

VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA INFORMAČNÍCH TECHNOLOGIÍ
ÚSTAV INTELIGENTNÍCH SYSTÉMŮ

FACULTY OF INFORMATION TECHNOLOGY
DEPARTMENT OF INTELLIGENT SYSTEMS

NÁVRH A IMPLEMENTACE MODELŮ EFEKTŮ V SIMULAČNÍM SYSTÉMU ONESAF TESTBED BASELINE

DIPLOMOVÁ PRÁCE

MASTER'S THESIS

AUTOR PRÁCE

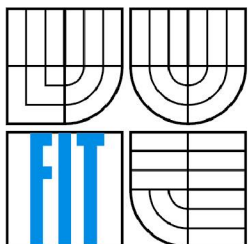
AUTHOR

BC. MARIAN HUBEŇÁK

BRNO 2009



VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
BRNO UNIVERSITY OF TECHNOLOGY



FAKULTA INFORMAČNÍCH TECHNOLOGIÍ
ÚSTAV INTELIGENTNÍCH SYSTÉMŮ

FACULTY OF INFORMATION TECHNOLOGY
DEPARTMENT OF INTELLIGENT SYSTEMS

NÁVRH A IMPLEMENTACE MODELŮ EFEKTŮ V SIMULAČNÍM SYSTÉMU ONESAF TESTBED BASELINE

DESIGN AND IMPLEMENTATION OF EFFECT MODEL IN SIMULATION SYSTEM ONESAF
TESTBED BASELINE

DIPLOMOVÁ PRÁCE
MASTER'S THESIS

AUTOR PRÁCE
AUTHOR

BC. MARIAN HUBEŇÁK

VEDOUCÍ PRÁCE
SUPERVISOR

ING. JAROSLAV ROZMAN

BRNO 2009

Abstrakt

Tato práce se zabývá oborem modelování a simulace, v němž popisuje základní charakteristiky a rozdělení tohoto oboru. Větší část je poté věnována modelování a simulaci ve vojenství, kde uvádím základní přínos pro vojenství, spolu s různým druhem použití jednotlivých simulací, dále seznamuji s vývojem modelování a simulace v Armádě České republiky. Práce poté popisuje obor konstruktivní simulace a navazuje se seznámením simulačního systému OneSAF Testbed Baseline. Nakonec uvádím návrh modelů pro tento simulační systém – model psychologických efektů a model generování suti při bojové činnosti dělostřelectva.

Klíčová slova

Modelování, simulace, model, simulátor, vojenství, konstruktivní simulace.

Abstract

This project deals with the modelling and the simulation, its basic characteristics as well as its respective branches. The majority of the project is aimed at modelling and simulation in military environment with a description of the asset in this area while considering different uses of these simulations. I also introduce the development of modelling and simulation in the army of Czech Republic. The project also deals with the constructive simulation and tries to bring out the OneSAF Testbed Baseline simulation system. At last, I propose some concepts of models for this particular simulation system - the model of psychological effects and the model of generating the rubble during the operations of the artillery.

Keywords

Modeling, simulation, simulator, military, constructive simulation

Citace

Hubeňák Marian: Návrh a implementace modelů efektů v simulačním systému OneSAF Testbed Baseline, diplomová práce, Brno, FIT VUT v Brně, 2009

Návrh a implementace modelů efektů v simulačním systému OneSAF Testbed Baseline

Prohlášení

Prohlašuji, že jsem tuto diplomovou práci vypracoval samostatně pod vedením Ing. Jaroslava Rozmana.

Další informace mi poskytla firma VR Group, a.s.

Uvedl jsem všechny literární prameny a publikace, ze kterých jsem čerpal.

.....
Marian Hubeňák
21.5.2009

Poděkování

Rád bych poděkoval za odbornou pomoc při tvorbě této diplomové práce mému vedoucímu - pan Ing. Jaroslav Rozman, konzultantům - Ing. Jiří Popelka, Ing. David Řezáč, Ph.D. a firmě, která mi tuto práci zadala – firma VR Group, a.s.

© Marian Hubeňák, 2009.

Tato práce vznikla jako školní dílo na Vysokém učení technickém v Brně, Fakultě informačních technologií. Práce je chráněna autorským zákonem a její užití bez udělení oprávnění autorem je nezákonné, s výjimkou zákonem definovaných případů.

Obsah

Obsah.....	1
1 Úvod	3
2 Simulace a modelování.....	4
2.1 Klasifikace simulace	4
2.1.1 Fyzická a interaktivní simulace	4
2.1.2 Počítačová simulace	5
2.1.3 Počítačová věda.....	6
2.2 Simulace ve výuce	6
3 Modelování a simulace ve vojenství	8
3.1 Taktický výcvik jednotlivců	8
3.2 Výcvik velitelů a štábů	10
3.3 Rekonfigurovatelné virtuální simulátory	11
4 Konstruktivní simulace.....	13
4.1 Systémy konstruktivní simulace	14
4.1.1 FLAMES	14
4.1.2 VR-Forces	15
4.1.3 JCATS	16
4.1.4 OOS	16
4.1.5 OTB	17
5 Simulační systém OneSAF Testbed Baseline	18
5.1 Základní znaky architektury DIS	18
5.2 Semi-Automated Forces (SAF)	19
5.3 OTBSAF	19
5.4 Moduly architektury OTBSAF	20
5.5 OTBSAF architektura	21
5.5.1 Kolektivní chování	21
5.5.2 Individuální chování.....	22
5.5.3 Fyzický subsystém.....	22
5.6 Metodologie návrhu.....	22
5.6.1 Vrstvy	22
5.6.2 Objektově orientované programování	23
5.6.3 Přesná specifikace rozhraní	23
5.6.4 Datově řízené spuštění.....	23
5.7 Architektura řízení a ovládání.....	24

5.8	Základní knihovny architektury OTBSAF.....	25
5.8.1	Řízení činností a úkolů	25
5.8.2	Práce s terénní databázi	26
5.8.3	Zpracování paketů PDU	26
5.8.4	Čtení konfiguračních souborů	26
6	Návrh modelu psychologických efektů při bojové činnosti dělostřelectva	27
6.1	Neformální specifikace	27
6.2	Návrh modelu	28
6.2.1	Spuštění běhu modelu.....	28
6.2.2	Konfigurace modelu	28
6.2.3	Stavy modelu	29
6.3	Implementace modelu psychologických efektů.....	31
6.3.1	Popis jednotlivých souborů knihovny	31
6.3.2	Načítání konfiguračních informací	31
6.3.3	Běh modelu.....	32
6.3.4	Výpis informací o psychologickém efektu.....	33
7	Návrh modelu generování suti a klád při bojové činnosti dělostřelectva	34
7.1	Neformální specifikace	34
7.2	Návrh modelu	38
7.2.1	Konfigurace modelu	38
7.2.2	Formát konfiguračního souboru	38
7.3	Návrh modelu generování klád	40
7.3.1	Zjištění hranice lesa.....	40
7.4	Návrh modelu generování suti	42
7.4.1	Práce s budovami na terénní databázi.....	42
7.5	Implementace modelu generování suti a klád.....	44
7.5.1	Popis jednotlivých souborů knihovny	44
7.5.2	Inicializace knihovny.....	44
7.5.3	Běh modelů generování suti a klád	45
8	Závěr.....	49
	Literatura.....	50
	Seznam příloh.....	52

1 Úvod

Diplomová práce se zabývá oborem modelování a simulace, zejména použití ve vojenství. Mým úkolem je vytvořit dva druhy modelů pro simulační systém OneSAF Testbed Baseline (OTBSAF), který používá Armáda České republiky k výuce a výcviku našich vojáků. Tento simulační systém je založen na principu konstruktivního simulování a byl vyvinut pro potřeby Armády USA.

Druhá kapitola této práce pojednává obecně o modelování a simulaci, přičemž uvádím základní rozdělení a klasifikaci tohoto oboru a zaměřuji se taktéž na nejčastější použití simulace, a to na použití k výukovým účelům.

V třetí kapitole uvádím význam, vlivy modelování a simulace v oboru vojenství, kde popisuji jednotlivé uplatnění konkrétních simulací v daném oboru a jejich dopad a přínos pro vojenské účely. Dále uvádím příklady různých výcviků za použití simulace a modelování v Armádě ČR.

Konstruktivní simulace je hlavní téma čtvrté kapitoly. Popisuji zde, čím se tato simulace zabývá, která metodologická hlediska jsou pro ni významná, popisuji jednotlivé metody a modely konstruktivní simulace. Poté se zabývám použitím modelů a problémy, které při této simulaci mohou nastat, nebo se zde již vyskytují. Na závěr uvádím přehled nejznámějších systémů konstruktivní simulace, jejich výhody a použití.

V páté kapitole popisuji detailněji simulační systém OneSAF Testbed Baseline (OTBSAF). Popisuji možnosti a použití tohoto systému, rozebírám architekturu, moduly, knihovny a metodologii návrhu v rámci celého systému OTBSAF.

V šesté a sedmé kapitole navrhuji pro výše popsany simulační systém OTBSAF modely, a to model psychologických efektů a model generování suti při bojové činnosti dělostřelectva. Uvádím kdy, jak a za kterých okolností dochází k daným efektům, rozebírám jednotlivé případy a jejich vliv na simulaci. Dále navrhuji a uvádím implementaci jednotlivých modelů do výše zmíněného systému.

Závěrečná osmá kapitola přináší shrnutí celé této práce.

Počátek této diplomové práce byl zpracován v semestrálním projektu, kde jsem uvedl základy simulace a modelování, význam a vlivy na obor vojenství a vysvětlil základy konstruktivní simulace, spolu s úvodem k systému OTBSAF. Nyní na tyto základy navazuji a rozšiřuji tyto části o další informace, spolu s návrhem a implementací modelů efektů do simulačního systému OneSAF Testbed Baseline.

2 Simulace a modelování

Simulace a modelování je proces napodobování reality, či událostí. Hlavní podstatou tohoto procesu je věrohodně napodobit charakteristické rysy nebo chování zvoleného fyzického, či abstraktního modelovaného systému.

Klíčové problémy v simulaci zahrnují získání základních informací o systému, výběr klíčových charakteristických rysů a chování tohoto systému. Dále je nutné použít zjednodušujících přiblížení a předpokladů uvnitř simulace a v neposlední řadě taktéž správně vyhodnotit zjištěné výsledky daného modelování [1].

2.1 Klasifikace simulace

Historicky se simulace vyvíjela v mnoha oblastech velkou měrou nezávisle, ale 20. století za použití systémových teorií a kybernetiky, kombinované spolu s rozšiřujícím se použitím osobních počítačů, vedlo ke sjednocení oblastí a vyčlenění modelování a simulace do tří konceptů.

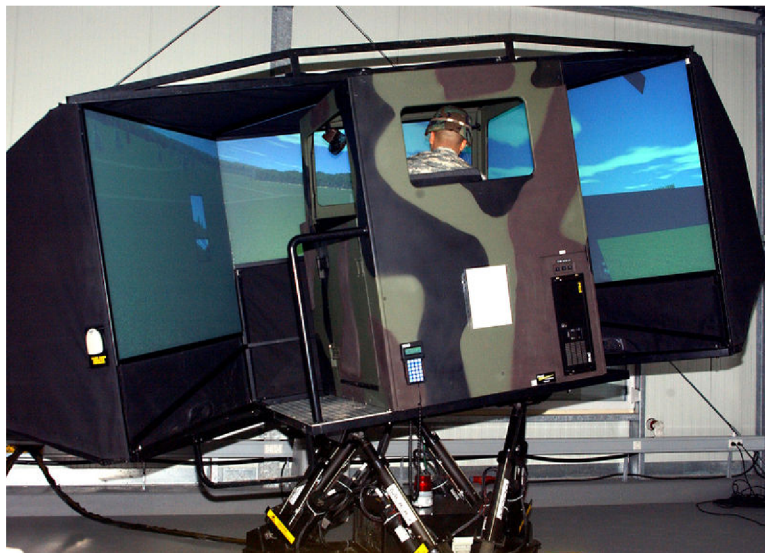
2.1.1 Fyzická a interaktivní simulace

- *Fyzická simulace* odkazuje na skutečný systém, ale vybrané prvky systému jsou nahrazeny pouze simulátory. Tyto prvky jsou často vybírány vzhledem k jejich velikosti, případně ceně vzhledem k celému skutečnému systému [1].



Obr. 1: Fyzická simulace - simulace crash-testu

- *Interaktivní simulace* je zvláštní druh fyzické simulace. Tato simulace zahrnuje do fyzické simulace jako činitele člověka (např. letecké simulátory) a často jsou tyto simulace propojeny s počítačovou simulací (např. modelování reakce obsluhy při krizi v jaderné elektrárně) [1].



Obr. 2: Interaktivní simulace - simulace řidiče kolového vozidla [1]

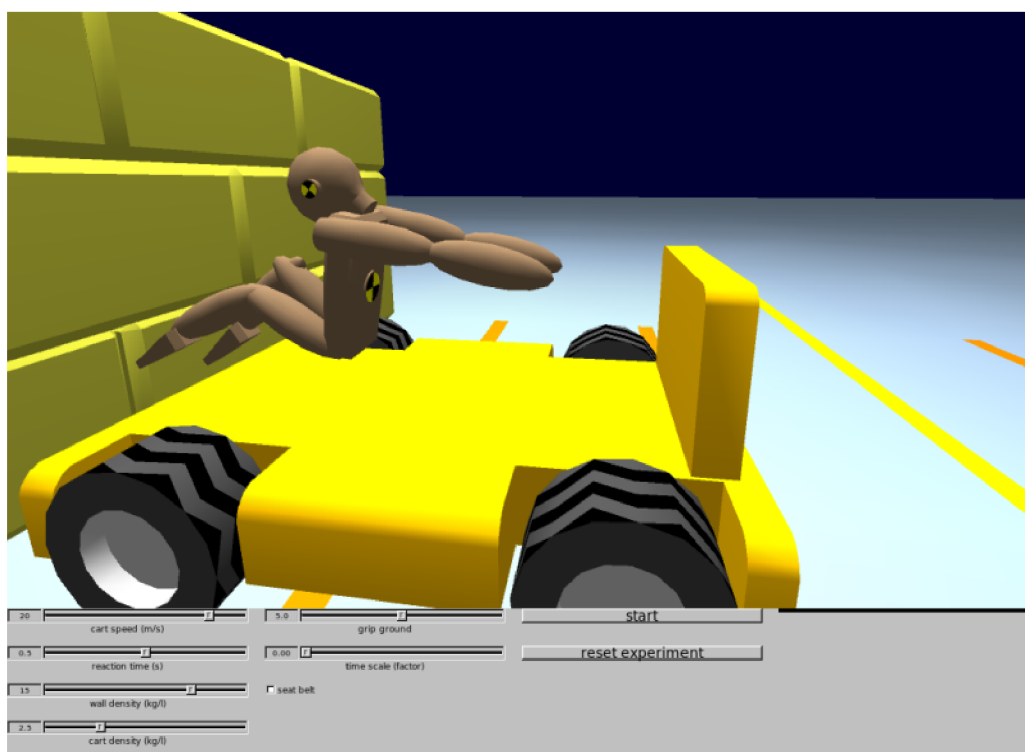
2.1.2 Počítačová simulace

Počítačová simulace je pokus o modelování reálného světa nebo hypotetické situace na počítači, abychom poté mohli tento model studovat a analyzovat jak pracuje. Nastavováním a změnou různých proměnných můžeme následně vyvozovat závěry o chování modelovaného systému [1].

