully customizable and capable of capturing and displaying any information desired. The system has a fully developed mission event system, in system and external video capture and playback, as well as network storage.

In its current implementation, the crown system has leveraged the Mak product suite for both plan view displays and 3D stealth displays. Line-of-sight and viewshed capabilities (i.e. area visible from a specific location) are resident within this software. As this AAR system is open source software, it would be very possible to replace the current use of Mak software with VBS2/3 to leverage commonality and familiarity, and achieve significantly more AAR capabilities and functionality for the sake of experimentation purposes.

The open source nature of this system also allows for integration of custom non-VBS software and hardware. Integration of experimental or development systems such as virtual reality devices, instrumented weapons, custom hardware and motion capture systems could be realized and coordinated with core software (VBS2/3) with such an open system.

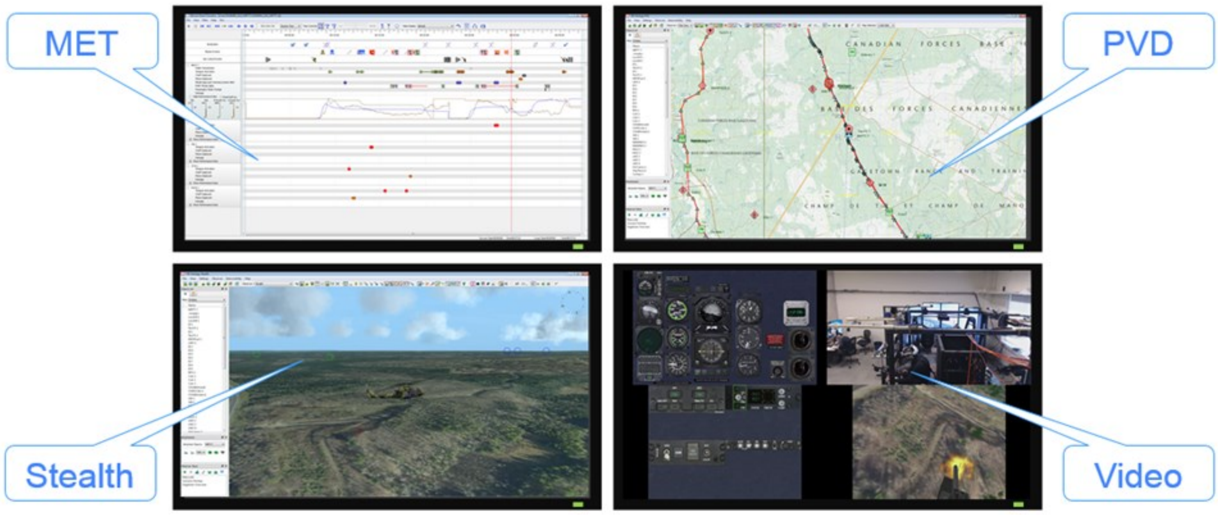

Figure 15: After Action Review System (DND)

4.2 Scenario Framework

A scenario framework was outlined for three different platoon-level Use Cases for organizing participants, confederates, and bots in a scenario formation, and strategies were identified for balancing scenario characteristics according to the organizational level of interest in team experimentation. Perspectives on scenario complexity and methods for setting and controlling such complexity were discussed in the context of scenario development. Based on this investigation, we believe that there are a number of developments necessary to realize and operationalize this framework.

It is recommended that DRDC:

1. Develop a summation scoring methodology to create an objective method of controlling scenario complexity.
2. Develop example scenarios using the scenario development framework and the summation scoring methodology.

5. Conclusions

To determine the requirements for a future Urban Team Experimentation (UTE) laboratory that can meet DRDC-Toronto's projected goals and objectives for future soldier team studies, this project aimed to determine high-level requirements for the UTE, evaluate candidate software and hardware systems, and provide a scenario development framework for creating and measuring performance in virtual small Unit missions.

Through consultation with key scientific and engineering stakeholders, goals and objectives were reviewed and capability requirements for the UTE were identified. Candidate virtual soldier software systems, and integrated hardware options, were evaluated against these capability requirements. While no one software solution currently meets all the requirements in their entirety most of the core UTE requirements can be achieved by the VBS2/3 software currently in use in DRDC-T. In the case of very specialized capability requirements, game engine software, which allows for a much deeper level of software functionality control, could be used selectively as appropriate for unique experimentation needs. Recommendations are also provided for the development of higher-level configuration, control and data capture systems to improve the speed and ease of software design and extraction of data.

A scenario framework was also outlined for organizing participants, confederates, and bots in a scenario formation, and strategies were identified for balancing scenario characteristics according to the organizational level of interest in team experimentation. Perspectives on scenario complexity and methods for setting and controlling such complexity were discussed in the context of scenario development. Based on this investigation, further work was proposed to realize and operationalize this framework.

6. References

Campbell, D. J. (1988). Task complexity: A review and analysis. Academy of management review, 13(1), 40-52.

Dunne, R., Schatz, S., Fiore, S., Nicholson, D., & Fowlkes, J. (2011). Optimizing decision preparedness by adapting scenario complexity and automating scenario generation. Proceedings of the ModSim International World Expo 2010, Hampton, VA. http://ntrs.nasa.gov/archive/nasa/casi.ntrs.nasa.gov/20110012060_2011012533.pdf.

Ferdinandus, G.R. (2012). Automated Scenario Generation, Coupling Planning Techniques with Smart Objects (Master's thesis).

Martin, G.A. (2012). Automatic Scenario Generation using Procedural Modeling Techniques (Doctoral dissertation).