Počítačová simulace se stala nejvíce užitečnou při modelování přirozených systémů ve fyzice, chemii a biologii, dále v lidských systémech jako ekonomika a sociální vědy a v neposlední řadě také v inženýrství.

Tradičně se simulovaný model popsal formálně za použití matematických modelů, které se pokoušely najít analytické řešení, umožňující předpovědět chování systému ze souboru parametrů a počátečních podmínek. Počítačová simulace je často užívána jako přídavek, nebo náhrada při modelování systémů, pro které analytické řešení není možné, případně je analytické řešení velmi složité [1].

V současné době již existují softwarové balíky určené pro počítačovou simulaci (např. Monte Carlo simulace a stochastické modelování), které výrazně zjednodušují návrh simulace a modelování námi zvoleného systému.



Obr. 3: Počítačová simulace - simulace crash-testu za pomoci počítače

2.1.3 Počítačová věda

V počítačové vědě má termín „simulace“ zvláštní význam: Alan Turing (1912 - 1954) používal termín „simulace“ k popisu, co se stává, když počítač postupně přechází v tabulce stavů, která popisuje změny stavů, vstupy a výstupy diskretního automatu. Počítač zde simuloval daný automat. Podle tohoto, je v teoretické informatice termín simulace vztah mezi přechodem jednotlivých stavů [1].

Zajímavou aplikací počítačů je simulace počítače na počítači. V počítačové architektuře takovýto simulátor nazýváme emulátorem a často se používá ke spuštění programů, které mohou běžet pouze na určité skupině strojů, jenž momentálně nemáme k dispozici. Tímto způsobem můžeme spouštět aplikace daných počítačů za pomoci emulátoru na stroji, který k dispozici máme.

2.2 Simulace ve výuce

Simulace se často využívá k výuce a trénování civilních a vojenských osob. Tyto simulace jsou často použity v případech, kdy použití skutečného vybavení ve výuce je příliš drahé, případně použití ve skutečném světě je příliš nebezpečné. V těchto situacích je výhodné použití simulátoru, kde dané osoby si mohou vše bezpečně vyzkoušet a naučit, a navíc v těchto simulátorech můžeme povolit a sledovat chyby, které by ve skutečném světě neměly nastat.

Trénování za použití simulátorů můžeme rozdělit do tří kategorií [1]:

- **živá simulace** – skuteční lidé zde používají simulované nástroje, ale ve skutečném prostředí
- **virtuální simulace** – skuteční lidé používají simulované nástroje ve virtuálním (simulovaném) prostředí
- **konstruktivní simulace** – zde je vše simulováno (lidé, nástroje i prostředí); tato simulace se někdy též nazývá „válečné hry“, kde velitel jakoby velí (simuluje) boj v daném prostředí za použití simulovaných nástrojů a lidí (vojáků)

3 Modelování a simulace ve vojenství

Ve vojenství jsou modely, simulace, simulátory a trenažéry bojové činnosti (taktické, operační a strategické) známé a používané již velmi dávno. Začátky modelování a simulace sahají hluboko do historie vojenství a jejich další rozvoj byl interaktivně ovlivňován pokroky ve vývoji vojenské techniky a ozbrojených sil.

V dnešním vojenství, díky vlivům výkonných počítačů, další informační techniky, technologie a rozvoji nových kvantitativních metod modelování a simulace, se prudce rozvíjí a v praxi úspěšně aplikuje široká škála modelů a simulací, virtuálních simulátorů, trenažérů, modifikovaných klasických vojenských her a cvičení s počítačovou podporou i bez ní, s nejrůznější specifikací a určením. Jsou to právě moderní simulační systémy, které tvoří základ principů nových výukových technologií umožňujících vykonávat inteligentní, efektivní a intenzivní výcvik i v podmínkách blízkých skutečným (bojovým), které se nedají v extrémních případech dosáhnout žádným jiným známým klasickým prostředkem či metodou. Simulace zde bývají lepším řešením pro zpravidla menší ekonomické náklady a další pozitivní efekty [2].

Modely a metody simulace se efektivně využívají při vývoji nových druhů zbraní, bojové techniky, munice a materiálů. Jsou taktéž jednou ze základních součástí automatizovaných systémů velení a řízení, nástrojem na zvýšení jejich inteligence, efektivity a produktivity při řešení reálných bojových úloh.

Zkušenosti a výsledky dosažené použitím klasických metod modelování a simulace ve vojenství pozitivně ovlivnily široký rozvoj a uplatnění těchto prostředků i v civilním sektoru. Poznatky a zkušenosti získané při vývoji a využívání modelování a simulace v civilní sféře naopak zpětně ovlivňují a stimulují jejich aplikace ve vojenství.

Níže uvádím různé příklady výcviků používaných v současné době v armádě ČR [4], [5].

3.1 Taktický výcvik jednotlivců

Výcvik jednotlivců je založen na systémech distribuované simulace kompatibilních se standardy DIS/HLA. Tato kompatibilita umožňuje propojovat osádkové virtuální simulátory se systémy konstruktivní simulace, a tak vytvořit komplexní scénář boje obsahující síly vlastních vojáků, nepřítelů a další účastníky. Funkčnost tohoto propojení je v současnosti zajištěna použitím systému OTBSAF [4].

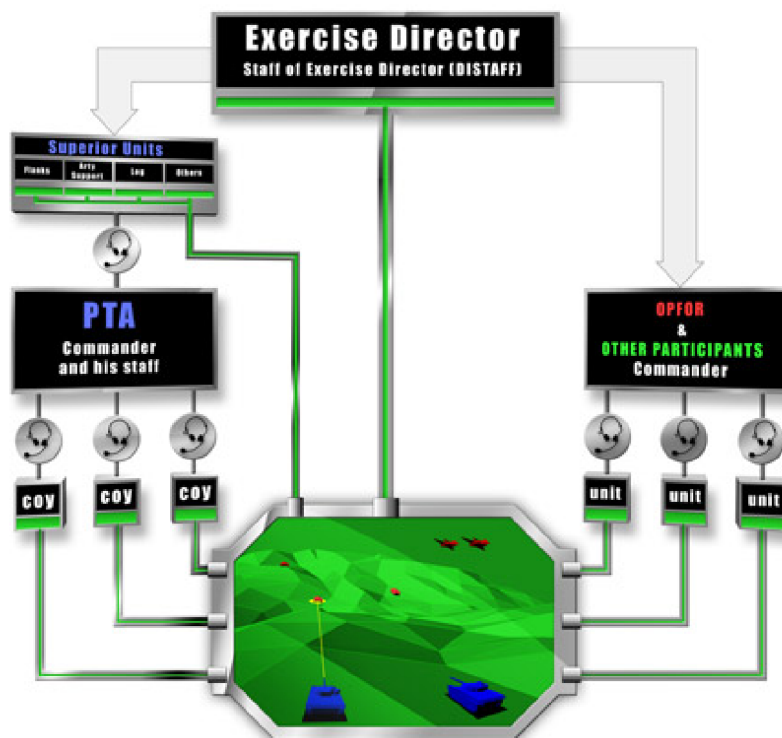
Komunikace mezi cvičícími a řídicím cvičením, který zároveň hraje roli jejich nadřízeného, probíhá prostřednictvím simulované rádiové sítě. Řídicí cvičení používá simulační nástroje pro přípravu, řízení a vyhodnocení výcviku.



Obr. 4: Schéma taktického výcviku [4]

- **Cvičící (Primary training audience)** – cvičící ovládají rekonfigurovatelné virtuální simulátory (VS II) tak, aby splnili úkoly a rozkazy nadřízeného
- **Řídicí cvičení (Exercise director)** – řídicí cvičení, který je zároveň v roli nadřízeného, řídí činnost cvičícího velitele a podle potřeby i jeho podřízených. Sleduje a zaznamenává všechny aktivity v syntetickém prostředí pomocí nástrojů konstruktivní a virtuální simulace včetně provozu v simulované rádiové síti.
- **Rozehra (Role players)** - Operátoři rozehry prostřednictvím systémů konstruktivní a virtuální simulace rozehrávají činnost nepřítele a vlastních sil (sousední jednotky, dělostřelecká podpora, letecká podpora a další složky).

3.2 Výcvik velitelů a štábů



Obr. 5: Schéma cvičení velitelů a štábů s počítačovou podporou [5]

- **Řídící cvičení (Exercise director)** - řídící cvičení řídí přípravu, provedení a rozbor cvičení. Řídí činnost nadřízeného cvičících, nepřítelů a dalších účastníků cvičení. Sleduje a zaznamenává všechny činnosti v syntetickém prostředí včetně komunikace v simulované rádiové síti. Připravuje podklady a řídí rozbor cvičení.
- **Nadřízený (Superior units)** - operátoři rozehrávají činnost nadřízeného cvičícího štábu, jeho jednotek a dalších prvků podle pokynů řídicího cvičení. Zde se vydávají rozkazy a informace cvičícímu štábu a přijímají od něj hlášení prostřednictvím simulovaného systému velení a řízení.
- **Cvičící štáb (PTA)** - Velitel a štáb používají systém velení a standardní systém spojení řízení k vedení bojové činnosti, vydávání rozkazů, přijímání a podávání hlášení od podřízených a nadřízených, stejně jako v reálném boji.
- **Nepřítel a další složky (OPFOR and other participants)** - operátoři ovládají vozidla jednotek nepřítelů a dalších zúčastněných v syntetickém prostředí podle pokynů řídicího cvičení.
- **Podřízené jednotky (Coy, Unit)** - operátoři podřízených jednotek ovládají přidělené jednotky a vozidla v syntetickém prostředí (systém konstruktivní simulace) tak, aby

realizovali rozkazy a nařízení cvičícího velitele a jeho štábu. Zároveň mu podávají informace získané ze simulačního systému, např. stav jednotky, činnost jednotky, kontakt s nepřítelem, požadavky na doplnění, nebo podporu.

3.3 Rekonfigurovatelné virtuální simulátory

Rekonfigurovatelné virtuální simulátory (VS II) jsou vhodné pro taktický výcvik malých jednotek a velitelů vozidel.

Vzrůstající potenciál simulátorů VS II umožňuje rekonfigurace na nové typy vozidel, vrtulníků, předsunutého dělostřeleckého pozorovatele, leteckého návodčího a další typy vozidel nebo osob podle požadavků zákazníka.

Simulátory VS II používají protokol DIS/HLA, tudíž jsou propojitelné s jinými simulátory nebo systémy konstruktivní simulace kompatibilní s těmito standardy. Tak lze podle výcvikových cílů vytvořit komplexní scénář boje včetně sil vlastních a nepřítele.

Simulátor VS II je navržen jako modulární systém, který se skládá ze simulační jednotky a jednotek generátoru obrazu umístěných v reku a prostorů pro členy osádky. Zjednodušené prostory osádky jsou vybaveny: joystickem, monitorem s 3D pohledem na válčiště a indikátorem vybraných informací o stavu vozidla, komunikačním systémem (simulované rádiové spojení a interkom) [6].



Obr. 6: Virtuální simulátor VS II [6]



Obr. 7: Výhled pilota Mi-17 ve virtuálním simulátoru VS II [6]



Obr. 8: Výhled řidiče BVP-2 ve virtuálním simulátoru VS II [6]

4 Konstruktivní simulace

Pod pojmem konstruktivní simulace se rozumí simulace, která se realizuje v zásadě na logicko-matematických modelech deterministického nebo stochastického charakteru. Síly a prostředky bojujících stran, jako i prostředí bojiště se svými vlastnostmi, v těchto modelech vyjadřují matematické výrazy, rovnice, nerovnice a algoritmy [2].

V konstruktivní simulaci nevystupují žádné reálné (fyzické) nebo virtuální (fotorealistické) jevy, objekty a procesy bojové činnosti, ani reálné či virtuální prostředky bojiště. Reálně a bezprostředně operuje jen subjekt simulace (uživatel, operátor), který připravuje výchozí situaci a zabezpečuje hodnocení výsledků simulace.

Jednou, avšak poměrně významnou výjimkou, jsou případy, kdy jsou v matematických modelech bojové činnosti zabudované otevřené algoritmy a jejich nealgoritimizované části řeší člověk jako integrální součást modelu a celého uzavřeného simulačního procesu probíhajícího v tzv. interaktivním režimu. Simulace tohoto druhu (označované někdy jako nepravé) se uplatňují v situaci, kdy se v simulačním procesu vyskytují nealgoritimizovatelné úseky, anebo pokud se algoritmy ještě nepodařilo sestavit, případně pokud tyto úseky záměrně řeší cvičící subjekty (velitelé, štáby, či obsluhy zbraňových a technických systémů) [3].

Modely konstruktivní simulace tvoří významné komponenty i v jiných druzích simulací ozbrojeného zápasu a představují tak univerzální sjednocující základ pro všechny kategorie simulací. Tyto modely a simulace mají oproti ostatním mnoho významných předností a výhod. Některé z nich umožňují například přímo nacházet optimální varianty řešení různých složitých rozhodovacích problémů. Jsou také velmi flexibilní (protože problém interpretace jejich parametrů a vazeb je zpravidla nenáročný na realizaci), mohou se pomocí nich opisovat a analyzovat situace libovolného druhu a rozsahu bojové činnosti s požadovaným stupněm detailu, a také jsou ekonomicky zpravidla nejvýhodnější, a tedy nejdostupnější pro široký okruh uživatelů [2].

Nejpádnější důvod na použití modelů konstruktivní simulace je, že umožňuje ekonomicky efektivně simulovat prostředky a procesy ničení, destrukce a umlčování živé síly a bojové techniky a objektů, a to nejen ty, které dnes už existují, ale také ty, které jsou prozatím předmětem výzkumu a vývoje. V současné době, kdy státy s mohutným vědecko-technickým potenciálem zkoumají možnosti nových druhů zbraní (laserových, infrazvukových, ultrazvukových, apod.), mají tyto modely mimořádný vědecký i praktický význam i pro menší armády, a to z hlediska zkoumání ochrany před možnými účinky těchto zbraní s následnou realizací potřebných opatření v ozbrojených silách s celostátním rozsahem.

K problémům v aplikaci modelů konstruktivní simulace může dojít při vyjadřování kvalitativních, tzv. měkkých faktorů ozbrojeného zápasu, které mají významný vliv na jeho příběh i konečný výsledek. Patří k nim psychické (rozumové, citové – emociální, etické apod.),

psychofyzilogické (vztahující se na smyslové ústrojí) a fyziologické (fyzická zdatnost, zdravotní stav, tělesná kondice, apod.) vlastností člověka – účastníka ozbrojeného střetnutí. Projevují se v charakteristicky spontánním nebo cílevědomém (záměrném) chování jednotlivců a sociálních skupin (vojenských kolektivů) při plnění úloh v reálné bojové činnosti. Tyto problémy se řeší různými způsoby, například tím, že se do matematických modelů zabudovávají různé kvalitativní koeficienty s prvky umělé inteligence nebo se do simulačního procesu začleňuje bezprostředně člověk v roli jeho objektu i subjektu [2].

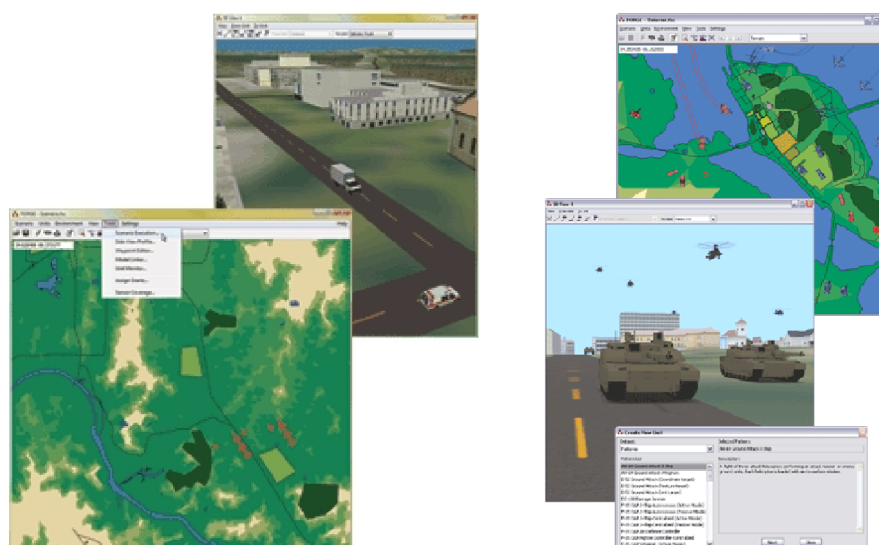
Metody konstruktivní simulace, jeho jednotlivé části, prvky a elementární procesy patří dnes k nejvýznamnějším a nejrozšířenějším metodám. V oblasti teorie i praxe se používá velký počet různých druhů modelů, různé složitosti, dokonalosti, přesnosti a kvality specifického určení. Konstruktivní simulace v těchto modelech se v současné době uskutečňuje zpravidla na počítačích, a to buď formou centrálního zpracování (na jednom počítači) anebo distribuovaně – v počítačové síti.

4.1 Systémy konstruktivní simulace

V současné době existují různé druhy systémů konstruktivní simulace, liší se navzájem od sebe architekturou, zaměřením, simulačními možnostmi, ovládním, prostředím, analýzou, atd. Níže uvádím přehled vybraných simulačních systémů, jejich možnosti a přednosti.

4.1.1 FLAMES

FLAMES (FLexible Analysis, Modeling, and Excercise System) je rodina komerčních softwarových produktů vytvořena pro poskytnutí základu pro konstruktivní simulaci a rozhraní mezi konstruktivní, virtuální a živou simulací. Architektura tohoto systému je otevřená a vystavěna na objektově orientovaném programování. FLAMES ve svém základu dokáže simulovat jakýkoli systém v různém druhu prostředí (pevnina, moře, vzduch, vesmír) s velmi vysokou věrností a přesností. Při simulacích je možno využít různých druhů analýz podporující velmi vysoké rozlišení, krokování, zpětné přehrávání, Monte-Carlo metody a komplexní vícedenní analýzy. Systém dokáže pracovat s rozlehlými mapami a vytvářet komplexní scénáře ve 2D i 3D prostředí. Kterákoli simulovaná entita může být za pomoci operátora (člověka) kontrolována a řízena. FLAMES taktéž dokáže spolupracovat s mnoha virtuálními simulátory, skutečnými senzory a komunikačními systémy [8].



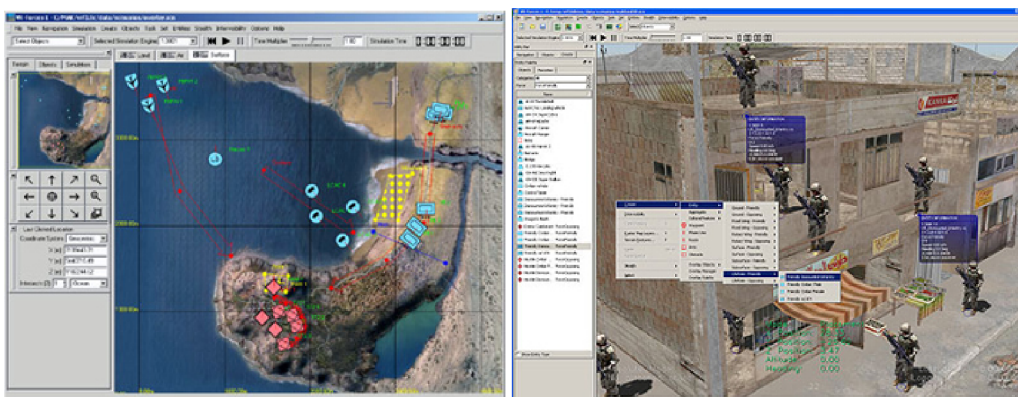
Obr. 9: Ukázka simulačního systému FLAMES [8]

4.1.2 VR-Forces

VR-Forces je velice silná a flexibilní simulační platforma pro generování a spuštění válečných scénářů. Zahrnuje v sobě veškeré nezbytné vlastnosti pro konstruktivní simulaci jako taktický trenažér, generování nepřátel, simulaci různého modelu chování, apod.

Se systémem VR-Forces je dodáváno intuitivní uživatelské rozhraní, které dovoluje vytvářet scénáře jednoduchým rozmístěním sil vlastních a nepřítele, budovat cesty a kontrolní stanoviště, přiřazovat mise, či úkoly jednoduchým systémem, rozmisťovat různé ikony na 2D taktické mapě, nebo přímo v 3D scéně umisťovat entity např. do budov. V průběhu simulace veškerá vozidla a lidé reagují na okolní terén, následují silnice, vyhýbají se překážkám, komunikují přes simulovanou rádiovou síť a detekují a reagují na nepřítele.

Dále je k systému VR-Forces přidán grafický editor jednotek, díky kterému je možné upravovat, či rozšiřovat stávající modely entit, měnit parametry vozidel, senzorů, nebo tabulek poškození přímou, či nepřímou palbou [9].

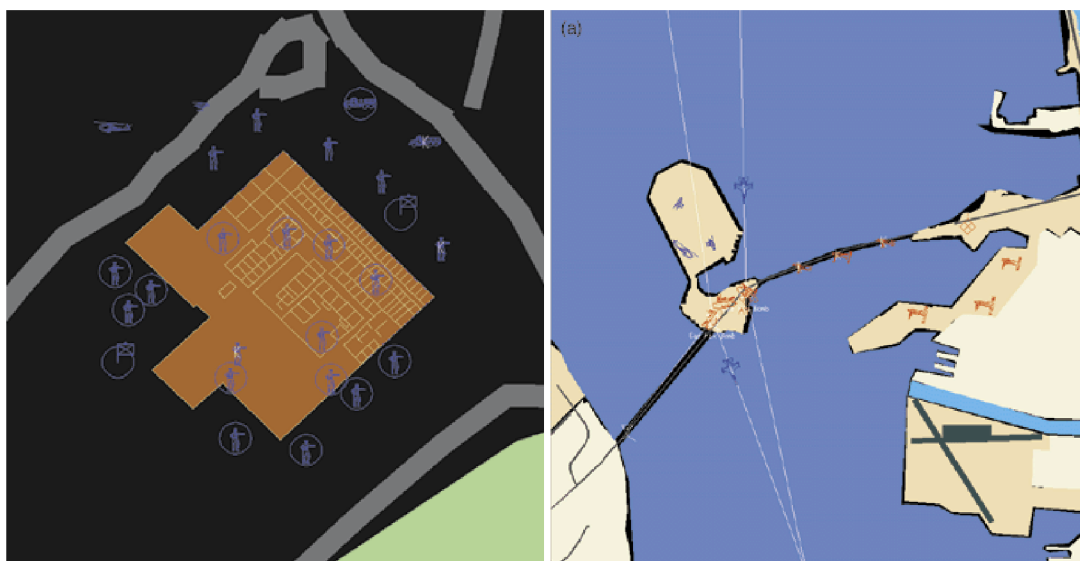


Obr. 10: Ukázka simulačního systému VR-Forces [9]

4.1.3 JCATS

Program JCATS (The Joint Conflict and Tactical Simulation) je interaktivní simulační nástroj využíváný k trénování, analýze, plánování misí a nácviku bojových složek. JCATS dokáže velice dobře simulovat operace v zalidněné oblasti, podporuje širokou škálu různých zbraní (od omračujících až po konvenční druhy zbraní) a uživatelům dovoluje rychle operovat s entitami a celými jednotkami [10].

Systém dokáže simulovat až 60 000 individuálních elementů počínaje vojákem, letadlem, až po dav různých účastníků. Typickou simulací je konflikt mezi dvěma různými stranami (vlastní a nepřítel), avšak JCATS dokáže nasimulovat až 10 stran s přiřazením do jedné z neutrálních, nepřátelských, či vlastních složek. V závislosti na nastavení pravidel boje dle aktuálních podmínek může voják být naprogramován k tomu, aby střílel ihned po spatření nepřátelských sil, skrýval se, vykopal zákop, střílel pouze po potvrzeném nepříteli, nebo provedl jinou z předem připravených akcí. Pravidla boje mohou být kdykoli během simulace změněna v závislosti na politických rozhodnutích, či jiných okolnostech boje [11].

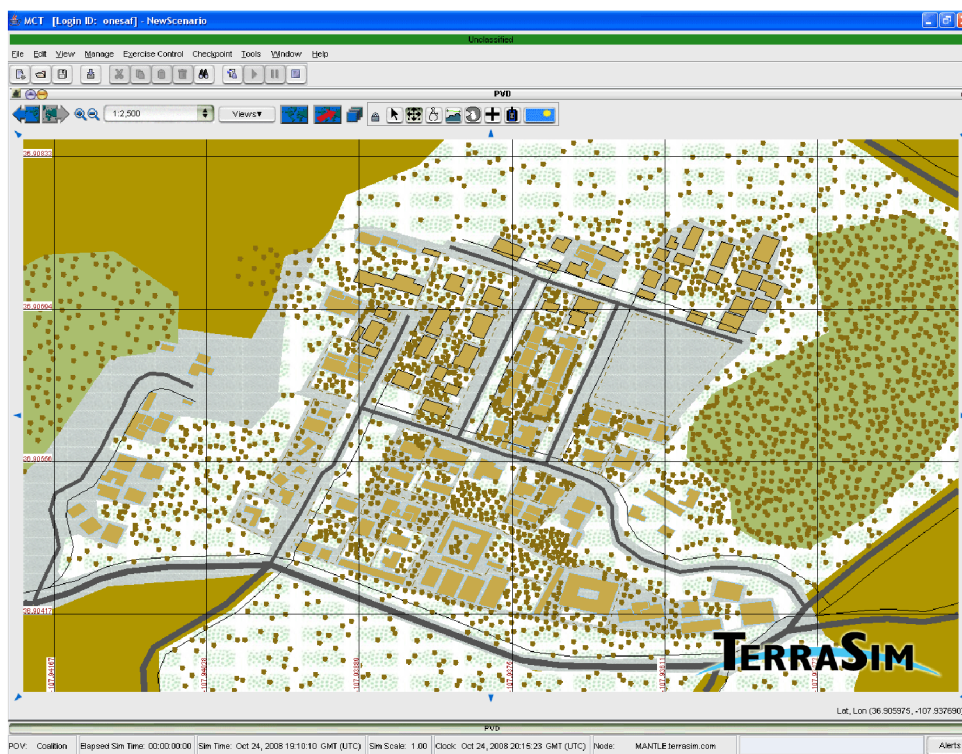


Obr. 11: Ukázka simulačního systému JCATS [11]

4.1.4 OOS

The OneSAF Objective System (OOS) je simulační systém na úrovni entit navržený k tomu, aby pokryl široké spektrum bojových akcí, bojové podpory a logistických služeb vojskům od úrovně brigády, až k jednotlivým jednotkám a vojákům. OOS je nástroj určený k vojenskému vzdělání, analytickým účelům, hodnocení sil různých vojenských uskupení, struktur, vybavení, taktiky a techniky [12]. Systém byl vytvořen pro potřeby Armády USA za účelem nahrazení všech

dosavadních simulátorů; byl vyvinut na základě zkušeností s používanými simulačními nástroji a při vývoji byl kladen důraz na verifikaci, validaci, rigorózní testování a přezkušování.



Obr. 12: Ukázka simulačního systému OOS

4.1.5 OTB

Simulační systém OTB (OneSAF Testbed Baseline) detailněji popisují na následujících stránkách.

5 Simulační systém OneSAF Testbed

Baseline

Simulační systém OTBSAF je určen pro konstruktivní simulaci a slouží v současné době pro přípravu velitelů a štábů na taktickém stupni. Využívá detailní stochastické simulace zbraňových systémů a podřízených jednotek [7]. Systém je postaven na architektuře distribuované interaktivní simulace (DIS).

DIS je sada protokolů umožňující spojení různých simulátorů k vytvoření společného a konzistentního simulovaného prostředí. Díky tomuto prostředí je zajištěna správná interakce mezi entitami simulovaných v těchto odlišných simulátorech. Celé DIS prostředí je ve své podstatě neustále měnící se databáze entit. Každá osoba, vozidlo, střela, počasí, či různé efekty jsou objektem této databáze [13].