Morton, A. & Adams, B. (2010). Development of a team scenario content generation framework. Report prepared for Defence Research and Development Canada – Toronto. DRDC Toronto CR-2010-105.

Tack, D.W. & Nakaza, E. (2016). Cognitive Task Analyses of Target Engagements in a Near Peer Scenario. Report prepared for Defence Research and Development Canada – Toronto. DRDC Toronto.

Tack, D., Bruyn Martin, L., Palmer, D. & Elderhorst, E. (2012). Section level experimentation, blue force tracking and identification friend or foe. Scenarios prepared for Defence Research and Development Canada – Toronto.

Templeman, J.N. and Denbrook, P.S. (2014) Avatar Control in Virtual Environments. In Handbook of Virtual Environments: Design, Implementation, and Applications. CRC Press – Taylor & Francis Group.

Wood, R. E. (1986). Task complexity: Definition of the construct. Organizational behavior and human decision processes, 37(1), 60-82.

DOCUMENT CONTROL DATA

*Security markings for the title, authors, abstract and keywords must be entered when the document is sensitive

1. ORIGINATOR (Name and address of the organization preparing the document. A DRDC Centre sponsoring a contractor's report, or tasking agency, is entered in Section 8.) DRDC – Toronto Research Centre Defence Research and Development Canada 1133 Sheppard Avenue West P.O. Box 2000 Toronto, Ontario M3M 3B9 Canada	2a. SECURITY MARKING (Overall security marking of the document including special supplemental markings if applicable.) CAN UNCLASSIFIED 2b. CONTROLLED GOODS NON-CONTROLLED GOODS DMC A

3. TITLE (The document title and sub-title as indicated on the title page.)

Urban Team Experimentation Requirements

4. AUTHORS (Last name, followed by initials; ranks, titles, etc., are not to be used.)

Tack, D.; Morton, A.; Arbuthnot, S.

5. DATE OF PUBLICATION (Month and year of publication of document.)	6a. NO. OF PAGES (Total pages, including Annexes, excluding DCD, covering and verso pages.)	6b. NO. OF REFS (Total references cited.)
July 2018	50	9

7. DOCUMENT CATEGORY (e.g., Scientific Report, Contract Report, Scientific Letter.)

Contract Report

8. SPONSORING CENTRE (The name and address of the department project office or laboratory sponsoring the research and development.)

DRDC – Toronto Research Centre
Defence Research and Development Canada
1133 Sheppard Avenue West
P.O. Box 2000
Toronto, Ontario M3M 3B9
Canada

9a. PROJECT OR GRANT NO. (If appropriate, the applicable research and development project or grant number under which the document was written. Please specify whether project or grant.)	9b. CONTRACT NO. (If appropriate, the applicable number under which the document was written.)

10a. DRDC PUBLICATION NUMBER (The official document number by which the document is identified by the originating activity. This number must be unique to this document.)	10b. OTHER DOCUMENT NO(s). (Any other numbers which may be assigned this document either by the originator or by the sponsor.)
DRDC-RDDC-2018-C125	

11a. FUTURE DISTRIBUTION WITHIN CANADA (Approval for further dissemination of the document. Security classification must also be considered.)

Public release

11b. FUTURE DISTRIBUTION OUTSIDE CANADA (Approval for further dissemination of the document. Security classification must also be considered.)

12. KEYWORDS, DESCRIPTORS or IDENTIFIERS (Use semi-colon as a delimiter.)

human factors; experimentation; laboratory; virtual environments; gaming software

13. ABSTRACT/RESUME (When available in the document, the French version of the abstract must be included here.)

The development of digital battlefield networks and robust new computing, sensing, and communications technologies promise improved effectiveness on the battlefield. However, significant questions exist regarding the true effectiveness and value of these systems, methods of best employment, and the potential burden of overloading soldiers with information on the battlefield. Defence Research and Development Canada at Toronto (DRDC-T) needs to determine the requirements for a future Urban Team Experimentation (UTE) laboratory that can be used to evaluate the claims and the implications of these information systems for future soldier systems.

Through consultation with key scientific and engineering stakeholders at Defence Research and Development Canada's Toronto laboratory, goals and objectives were reviewed and capability requirements for the UTE were identified. Candidate virtual soldier software systems, and integrated hardware options, were evaluated against these capability requirements. A framework was also discussed for developing infantry scenarios to employ in UTE experimentation. The findings of these investigations are discussed and recommendations are provided.

Le développement de réseaux de champs de bataille numériques et de nouvelles technologies robustes d'informatique, de détection et de communication promet une efficacité accrue sur le champ de bataille. Cependant, d'importantes questions subsistent quant à l'efficacité et à la valeur réelle de ces systèmes, aux meilleures méthodes d'utilisation et au fardeau potentiel que représente la surcharge d'information pour les soldats déployés sur le champ de bataille. Recherche et développement pour la défense Canada (RDDC) – Centre de recherches de Toronto doit déterminer les besoins d'un futur laboratoire d'expérimentation par une équipe urbaine (EEU) afin d'évaluer les prétentions et les implications de ces systèmes d'information pour les futurs systèmes de soldat.

En consultation avec les principaux intervenants scientifiques et techniques du laboratoire de RDDC à Toronto, nous avons examiné les buts et les objectifs recherchés et déterminé les besoins en matière de capacités EEU. Les systèmes logiciels de soldat virtuel en projet et les options matérielles intégrées ont été évalués en fonction de ces besoins en capacités. On a également discuté d'un cadre d'élaboration de scénarios d'infanterie EEU. Les conclusions de ces enquêtes font l'objet d'une discussion et des recommandations sont énoncées.