5.1 Základní znaky architektury DIS

- Žádný centrální počítač neřídí celé simulované prostředí, odpovědnost za simulaci stavu jednotlivých entit je na samostatných aplikacích, které běží na počítačích propojených do sítě.
- Autonomní simulační aplikace (simulace) rozesílají zprávy o stavu jejich simulovaných entit ostatním simulacím, aby mohly reagovat na všechny pozorovatelné aktivity. Simulace dále udržují rovněž model stavu prostředí.
- Pro sdělování stavových informací o každé entitě ostatním simulacím v síti se používá standardizovaný protokol.
- Změny stavu určité entity jsou oznamovány simulací, která tuto entitu řídí.
- Vnímání událostí, nebo jiných entit určují simulace, které přijímají stavové informace.
- Pro snížení frekvence vysílání stavových zpráv o každé simulované entitě se používá algoritmus pro odhad polohy, který extrapoluje její polohu, orientaci a rychlost. Stejný algoritmus používá vysílající strana i strany přijímající. Teprve až je odchylka takto odhadnuté pozice a orientace od vnitřního modelu entity větší než jistá prahová hodnota, simulace vyšle o své entitě aktualizovanou stavovou informaci [7].

5.2 Semi-Automated Forces (SAF)

Vývoj polo-automatizovaných sil (Semi-Automated Forces, SAF) začal na počátku roku 1987. Byl navržen tak, aby mohl vyhovět vyvíjejícím se požadavkům na vedení bojové činnosti, a dále tuto činnost rozvíjet v distribuovaném interaktivním simulačním prostředí (DIS prostředí). Tyto polo-automatizované síly byly velice rychle aplikovány v DIS prostředí pro podporu výuky vojenského umění. Vojáci pak mohli trénovat v simulátorech s virtuálním bojištěm bez skutečných omezení jako bezpečí, cena a dostupnosti jakýchkoli terénů a prostředí. SAF je velmi komplexní systém, který může vytvářet různé bojové jednotky a generovat je jak na vlastní straně, tak na straně nepřítele [13].

5.3 OTBSAF

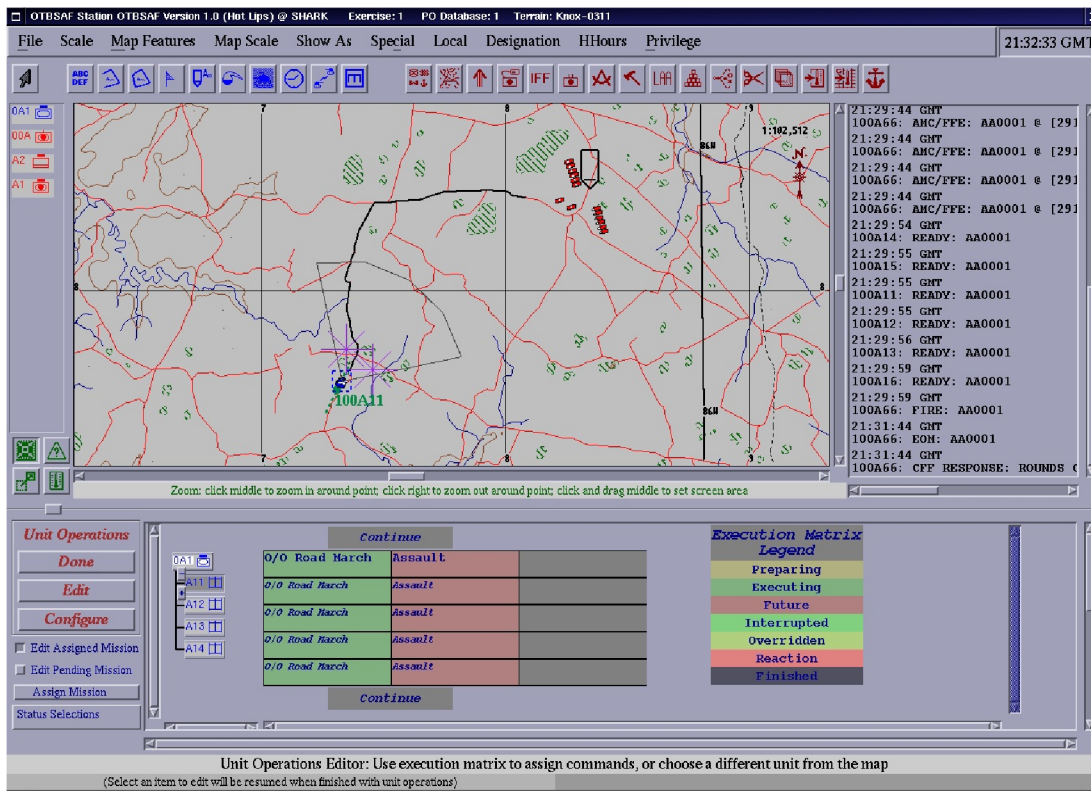
OneSAF Testbed Baseline Semi-Automated Forces (OTBSAF) je otevřená softwarová architektura poskytující jednotnou metodologii a programovou podporu pro vytváření a kontrolování různých druhů entit na virtuálním bojišti. Simulované OTBSAF-jednotky se mohou chovat autonomně – což znamená, že se mohou pohybovat, střílet, detekovat, komunikovat a reagovat bez zásahu jakékoliv obsluhy. Tyto entity mohou také ovlivňovat ostatní entity, včetně entit simulovaných, a to za pomoci virtuálních simulátorů s lidskou posádkou, přes síť za pomoci podpory protokolu DIS, nebo podpory protokolu HLA.

Jednou ze součástí systému OTBSAF je grafické uživatelské rozhraní (GUI). Toto GUI automaticky vytváří uživatelsky přívětivá rozhraní s různou funkčností, poskytuje uživateli možnost modifikovat parametry přímo za běhu a navíc mu dovoluje snadno vytvářet entity, simulovat jejich chování, měnit jejich reakce, sledovat bojiště a stavy jednotek.

Jednotka OTBSAF po vytvoření dostává rozsáhlé schopnosti – např. umí projíždět terénem a vyhýbat se přitom překážkám, útočit na nepřátelské či zvolené objekty, používat svých senzorů, zasílat rádiové zprávy ostatním jednotkám a v neposlední řadě taktéž provádět zadané mise. Cílem systému OTBSAF je věrohodně replikovat vnější chování simulovaných jednotek, jejich zařízení a zbraňových systémů postačujícímu pro trénink a bojový vývoj.

V závislosti na typu simulované entity, OTBSAF poskytuje rozsáhlou množinu realistických akcí, které tato entita může dle pokynů vykonat. Například simulovaný tank může jet podél zadané cesty, pozorovat za pomocí věže a otáčet se na místě. Simulované letadlo může stoupat, kroužit a přistávat. Podobně taktéž zbraňové systémy vykazují realistickou kadenci a trajektorii střel. Všechny entity mohou být dále poškozeny jednou z následujících možností: poškození mobility, poškození palby, poškození mobility i palby, katastrofické poškození. Veškeré zásoby (jako palivo a střely) se vyčerpávají v závislosti na pohybu a střelbě entity.

Pokud se simulují celé jednotky, systém nejenom že vytvoří všechny entity dané jednotky, ale taktéž vytváří strukturu korespondující s hierarchií dané jednotky. Povelů poté mohou být vydávány nejenom jednotlivým entitám, ale taktéž celé struktuře jako celku - od nejvyšší hierarchie až po jednotlivé entity.



Obr. 13: Ukázka simulačního systému OTBSAF

5.4 Moduly architektury OTBSAF

Architektura OTBSAF se skládá z různých modulů, které jsou dále skládány do aplikačních programů. Tyto aplikační programy poté spolu komunikují skrz různé protokoly.

V systému OTBSAF rozeznáváme tyto 4 hlavní aplikační programy [13]:

1. **SAF-stanice:** pracovní stanice poskytující kontrolu nad simulací entit a jednotek. SAF-stanice využívají grafického uživatelského prostředí (GUI) díky kterému mohou jednotliví uživatelé nahrávat scénář boje, vytvářet bojové jednotky, přiřazovat jednotlivým entitám i celým skupinám různé druhy úkolů, či sledovat výsledky simulace. Uživatel dále může využívat různé utility pro práci se simulací jako např. práce s terénem a počasím, grafický editor pro zakreslování taktických značek, práce se šablonami a vrstvami.

2. **SAF-simulátory:** simulují kolektivní a individuální chování jednotek, dále jejich fyzikální subsystém a vzájemnou interakci na bojišti. SAF-simulátory udržují stav bojiště v objektově orientované databázi. Tato databáze obsahuje lokální jednotky (aktualizované pomocí SAF-stanice) a vzdálené jednotky (aktualizované pomocí jiných SAF-stanic připojených v síti). SAF-stanice dokáže modifikovat pouze své vlastní lokální simulované entity a dále jejich stav posílat na síť. Odpovědnost za správnou simulaci a modifikaci vzdálených entit na stanicích přebírá každý ze SAF-simulátorů, který musí správně modifikovat stav udržující databáze. Tímto je dán předpoklad pro distribuovanou simulaci.
3. **SAF-komplet:** Obvyklá konfigurace architektury OTBSAF vyžaduje jednu SAF-stanici pro zpracování uživatelského rozhraní plus jeden počítač se SAF-simulátorem pro zpracování vlastní simulace. Avšak je taktéž možné nakonfigurovat daný systém tak, aby na jednom jediném počítači běželo zpracování uživatelského rozhraní i zpracování simulace zároveň. Této modifikaci se říká SAF-komplet a využívá se při malých cvičeních.
4. **SAF-záznamník:** slouží k zaznamenávání všech událostí během cvičení a následném přehrávání těchto událostí kdykoli později. SAF-záznamník dokáže ukládat jak chování entity, tak i fyzikální stavy této entity během celé simulace. Záznamník taktéž zaznamenává veškeré pakety, které byly v souvislosti se simulací posílány na síti. Následné záznamy slouží k možnosti restartovat a připojit se k již proběhlé simulaci a taktéž záznamy slouží k vyhodnocení a analýze proběhlého cvičení.

5.5 OTBSAF architektura

OTBSAF softwarová architektura je flexibilní, hierarchická a podpůrná. Je flexibilní tím, že dovoluje vytvořit novou reprezentaci modelu chování a začlenit tento model do stávajícího systému. Navíc hierarchie této architektury poskytuje vývojářům snadnější přidávání, odebrání, modifikaci a testování kterékoli jednotlivé funkce. Taktéž poskytuje uživatelské rozhraní ke kontrole a modifikaci všech vytvářených modelů [13].

OTBSAF architektura se skládá z následujících částí:

5.5.1 Kolektivní chování

Příkladem kolektivního chování v simulaci je např. situace, kdy skupina vozidel napadá cíl, čímž dochází ke spuštění sekvencí akcí na celé skupině těchto vozidel. Ačkoli je možné tyto akce simulovat a naplánovat za pomoci individuálního chování, je jednodušší využít kolektivního chování.

5.5.2 Individuální chování

V systému OTBSAF jsou simulované entity zřetelně od sebe odlišeny, čímž můžeme v simulátoru vytvářet jednotlivé tanky, vojáky, letadla, apod. a přiřazovat těmto entitám rozdílné chování.

5.5.3 Fyzický subsystém

Jelikož se entity navzájem v simulačním prostředí ovlivňují, systém OTBSAF nejenom že simuluje chování těchto entit, ale taktéž vhodně modeluje fyzický efekt tohoto chování na ostatní entity. Do simulace fyzického subsystému patří např.: kinematika (pohyb) vozidel a osob, řízení střel a zbraňových systémů, řízení senzorů, komunikace, apod.

5.6 Metodologie návrhu

Aby bylo možno snadno nahradit a vylepšit některý ze subsystémů softwarových modulů využívá architektura OTBSAF tyto 4 techniky [13]:

- Vrstvy
- Objektově orientované programování
- Přesnou specifikaci rozhraní
- Datově řízené spouštění

Následující podkapitoly popisují tyto techniky podrobněji.

5.6.1 Vrstvy

Vrstvy jsou jedním z návrhu, kdy subsystémy softwarových modulů jsou seskupeny do funkčně podobných skupin (vrstev). Programování je poté jednodušší a snadněji ovladatelné, protože software vytvářený v jedné vrstvě může využívat pouze funkce a služby které jsou dostupné jen v nižších vrstvách. Navíc využití vrstev v architektuře OTBSAF dovoluje nahrazení subsystému na různých úrovních a svým návrhem taktéž dovoluje lepší testování implementovaných změn.

Následující schéma ukazuje využití vrstev při reprezentaci vzdálených entit v rámci systému OTBSAF (v závorce jsou uvedeny názvy knihoven implementující danou vrstvu).



Obr. 14: Schéma vrstev implementující reprezentaci vzdálených entit [13]

5.6.2 Objektově orientované programování

Objektově orientované programování zapouzdřuje subsystém, nebo moduly knihoven do objektových tříd, které jsou definovány strukturou dat a souborem funkcí které s těmito daty pracují. Datové struktury jednotlivých objektů jsou rozděleny na veřejné a privátní, tak aby minimalizovaly vzájemnou závislost mezi moduly a zjednodušily pozdější změnu těchto modulů.

Tradiční koncepce objektově orientovaného programování vytváří nové objekty z dřívějších objektů za pomoci jednoduché dědičnosti. Tato koncepce je v systému OTBSAF zajištěna za pomoci kompozice (složení), kdy větší třídy jsou skládány z menších podtříd. V architektuře OTBSAF spolu přesně korespondují objektové třídy a knihovny.

5.6.3 Přesná specifikace rozhraní

Aby bylo možno v architektuře OTBSAF nahrazovat již vytvořené moduly novými, je zcela přesně definováno rozhraní (v hlavičkových souborech zdrojových kódů) mezi objekty daných tříd (knihovny). Taktéž je dané rozhraní velice dobře dokumentováno a vysvětleno, co ostatní knihovny od modulu očekávají.

5.6.4 Datově řízené spouštění

Většina modulů architektury OTBSAF jsou datově řízené; jejich implementace v systému je velice obecná a široká. Detailní specifikace mnoha objektů se určí až při vytváření samotného objektu za

pomocí parametrů získaných z datových souborů (tyto soubory jsou obvykle označovány jako konfigurační soubory a parametrické soubory). Různé parametry v datových souborech specifikují charakteristiku a chování daného objektu a dovolují kdykoli změnit tyto charakteristiky aniž by se musel překládat celý systém OTBSAF.

5.7 Architektura řízení a ovládání

Systém OTBSAF v sobě zahrnuje taktéž architekturu pro reprezentaci řízení a ovládání entit, která velice dobře simuluje lidské chování a vojenské velení. Tato architektura sestává z definic tříd pro tzv. trvalé objekty (Persistent Object - PO) a algoritmů, které popisují jak tyto objekty jsou uživatelským rozhraním a simulací obsluhovány.

Architektura řízení a ovládání zajišťuje a poskytuje následující možnosti [13]:

- Zajišťuje, že uživatel může vytvářet libovolně komplexní mise, které mohou zahrnovat předběžně naplánované nepředvídané situace.
- Zajišťuje, že uživatel může postupně měnit naplánovanou misi, a to i v jejím průběhu.
- Zajišťuje, že uživatel může jakkoli změnit volbu, kterou učinila simulovaná entita.
- Poskytuje způsob, kterým můžeme vyjádřit typ mise, kterou entity a jednotky vykonávají.
- Poskytuje možnost jak reprezentovat neurčitost a nejasnost v simulovaném bojišti
- Poskytuje uživatelskému rozhraní strukturu, která uživateli jasně popíše a vysvětlí chování dané jednotky.
- Poskytuje přímé mapování mezi architekturou systému a uživatelským rozhraním.
- Poskytuje prostředky k uložení aktuálního stavu simulace, která může být později znovu nahrána a spuštěna.

Základem architektury řízení a ovládání je koncepce „úkolů“. Úkol je chování, nebo reakce vykonávána entitou, či jednotkou v simulovaném prostředí bojiště. Úkoly mohou:

- přímo ovládat fyzický subsystém (např. řídit auto směrem k cílovému bodu)
- vykonávat nějaký plán (např. napadnout cíl)
- být nezávislé na misi a nepřetržitě trvat (prohledávání okolí)
- reagovat na aktuální stav boje a bojiště (např. reakce psychologickým efektem při dopadu granátů)

Úkoly jsou reprezentovány v „PO databázi“ (Persistent Object database) třemi hlavními znaky:

1. **Identifikátor úkolu:** jedinečné číslo určující kód, který se bude vykonávat po spuštění úkolu

2. **Parametry úkolu:** vyjadřují různé volby a možnosti dostupné při provádění úkolu
3. **Stavové informace úkolu:** tyto informace jsou využity uživatelským rozhraním a samotným úkolem v době jeho běhu.

Aby bylo možno sdílet stavové informace, chování a informace o řízení a ovládání, byl vyvinut a použit „PO protokol“. Díky tomuto protokolu mají moduly systému OTBSAF přístup ke všem těmto informacím. PO protokol umožňuje nejen efektivně sdílet velké množství informací, ale také dovoluje nám tyto informace dále měnit. Tento mechanismus je zajištěn díky pravidelnému rozesílání paketů (PDU), které mohou popisovat objekt, vymazat objekt, žádat informace o objektu, apod. Důsledkem této koncepce sdílení informací přes PO databázi je, že jakákoli změna na lokální, či vzdálené stanici se jeví aplikacím jako identická [13].

5.8 Základní knihovny architektury OTBSAF

Architektura OTBSAF v sobě zahrnuje množství základních knihoven a modulů, které poskytují podporu při vytváření nových knihoven a modulů. Níže uvádím popis 4 knihoven, které jsem nejvíce využil právě při vytváření modelu psychologických efektů a modelu generování suti při bojové činnosti dělostřelectva.

5.8.1 Řízení činností a úkolů

Jednou z velmi důležitých knihoven systému OTBSAF je knihovna „libtask“, která řídí a spojuje různé knihovny řešící určité činnosti a úkoly k jednotlivým entitám a jednotkám. Jakmile je inicializována některá z knihoven úkolů, vysílá zprávu o připravenosti právě knihovně „libtask“, která převezme informace a zařídí její spuštění ve správné chvíli. „libtask“ poskytuje různé pomocné funkce, které mohou využívat knihovny úkolů a taktéž funkce které využívá generátor kódu rozšířeným asynchronním konečným automatem (AAFMSM) [14].

Rozšířený asynchronní konečný automat je využíván při implementaci chování a reakcí entit a jednotek. Úkoly chování a reakcí jsou asynchronní, protože své výstupy generují na reakce různých událostí, které nastávají v simulovaném prostředí. Zajímavostí je, že úkoly ovlivňují a používají mnoho jiných proměnných, než svých vlastních. Implementace konečného automatu se neděje za použití standardních „switch - case“ konstrukcí, ale definuje se pro stavový stroj přechodová tabulka stavů.

Utilita preprocesoru (fsm2ch) čte AAFSM formát a překládá jej do jazyka C, který poté může být zkompilován. Navíc preprocesor může z tohoto formátu generovat schéma přechodů ve formátu „.fig“ [13].

5.8.2 Práce s terénní databází

Knihovna „libctdb“ (The Compact Terrain Database) je využita aplikacemi k přístupu k datům a informacím o použitém terénu, nadmořské výšce a jiných vlastnostech mapy, která je uložena ve formě databázového typu. Aplikace mohou díky „libctdb“ nahrát tuto databázi do paměti, a poté použít funkce „libctdb“ k přístupu na tento terén. Funkce „libctdb“ zahrnují [15]:

- Načtení databáze do paměti, nebo cache.
- Poskytují užitečné informace o databázi, jako je velikost této databáze, minimální a maximální nadmořské výšce terénu, UTM zóna, apod.
- Zjištění hodnoty nadmořské výšky v daném bodě nebo po směru dané úsečky
- Zjištění typu půdy
- Rozmístění entit
- Kalkulace viditelnosti
- Kalkulace radarového rušení
- Vytváření grafické reprezentace terénu v reálném čase (vrstevnicové mapy, výškové mapy, apod.)

5.8.3 Zpracování paketů PDU

Aplikační programy nemusí znát přesně které bity a jak komunikují na síti, poněvadž knihovna „libPDUAPI“ poskytuje abstrakci PDU struktury. Knihovna sestává ze skupin datových soborů, které definují standardizované protokoly a poskytuje aplikačním programům funkce ke čtení a interpretaci těchto struktur. Uvnitř aplikací je definice datových struktur PDU privátní, což znamená, že nemusí být deklarovány ve veřejném rozhraní a tím redukuje komplexnost celého systému [16].

5.8.4 Čtení konfiguračních souborů

Konfigurační soubory (soubory typu .rdr) jsou čteny za pomoci knihovny „libreader“. Tato knihovna zprostředkovává čtení datových souborů a jejich následnou konverzi do struktur jazyka C. Syntaxe konfiguračního souboru dovoluje načítat celá čísla (v desítkové, hexadecimální, nebo osmičkové soustavě), čísla s pohyblivou řádovou čárkou a řetězce [17].

6 Návrh modelu psychologických efektů při bojové činnosti dělostřelectva

Psychologické efekty v konstruktivní simulaci patří k tzv. měkkým faktorům ozbrojeného zápasu, které mají významný vliv na jeho příběh i konečný výsledek. Projevují se v charakteristickém spontánním nebo cílevědomém chování jednotlivců, či skupin při plnění úloh v reálné bojové činnosti [2].

Níže je uveden návrh a implementace modelu psychologických efektů do systému OTBSAF při bojové činnosti dělostřelectva.

6.1 Neformální specifikace

Psychologické efekty se projeví při dopadu granátu blízko entity, a to snížením vycvičenosti entity o 50% na dobu 5 minut. Zároveň nepovolí jednotkám dosáhnout větší rychlosti pohybu než 10 km/h u vozidel a 3 km/h u osob a zvířat. Po uplynutí dané doby se vycvičenost vrací na původní hodnotu a entity se opět mohou pohybovat rychlostí, kterou umožní terén a takticko-technická data (TTD) v závislosti na úkolu. Při dopadu více granátů tak, že všechny ovlivňují vycvičenost dochází k součtu snížení vycvičenosti. Doba se ale prodlužuje o polovinu tedy o 2,5 minuty.

Efekt nastane pokud dělostřelecký granát ráže 152, nebo 155 mm dopadne:

- do 50 m od entity vždy nastane (tj. ve 100% případech, pravděpodobnost 1);
- do 75 m od entity 80%
- do 100 m od entity 60%
- do 150 m od entity 40%

V případě minometných granátů je efekt poloviční (poloviční pravděpodobnost toho, že efekt nastane). U raketometné munice je naopak navýšení o 20%. V případě leteckých pum a raket je navýšení o 20% a prodloužení intervalů ve kterých efekt nastává s danou pravděpodobností o 25 metrů. Pokud hodnota toho, že efekt proběhne, vyjde větší než 100%, zůstává na 100%

Příklad: v okolí BVP s vycvičeností 60% dopadnou dva dělostřelecké granáty, které jsou v pásmu ovlivňujícím vycvičenost (mají psychologický efekt). Po dopadu prvního se sníží vycvičenost na 30%. Za 5 minut by se zvýšila na původní hodnotu. Po půl minutě od dopadu prvního bude v nebezpečném prostoru druhá exploze. V tuto dobu dojde ke snížení vycvičenosti o další 1/2 na 15%

a prodloužení doby návratu do původního stavu o 2,5 minuty na 7 minut. Pokud do této doby nebude vozidlo zničeno či ohroženo dalším výbuchem navrátí se hodnoty vycvičenosti po sedmi minutách na 60%.

6.2 Návrh modelu

Při návrhu a implementaci modelu do systému OTBSAF máme dva různé pohledy na začlenění a funkci tohoto modelu.

1. Model chování je specifický pro entitu a jeho běh je závislý na vytvořené entitě.
2. Model chování není specifický pro entitu a běh je v rámci celého globálního systému od začátku spuštění.

Model psychologických efektů je závislý na vytvořené entitě a probíhá pouze tehdy, když je v simulačním systému OTBSAF přítomna minimálně jedna entita.

6.2.1 Spuštění běhu modelu

Model chování psychologických efektů je přiřazen pouze některým entitám, a to vozidlům, lidem a zvířatům. Model se spouští na pozadí hned po vložení entity do systému OTBSAF. Při spuštění modelu se předávají argumenty, které ovlivňují nastavení a celkový běh modelu chování.

Parametry zadávané při spuštění modelu:

1. `in_background` – parametr definuje zda model běží na pozadí
2. `duration_time` – délka trvání psychoefektu při první reakci na dopad střely
3. `extend_time` – délka o kterou se prodlužuje psychoefekt při další reakci na dopad střely
4. `limit_speed_lifeform` – maximální povolená rychlost u lidí a zvířat při psychologické reakci
5. `limit_speed_vehicle` – maximální povolená rychlost u vozidel při psychologické reakci

Výše uvedené parametry se mohou kdykoli změnit dle požadavků neformální specifikace, a to bez zásahu do implementace modelu.

6.2.2 Konfigurace modelu

Z neformální specifikace je patrné, že model chování reaguje různě podle typů střel. Proto je vytvořen konfigurační soubor, který se načítá při nahrávání modelu a obsahuje informace o jednotlivých typech střel, jejich účinnosti a pravděpodobnosti, že psychologický efekt nastane.

Formát konfiguračního souboru:

```
(<název munice>
  (range <vzdálenost1> <vzdálenost2> <vzdálenost3>
    <vzdálenost4>)
  (prob <pravděpodobnost1> <pravděpodobnost2> <pravděpodobnost3>
    <pravděpodobnost4>)
)
```

Příklad konfiguračního souboru uvádím v příloze 1.

Do konfiguračního souboru se mohou kdykoli nové typy munic doplnit a tím rozšířit působení psychologického efektu i na jiné (další) typy munic.

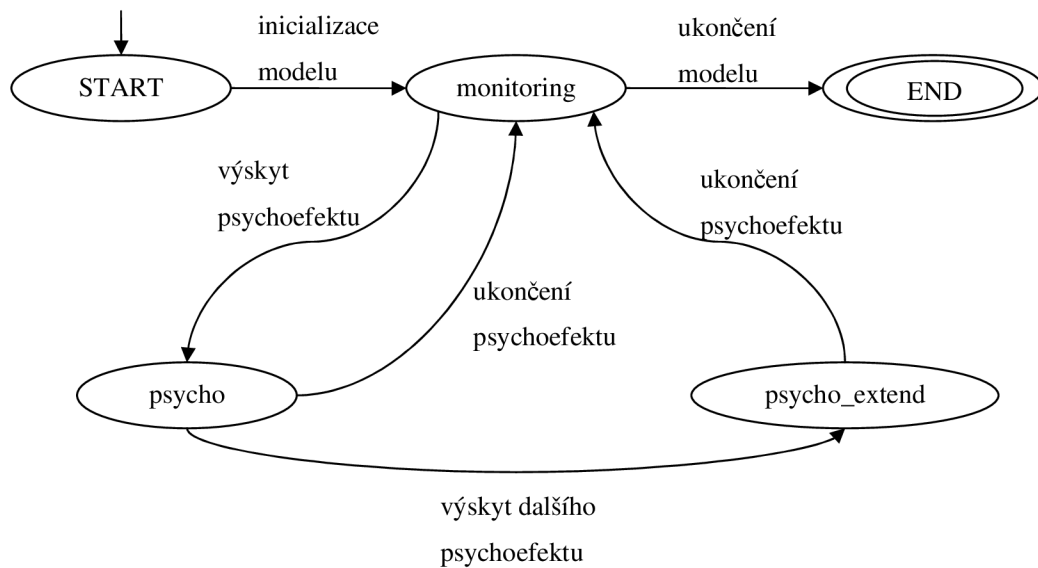
Pokud dopadající munice není specifikována v konfiguračním souboru, psychologický efekt pro tuto municu nenastává.

6.2.3 Stavy modelu

Model psychologických efektů je ve své podstatě navrhnout jako stavový stroj (konečný automat), který dle stavu, ve kterém se právě nachází, ovlivňuje danou entitu. Jelikož každá entita má svůj vlastní model, může se kterákoliv entita nacházet v různých stavech vzhledem k jiné entitě používající daný model.

6.2.3.1 Formální specifikace modelu pomocí konečného automatu

$$M = (Q, \Sigma, \delta, q_0, F)$$
$$Q = \{\text{START, END, monitoring, psycho, psycho_extend}\}$$
$$\Sigma = \{\text{inicializace modelu, ukončení modelu, výskyt psychoefektu, ukončení psychoefektu}\}$$
$$\delta: \delta(\text{START, inicializace modelu}) = \{\text{monitoring}\}$$
$$\delta(\text{monitoring, výskyt psychoefektu}) = \{\text{psycho}\}$$
$$\delta(\text{monitoring, ukončení modelu}) = \{\text{END}\}$$
$$\delta(\text{psycho, výskyt psychoefektu}) = \{\text{psycho_extend}\}$$
$$\delta(\text{psycho, ukončení psychoefektu}) = \{\text{monitoring}\}$$
$$\delta(\text{psycho_extend, ukončení psychoefektu}) = \{\text{monitoring}\}$$
$$q_0 = \text{START}$$
$$F = \{\text{END}\}$$



Obr. 15: Grafická reprezentace modelu

6.2.3.2 Popis jednotlivých stavů modelu

START – počáteční stav ve kterém se model nachází po spuštění simulace

Monitoring – stav ve kterém se nacházíme, pokud entita nemá aktuálně psychologickou reakci. V tomto stavu čekáme na nepřímou palbu a případné potvrzení psychoefektu pro danou entitu. Pokud jsou podmínky splněny, přecházíme do stavu „psycho“.

Psycho – stav psychologické reakce na nepřímou palbu. Entita má v tomto stavu sníženou vycvičenost a sníženou rychlost pohybu. Pokud v zadaný čas nepřijde další nepřímá palba a nenastává psychologický efekt znovu, entita přechází do stavu „monitoring“, pokud přichází další psychologická reakce, přecházíme do stavu „psycho_extend“.

Psycho_extend – stav prodloužené psychologické reakce, kdy narůstá doba trvání psychologického efektu a ještě víc se snižuje vycvičenost entity. Pokud v tomto stavu dochází k dalším reakcím, prodlužuje se pouze doba trvání psychologického efektu. Z tohoto stavu se po uplynutí doby reakce na dopad střelby přechází do stavu „monitoring“.

END – stav ukončující běh modelu, nastává při ukončení simulace

6.3 Implementace modelu psychologických efektů

Model psychologických efektů je implementován v systému OTBSAF jako jedna knihovna pod názvem „libvpsychoeffect“. Tato knihovna obsahuje 11 souborů, ve kterých je rozdělena a zahrnuta implementace daného modelu.

6.3.1 Popis jednotlivých souborů knihovny

1. GNUmakefile.in – soubor slouží pro překlad knihovny
2. libpe_local.h – hlavičkový soubor obsahující deklarace funkcí, struktur a proměnných potřebných v rámci běhu modelu.
3. libvpsychoeffect.h – hlavičkový soubor nutný k propojení knihovny s celým systémem OTBSAF
4. libvpsychoeffect.texinfo – soubor popisující implementovanou knihovnu v jazyku LaTeX
5. pe_class.c – zdrojový soubor definující funkce pro správné spuštění a napojení běžícího modelu na simulační systém OTBSAF
6. pe_config.rdr – konfigurační soubor (popsán v kapitole 6.2.2)
7. pe_init.c – zdrojový soubor sloužící k počáteční inicializaci modelu
8. pe_params.c – zdrojový soubor zahrnuje funkce nutné pro načtení počátečních parametrů nutných k spuštění modelu
9. pe_task.fsm – soubor se zvláštní syntaxí implementující stavový stroj (konečný automat) daného modelu (viz. kapitola 6.2.3)
10. pe_util.c – zdrojový soubor obsahující definice podpůrných funkcí využívající model
11. vpsychoeffect.rdr – konfigurační soubor popisující editor pro model psychologických efektů

6.3.2 Načítání konfiguračních informací

Pro uložení informací o municích využívám struktury s následujícími informacemi:

1. Typ munice.
2. Příslušné rozsahy účinnosti psychologického efektu.
3. Pravděpodobnosti výskytu efektu k danému rozsahu.

Jelikož na počátku neznáme počty načítaných munic, alokujeme dynamicky až při načítání informací z konfiguračního souboru počet těchto struktur. Při načítání příslušných rozsahů účinnosti psychologického efektu si taktéž zaznamenáme maximální rozsah, pro který budeme počítat psychologický efekt.

6.3.3 Běh modelu

Model psychologických efektů sestává ze čtyř různých fází, které jsou ve své podstatě stavovým strojem a mezi kterými se přechází dle různých vnějších i vnitřních vlivů působících na entitu.

6.3.3.1 Inicializační fáze

Na počátku se nacházíme ve stavu „START“, kde provedeme nezbytnou inicializaci. Inicializace spočívá v zaregistrování funkce, která nám bude přijímat pakety PDU ze sítě i lokální stanice s právě probíhající nepřímou palbou. Paket PDU v sobě obsahuje informaci o typu munice a souřadnice, kde daná munice dopadla. Jelikož efekt se projevuje jak u vozidel tak u osob, či zvířat, musíme také při inicializaci zjistit, na jakém typu podvozku se příslušná entita nachází, abychom při nastavování rychlostních limitů ovládali příslušný podvozek, který určuje pohyb entity.

6.3.3.2 Stav vyhodnocující psychologický efekt

Z počátečního stavu „START“ přecházíme do stavu „monitoring“. V tomto stavu čekáme na nepřímou palbu. Tuto palbu nám potvrdí příznak, který nastavuje funkce, kterou jsme si zaregistrovali pro příjem PDU z nepřímé palby. Jakmile je příznak nastaven, zjistíme, zda psychoefekt nastane, či nenastane. O testování, zda efekt nastává, se stará funkce, která prohledá seznam všech zaregistrovaných munic a zjistí, zda dopadlá munice se nachází v seznamu. Pokud se tato munice v seznamu nachází, funkce zjistí v jaké vzdálenosti od entity dopadla a vybere příslušnou pravděpodobnost výskytu efektu pro danou vzdálenost. Nakonec funkce generuje náhodné číslo, které dle výskytu pravděpodobnosti efektu nám udává, zda psychologický efekt u entity nastává. Pokud by funkce nenašla dopadlou munic v seznamu, případně by generované číslo nebylo v platnosti pravděpodobnosti výskytu - efekt nenastává a entita se stále nachází ve stavu „monitoring“.

6.3.3.3 Stav psychologického efektu

Jakmile nastane psychoefekt, entita přechází ze stavu „monitoring“ do stavu „psycho“. Při prvním běhu (tiku) ve stavu „psycho“ si musíme uložit původní vycvičenost entity, abychom tuto vycvičenost po skončení psychologického efektu nastavili správně zpět. Po uložení původní vycvičenosti snižujeme danou vycvičenost o příslušný faktor. Tento faktor je v programu uložen jako konstanta, která se může kdykoli změnit dle neformální specifikace. Pokud se tento faktor změní, musíme knihovnu zabezpečující model přeložit, aby se nám změna projevila v systému.

Po snížení vycvičenosti musíme dále nastavit rychlostní limit. Jelikož limit pro vozidla je jiný než limit pro vojáky a zvířata, zjistíme si za pomocí funkce z knihovny „libentity“ na které entitě daný model běží. Po správném zjištění nastavujeme rychlostní limit z parametru předaného při spuštění modelu (viz. kapitola 6.2.1).

Entita setrvává ve stavu „psycho“ požadovanou dobu (dle nastavení při spuštění modelu), po uplynutí dané doby se vycvičenost vrací na původní hodnotu, kterou jsme záložovali při prvním

provedení tohoto stavu a dále je entitě odebrán rychlostní limit, takže se může pohybovat dle konkrétních podmínek systému OTBSAF. Nakonec opět přecházíme do stavu „monitoring“ (popsán výše) a čekáme případně na další reakci.

Pokud se však vyskytne ve stavu „psycho“ další nepřímá palba a vyhodnotí se nový výskyt psychologického efektu, přecházíme do stavu „psycho_extend“.

6.3.3.4 Prodloužení psychologického efektu

Stav „psycho_extend“ slouží k prodloužení psychoefektu, pokud nastává další psychologický efekt v rámci již trvajících psychologického efektu. Při prvním běhu (tiku) snižujeme vycvičenost entity o další faktor, avšak již nenastavujeme rychlostní limit, poněvadž ten je nastaven z předcházejícího stavu („psycho“).

Pokud se entita nachází ve stavu „psycho_extend“ a přijde na ni nová psychologická reakce, již se nesnižuje vycvičenost entity, ale pouze se prodlužuje čas setrvání v tomto stavu a tím pádem se prodlouží psychologický efekt působící na entitu.

Po uplynutí psychoefektu, vracíme vycvičenost na původní hodnotu, kterou jsme si uložili ve stavu „psycho,“ a resetujeme hodnotu rychlostního limitu. Jakmile vše vrátíme na původní hodnoty, přecházíme do stavu „monitoring“ a čekáme na nový psychologický efekt.

6.3.4 Výpis informací o psychologickém efektu

Při ovládání entity je nutné, aby operátor byl informován o tom, že daná entita má psychologický efekt a nemůže naplno plnit předepsané úkoly. Informace o právě probíhajícím psychologickém efektu se proto vypisují ve stavovém okně entity. O vypisování informací se stará funkce „status,“ která se nachází v inicializačním souboru „pe_init.c“.

Funkce „status“ obsahuje programovou konstrukci přepínače „switch,“ který dle stavu, ve kterém se aktuálně nachází model psychologických efektů, vypisuje požadované informace. Při vypisování je použita funkce „edt_format“ z knihovny „libeditor,“ která dokáže správně zobrazit informaci (např. rychlost) ve formátu, který je aktuálně zvolen v celém systému OTBSAF.

```
Status for Unit 100A11 [IC AK47] (dismounted):  
Psycho Effects: strength at 25 %, maximum speed is limit to 3 kph
```

Obr. 16: Ukázka výpisu informací o psychologickém efektu

7 Návrh modelu generování sutí a klád při bojové činnosti dělostřelectva

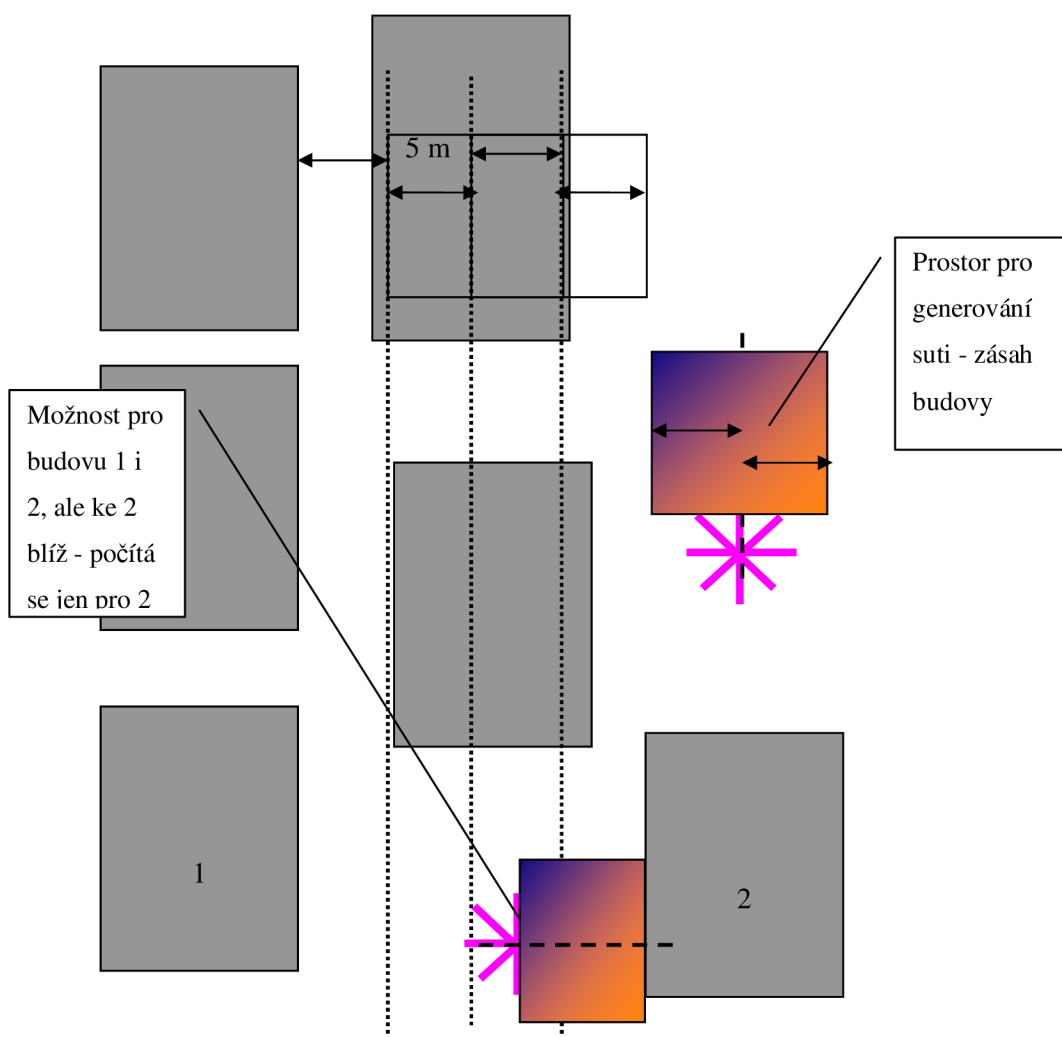
V systému OTBSAF je bojová činnost dělostřelectva implementována jako základní model, kde operátor může libovolně střílet z daných prostředků předepsanou municí a po dopadu střely se nám na zadaných souřadnicích vypočítává poškození pro jednotlivé druhy techniky a živé síly. Tento zjednodušený model nijak nepočítá s terénem a ostatními entitami, které se mohou v systému vyskytovat. Proto byl dán z Centra simulačních a trenažerových technik požadavek, aby byl implementován nový a další model, který by se více přibližoval dělostřeleckým účinkům v reálném prostředí.

Tento nový model bere v úvahu i terén, na který střela dopadne a ostatní entity vyskytující se v simulátoru. Jeho hlavním účelem je vytvářet suť, popř. klády, které se v reálném světě vyskytují právě při bojové činnosti dělostřelectva.

7.1 Neformální specifikace

V případě zásahu budovy na terénní databázi dojde k vygenerování sutí následujícím způsobem:

- přímý zásah budovy – vygeneruje 5 entit (max 3 stejné) ve 100% případech
- zásah do 5 m od budovy – vygeneruje 3 entity v 75% případech
- zásah do 10 m od budovy – vygeneruje 1 entitu ve 40% případech
- zásah do 15 m od budovy – vygeneruje 1 entitu v 15% případech



Obr. 17: Příklad generování suti při různém dopadu střel v okolí budov

Pravidla pro generování entit suti:

1. Entity se generují mezi místem zásahu a okrajem budovy v pásmu širokém 10 metrů (platí pro těžiště entit), 5 metrů na každou stranu od osy mezi místem zásahu a nejbližší budovou.
2. V případě přímého zásahu mezi nejbližším okrajem budovy a vzdáleností 10 metrů od ní, opět v šířce pásma 10 m.
3. Pravděpodobnost umístění v celém prostoru je rovnoměrná – rovnoměrné rozložení, suti bude lépe zaplňovat ulice a zneprůjezdnovat je.
4. Pokud granát zasáhne vozidlo (přímý zásah), ne pěchotu a entity typu jehlan apod, efekt generování suti pro budovu (to že efekt nastane) se snižuje na 50% původní hodnoty.
5. Efekt se počítá vždy pro nejbližší budovu, i když by mohl granát poškodit více budov.

6. Natočení entit je náhodné.
7. Pokud je ve vzdálenosti do 10 m od výbuchu 6 – 10 z dále uvedených entit, snižuje se pravděpodobnost efektu o 50% (takže ze 40% bude 20%).
8. Pokud je ve vzdálenosti do 10 m od výbuchu 11 – 15 z dále uvedených entit, snižuje se pravděpodobnost efektu o 75% (takže ze 40% bude 10%).
9. Při více než 15 entitách do vzdálenosti 10 m od výbuchu efekt nenastane, stejně tak efekt nenastane pokud je do vzdálenosti 50 metrů od výbuchu více než 50 entit.
10. Při více než 100 vygenerovaných entitách na stanici, efekt nenastává vůbec (součet sutí i klád).

Překážky které jsou generovány jako suť a pravděpodobnost jejich výskytu. V tabulce je uvedeno i nastavení průchodnosti překážky danou skupinou entit.

Entita	Výskyt na	Pásový	kolový - OT	kolový - nákladní	kolový - osobní	Pěší
Hromada drátů	10%	0	1	1	0	0
Zborcená zeď	15%	0	0	0	0	1
Skruž	10%	1	0	0	0	1
Skříň	10%	1	1	1	0	1
Stavební suť M	10%	1	1	0	0	1
Stavební suť V	15%	0	0	0	0	0
Traverza	10%	1	1	0	0	1
Zlomený panel	10%	0	0	0	0	1
Tram	10%	0	0	0	0	0

Tab. 1: Pravděpodobnost generování různých entit vystupující jako suť

Výše zmíněné požadavky byly určeny pro municí ráže 152 a 155 mm, pokud dojde k zásahu jinou municí, použijí se následující pravidla:

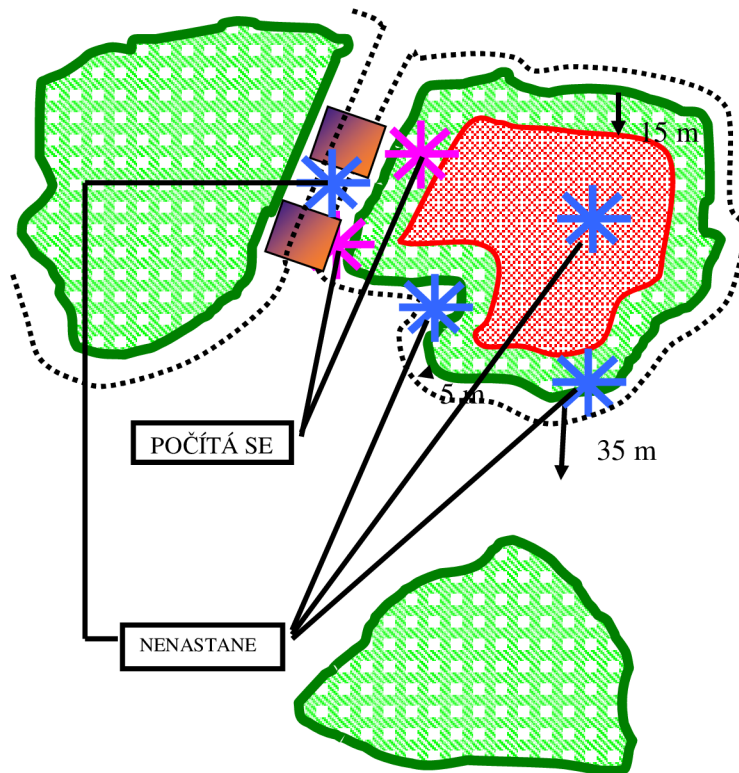
- minometná munice má vliv jen při přímém zásahu a při zásahu do 5m. Zároveň se pravděpodobnost projevení efektu redukuje na 50% respektive 35%.
- raketometná munice má stejný efekt jako dělostřelecká munice
- letecké bomby a rakety zvětšují všechny intervaly o 5 metrů a pravděpodobnost efektu je plus 25% (100, 100, 65, 40)

V případě zásahu lesa na terénní databázi dojde k vygenerování klády následujícím způsobem:

- zásah neprůchodného lesního porostu do 15 metrů od okraje způsobí vygenerování klády v prostoru mezi okrajem lesa a vzdáleností 10 m od okraje (mimo plochu lesa) v 25% případů
- zásah vedle lesa do 5 metrů do okraje způsobí vygenerování klády v prostoru mezi okrajem lesa a vzdáleností 10 m od okraje (mimo plochu lesa) v 10% případů
- zásah do lesa ve vzdálenosti nad 15 metrů a mimo les nad 5 metrů od okraje nezpůsobí nic

Pravidla pro generování klád:

- efekt nenastane pokud není do vzdálenosti 35 metrů od okraje lesa další lesní plocha
- pravděpodobnost efektu se sníží na polovinu pokud ve vzdálenosti 25 metrů od výbuchu je více než 5 klád
- pokud je ve vzdálenosti do 25 metrů od výbuchu více než 10 klád efekt nenastává
- pokud je do vzdálenosti 100 metrů od výbuchu víc než 25 klád efekt nenastane
- efekt nenastane pokud je na stanici více než 100 generovaných entit (součet suti i klád)
- platí to samé co pro sut v bodě 10)
- pokud jsou dva lesní porosty pro které by mohlo dojít k efektu počítá se jen pro bližší



Obr. 18: Příklad generování klád při různém dopadu střel do lesa

7.2 Návrh modelu

Model se ve své podstatě skládá ze dvou různých modelů:

1. Model generování klád při dopadu dělostřelecké munice do lesa, či na jeho okraj.
2. Model generování suti při dopadu dělostřelecké munice na budovu, či v blízkosti budovy.

V rámci začlenění modelů do systému OTBSAF můžeme různé modely rozdělit na:

1. Model je specifický pro entitu a jeho běh je závislý na vytvořené entitě.
2. Model není specifický pro entitu a běh je v rámci celého globálního systému od začátku spuštění.

Oba dva modely působí v rámci celého systému OTBSAF, nejsou žádným způsobem přiřazeny ke kterékoli živé/neživé entitě a jejich vliv a uplatnění se projevuje během celého běhu systému.

Modely se spouští zároveň se spuštěním celého systému OTBSAF a běží na pozadí až do ukončení systému.

7.2.1 Konfigurace modelu

Celý model generování suti a klád je velmi komplexní a dle neformální specifikace je patrné, že v těchto modelech vystupuje mnoho různých proměnných, které ovlivňují daný model. Např. různé účinky dělostřeleckých granátů, různé pravděpodobnosti výskytu suti, různé výskytu suti závislé na vzdálenosti dopadu střely od budovy, aj.

Z těchto důvodů je vytvořen konfigurační soubor, který se načítá již při nahrávání modelu a obsahuje různé typy informací.

7.2.2 Formát konfiguračního souboru

Konfigurační soubor je rozdělen do těchto částí:

1. Účinky jednotlivých munic při generování suti.
2. Entity, které se generují jako suť a jejich pravděpodobnost výskytu.
3. Entita, která se generuje jako kláda.
4. Koeficienty pro snižování pravděpodobnosti efektu vzhledem k již generovaným entitám.
5. Podmínky generování klád.
6. Entity vystupující jako budovy.

Formát jednotlivých částí je následující:

7.2.2.1 Účinky munic

```
(munition_rubble
 (<název munice>
   (range      <vzdálenost1>      <vzdálenost2>
             <vzdálenost3>      <vzdálenost4>)
   (prob       <pravděpodobnost1> <pravděpodobnost2>
             <pravděpodobnost3> <pravděpodobnost4>)
   (num_entity <počet gen. entit1> <počet gen. entit2>
             <počet gen. entit3> <počet gen. entit4>)
 )
)
```

7.2.2.2 Entity generující suť

```
(rubble_entity
 (<název entity> <pravděpodobnost výskytu>)
)
```

7.2.2.3 Entita generující kládu

```
(stem_entity
 (<název entity>)
)
```

7.2.2.4 Koeficienty snižující pravděpodobnost efektu

```
(building_rubble
 (max_entity      <maximální počet entit suti v simulátoru>)
 (prob_vehicle_shoot <snížení pravděpodobnosti při zásahu entity>)
 (max_same_gen_entity <počet generování stejných entit jako suť>)
)
```

7.2.2.5 Podmínky generování klád

```
(forest_rubble
 (range_in_forest <vzdálenost dopadu uvnitř lesa>)
 (prob_in_forest <pravděpodobnost výskytu uvnitř lesa>)
 (range_to_forest <vzdálenost dopadu mimo les>)
 (prob_to_forest <pravděpodobnost výskytu mimo les>)
 (next_near_forest <vzdálenost nejbližšího lesa>)
)
```

7.2.2.6 Entity budov

```
(building_entity
  (<seznam entit vystupující jako budovy>)
)
```

Příklad konfiguračního souboru uvádím v příloze 2.

Z výše uvedeného je patrné, že pomocí změny konfiguračního souboru můžeme ovlivnit chování celého modelu bez toho, že bychom jakkoli zasahovali do zdrojového kódu.

7.3 Návrh modelu generování klád

Generování klád nastává při dopadu dělostřelecké munice do, nebo v blízkosti lesa, dle požadavků uvedených v neformální specifikaci (viz. kap. 7.1). Les je zakomponován přímo v terénní databázi, která se nahrává při spuštění simulačního systému a za pomocí knihovny „libctdb“ můžeme s touto terénní databází pracovat.

Pro práci s lesy nám knihovna „libctd“ poskytuje pouze informaci, zda se námi uvedený bod nachází v lese, či nenachází v lese - tím vyvstává problém s určením hranice lesa, které potřebujeme v modelu znát pro vypočítání podmínek splňujících generování klád.

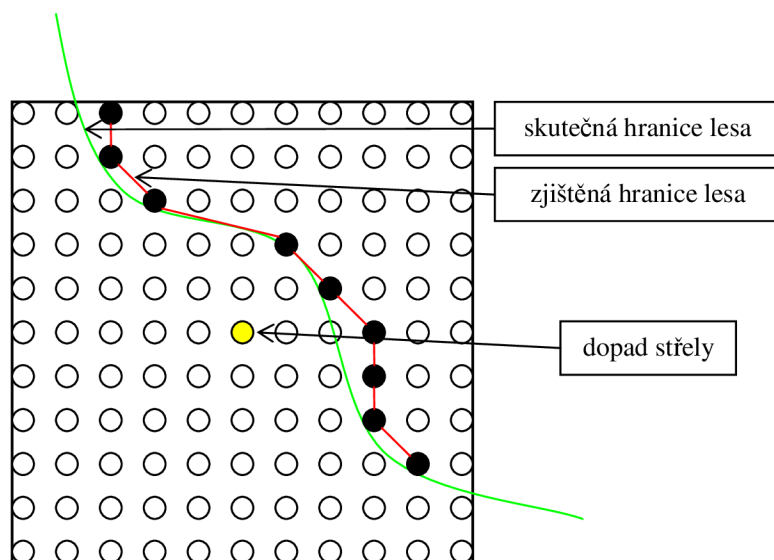
Jakmile si zjistíme hranici lesa, hledáme, zda se v blízkosti dané hranice nenachází jiný les a dle podmínek neformální specifikace generujeme entitu klády simulující např. spadlý strom.

7.3.1 Zjištění hranice lesa

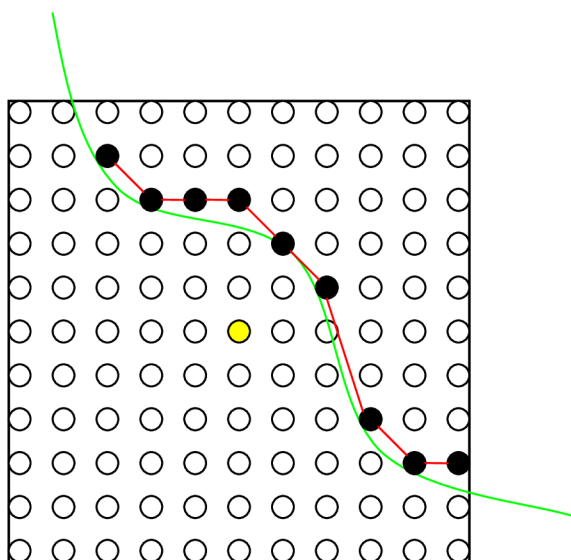
Pro zjištění hranice lesa využívám jednoduchý a rychlý algoritmus s použitím dané funkce z knihovny „libctd“, která nám určuje, zda se daný bod nachází nebo nenachází v lese. Vycházím z dopadu střely na terénní databázi, kde znám přesnou pozici dopadu a postupně vyhledávám v okolí body, které se nacházejí uvnitř, či mimo les.

Algoritmus prochází čtverec o zadané velikosti, jehož střed tvoří bod dopadu střely (viz. obr. 19 a 20). Jako první procházím postupně body zleva doprava od horního okraje čtverce postupně dolů. Jakmile takto procházím, zjišťuji, zda daný bod leží v lese, či je mimo les. Pokud se vyskytnou dva body jdoucí po sobě, přičemž jeden z nich se nachází v lese a druhý mimo les, zjistil jsem část hranice lesa. Všechny body které takto zjistím si uložím a ve výsledku z těchto bodů skládám přibližnou hranici lesa. Pokud se naopak všechny body nacházejí mimo les, případně v lese, nedochází k vytvoření žádné hranice bodů označující les.

Abych vyloučil špatně popsanou hranici lesa procházím v druhém kroku zadáný čtverec ještě jednou ale nyní procházím stejné body směrem od shora dolů postupně od levého okraje čtverce k pravému okraji, přičemž výsledná hranice se popisuje stejným způsobem uvedeným výše.



Obr. 19: Procházení bodů zleva doprava od shora dolů a vytvoření hranice lesa



Obr. 20: Procházení bodů od shora dolů zleva doprava a vytvoření hranice lesa

7.4 Návrh modelu generování suti

Model generování suti nám vytváří při dopadu dělostřeleckého granátu na budovu, případně v blízkosti budovy suť, skládající se z různých entit uvedených v konfiguračním souboru.

V systému OTBSAF mohou být budovy realizovány dvěma odlišnými způsoby. První způsob implementace budov je zahrnutí budov již do mapového podkladu (terénní databáze) a tak jsou tyto budovy závislé na použité mapě. Tyto budovy jsou složeny z různých polygonů o maximálním počtu vrcholů 16. Pokud bychom chtěli udělat budovu s rozmanitým tvarem převyšujícím 16 vrcholů jednoduše ji stvoříme z více složených budov dohromady. Nevýhodou budov, které jsou umístěny již v mapovém podkladu je fakt, že s těmito budovami nemůžeme v systému nikterak pracovat jako je např. umazávat, či přemísťovat za běhu simulace.

Druhý způsob implementace budov je realizace jako obecné entity, kterou můžeme umístit do systému. Takto realizovaná budova je nezávislá na terénní databázi (mapě) a může být umístěna kdekoli na tomto podkladu. Výhodou je, že takto vytvořenou budovu můžeme kdekoli umístit a natočit v námi požadovaném směru. Nevýhodou této implementace je, že budova nemůže vytvářet rozmanitý tvar, ale její půdorys je čtvercového, nebo obdélníkového charakteru.

Níže uvádím tabulku srovnávající obě dvě implementace budov v systému OTBSAF.

Implementace	Terénní databáze (mapa)	Systém OTBSAF
Maximální počet vrcholů	16	4
Závisí na použité mapě	ANO	NE
Vytvoření při startu systému	ANO	NE
Možnost přidávat budovy za běhu	NE	ANO
Možnost editovat budovy za běhu	NE	ANO

Tab. 2: Srovnání různých implementace budov v systému OTBSAF

7.4.1 Práce s budovami na terénní databázi

Jelikož většina budov v systému OTBSAF je simulována implementací v terénní databázi a mají velice rozmanitý tvar, skládají se proto tyto budovy z menších a jednodušších tvarů (budov) o maximálním počtu hran 16 (viz. obr. 21).

Pro práci s budovami na podkladové mapě slouží funkce z knihovny „libctdb,“ která nám vrací z daného středu a zvoleném poloměru všechny budovy, které se v tomto okruhu nacházejí. Problém nastává s budovami, které se skládají z více menších budov, kdy funkce vrací právě tyto menší kousky, z kterých se celá budova skládá.



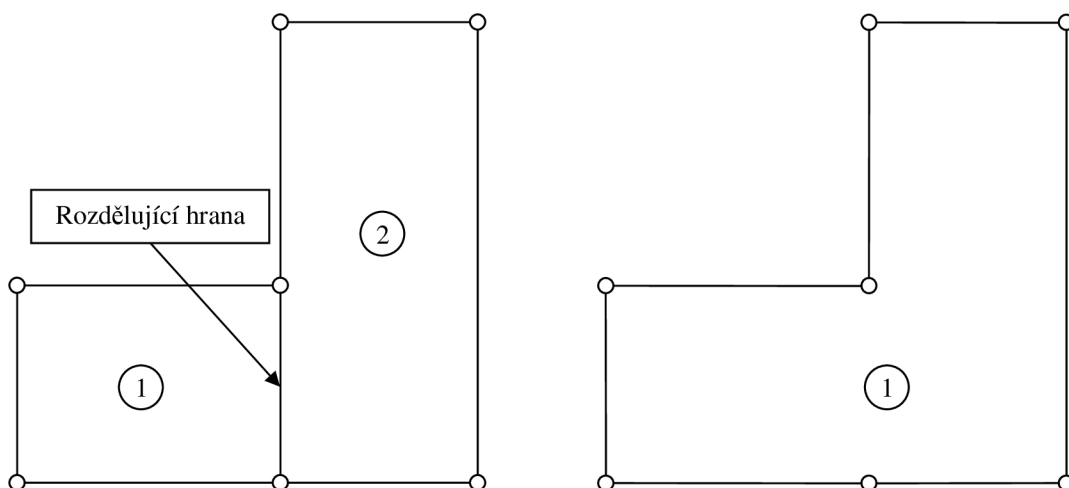
Obr. 21: Ukázka budovy složené z komplexu 7 malých budov

7.4.1.1 Sloučení budov

Budovy je nutné sloučit v jednu jedinou před vyhodnocením dopadu střely, tak jak je uživatel vidí na mapě, abychom mohli správně vypočítat a určit, kde se mají generovat sutiny z této budovy.

Algoritmus pro sloučení budov je jednoduchý a vychází z principu, že dvě sousedící budovy sdílejí minimálně jednu hranu, tím pádem obě dvě budovy musí mít minimálně dva společné body. Tyto dva společné body stačí na obou budovách najít a poté vytvořit novou budovu, která se bude skládat ze všech hran dřívějších budov, kromě hrany tvořící aktuální rozdělení.

Novou budovu začínám tvořit od rozdělující hrany vložím jednoho z vrcholu této hrany a poté postupným vkládáním jednotlivých vrcholů ostatních hran jedné z budov. Jakmile takto vložím veškeré vrcholy, dostávám se opět k vrcholu rozdělující hrany. Tento vrchol vkládám do nově vytvářené budovy a dále pokračuji ve vkládání vrcholů, ale nyní z druhé (slučované) budovy. U této budovy začínám vkládat vrcholy taktéž od rozdělující hrany, ale nyní tyto vrcholy rozdělující hrany vynechám. Ve výsledku nám vznikne nová budova, která bude mít veškeré vrcholy dřívějších budov, avšak nebude obsahovat hranu rozdělující tyto dřívější budovy. Obrázek 22 nám lépe ilustruje výsledek algoritmu před sloučením a po sloučení budov.



Obr. 22: Demonstrace algoritmu slučování budov: vlevo dům před sloučením – složen ze dvou nezávislých budov, vpravo dům po sloučení – složen jen z jedné budovy

7.5 Implementace modelu generování suti a klád

Model generování suti a klád je implementován v systému OTBSAF v jedné jediné knihovně pod názvem „libartyrubble“. Tato knihovna obsahuje 7 souborů potřebných pro správnou kompilaci a běh daného modelu v systému.

7.5.1 Popis jednotlivých souborů knihovny

1. arrub_config.rdr – konfigurační soubor (popsán v kapitole 7.2.2)
2. arrub_init.c – zdrojový soubor sloužící k počáteční inicializaci modelu
3. arrub_tick.c – zdrojový soubor implementující funkčnost celého modelu
4. GNUmakefile.in – soubor slouží pro správný překlad knihovny
5. libarrub_local.h – hlavičkový soubor obsahující deklarace funkcí, struktur a proměnných které využívá daný model
6. libartyrubble.h – hlavičkový soubor nutný k propojení knihovny s celým systémem OTBSAF
7. libartyrubble.texinfo – soubor popisující implementovanou knihovnu v jazyku LaTeX

7.5.2 Inicializace knihovny

Při inicializaci knihovny dochází nejdříve k zaregistrování funkce „arrub_process_pdu“, která se nachází v souboru „arrub_tick.c“, a která se vykoná pokaždé, když je zachycen paket PDU signalizující právě dopadlou střelu v simulátoru OTBSAF.

Po zaregistrování záchytu PDU paketů přichází na řadu inicializace konfiguračních dat z RDR souboru. Veškerá potřebná data se nahrají do připravených lokálních proměnných, aby mohly být v co nejkratším čase využity při následném výpočtu.

7.5.3 Běh modelů generování suti a klád

Jádro a funkčnost modelu generování suti a modelu generování klád je implementováno v souboru „arrub_tick.c“. V tomto souboru se nacházejí všechny hlavní i pomocné funkce nutné ke správnému běhu obou modelů. Oba dva modely jsou implementovány v jediném souboru proto, že podstata těchto modelů je téměř stejná, liší se pouze odlišným účinkem dělostřelecké palby.

7.5.3.1 Hlavní funkce

Hlavní funkce starající se o chod obou modelů se nazývá „arrub_process_pdus“. Tato funkce byla při inicializaci zaregistrována pro zachycení paketů PDU s dopadající střelou a proto když v simulátoru nastane tato situace, vykoná se právě tato hlavní funkce.

Jakmile je funkce vyvolána, dojde k získání terénní databáze, začne se prohledávat seznam munic, který byl nahrán z RDR souboru v inicializační části a vyhledává se právě dopadlá munice, která vyvolala tuto funkci. Pokud munice není v seznamu nalezena, nic se dále nezpracovává. Při nalezení munice v seznamu dochází k sestavení pravděpodobností výskytu efektu.

Po sestavní pravděpodobnosti výskytu efektu již dochází k zjištění, kde daná střela dopadla a kontroluje se, zda nedopadla v lese nebo blízkém okolí, na budovu nebo v blízkém okolí, případně na entitu struktury vystupující jako budova, či v jejím okolí a dle tohoto zjištění dochází ke generování suti a klád.

7.5.3.2 Sestavení pravděpodobnosti výskytu efektu

V neformální specifikaci je mnoho podmínek, které snižují pravděpodobnost daného efektu, proto je vytvořena speciální funkce „arrub_prob_thermometer“, která slouží ke zjištění daných podmínek v systému OTBSAF a případné korekci pravděpodobnosti výskytu efektu generování suti a klád.

Jako první zjišťujeme, kolik entit suti a klád je již celkem v simulátoru vygenerováno a pokud tato hodnota překračuje předem určenou mez, efekt nenastane. Dále snižujeme pravděpodobnost výskytu dle počtu již vygenerovaných sutin a klád v okolí daného dopadu. Nakonec kontrolujeme zda střela nezasáhla vozidlo, i v tomto případě dochází k snížení pravděpodobnosti výskytu efektu.

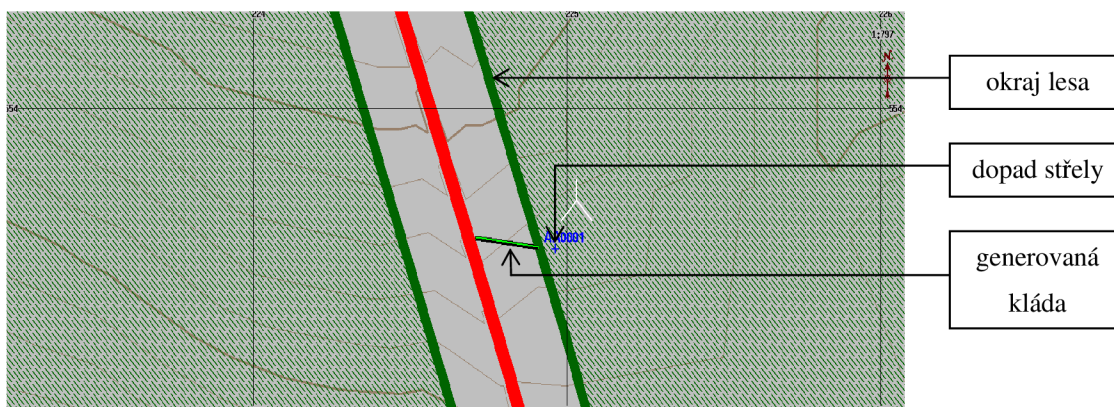
7.5.3.3 Dopad střely v lese, nebo blízkém okolí

Při dopadu střely do lesa nebo jeho blízkého okolí, zajišťují zjištění hranice tohoto lesa dle algoritmu popsaného v kapitole 7.3.1. Po zjištění hranice dále kontrolují, zda se v blízkosti této hranice nachází další lesní plocha, a to tím způsobem, že postupně beru jednotlivé body ohraničující hranici lesa a dle

zadané délky kontrolují, zda se kolmo na vybraný a příští bod nachází na obou stranách neprostupný les. Pokud tato podmínka je splněna, střela dopadla do místa, kde se může generovat kláda.

7.5.3.4 Tvorba klády

Tvorba jakékoli entity v simulátoru OTBSAF zajišťuje knihovna „libunitutil“ za pomoci funkce „unitutil_create“. Při vytváření klády vybíráme náhodný bod z vytvořené hranice lesa. Jelikož tento bod leží v lese a entita klády se má generovat mimo les, vytvoříme bod generování kolmo na tento bod mimo lesní plochu. Nakonec náhodně natáčíme entitu klády a vytváříme ji pomocí funkce „unitutil_create“.



Obr 23: Ukázka vygenerované klády při dopadu střely do lesa

7.5.3.5 Dopad střely na budovu implementovanou v mapovém podkladu

Zjištění, zda střela, která právě dopadla, zasáhla budovu, nám zajišťuje funkce „vgrdar_is_in_area“ z knihovny „libvgrdar“ – tato funkce vrací logickou hodnotu dle zjištění, zda bod leží uvnitř polygonu (budovy), či leží mimo polygon.

Jelikož se má suť generovat u nejbližší hrany budovy vzhledem k dopadu střely, seřazují si postupně všechny hrany budovy od nejbližší k nejvzdálenější hraně. Po seřazení hran kontrolují, zda nejbližší hrana není náhodou vnitřní hranou budovy, která sousedí s jinou budovou a dohromady tvoří rozsáhlejší komplex jedné jediné budovy. Pokud hrana není vnitřní hranou, jedná se o venkovní hranu budovy a my můžeme generovat suť u této hrany, avšak jakmile zjistíme, že u této hrany sousedí další budova, dochází k použití algoritmu popsaného v kapitole 7.4.1.1 (Sloučení budov), kdy postupně vytvoříme nové hrany ohraničující budovu. Jakmile máme sloučenou budovu s novými hranami, opět seřadíme tyto hrany od nejbližší k nejvzdálenější a algoritmus celý následně opakujeme.

7.5.3.6 Dopad střely na budovu vkládanou do systému OTBSAF

Budovy vkládané do systému OTBSAF jako jednotlivé entity mají tu výhodu, že jejich půdorys je vždy čtyřhranný a nemohou tvořit rozsáhlejší komplexy budov, tím odpadá jakékoli testování a případné slučování vnitřních hran s jinou budovou.

Nejprve si získám ze simulátoru všechny vložené struktury do scénáře za pomoci funkce „pbt_get_vehicles“ implementované knihovnou „libpbt“. Poté procházím seznamem všech takto zjištěných struktur a kontroluji, zda daná struktura je budovou, či nikoli (seznam všech budov je zapsán v konfiguračním RDR souboru a načítá se při inicializaci). Po vybrání všech budov kontroluji za pomoci funkce „vgrdar_is_in_area“, zda střela dopadla na budovu, či nikoli. Jakmile je potvrzeno, že daná střela dopadla na budovu, zjišťuji, která ze čtyř hran je nejbližší a u této hrany můžu generovat suť.

7.5.3.7 Dopad střely v blízkosti budovy

Pokud nám funkce „vgrdar_is_in_area“, která nám říká, zda střela dopadla či nedopadla na budovu hlásí, že se nenacházíme v žádné z budov, musíme zkontrolovat, zda munice nedopadla v požadované blízkosti u některé z budov.

Postupně bereme jednotlivé budovy do zadané vzdálenosti a za pomoci funkce „vgrdar_point_to_segment_distance“ si zjistíme v jaké vzdálenosti jsou od místa dopadu jednotlivé hrany každé z budov. Takto nalezneme nejbližší hranu od místa dopadu a mezi touto hranou a místem dopadu můžeme generovat suť.

7.5.3.8 Generování suti

Generovaná suť se skládá z různých typů entit, které jsou uvedeny v konfiguračním RDR souboru (viz. kapitola 7.2.2). Generované entity mají různou pravděpodobnost vygenerování. Další omezující podmínka je v různém počtu stejně vygenerovaných entit za sebou. Proto si při generování entity ukládám seznam již dříve vytvořených entit a při vytváření nové entity kontroluji, zda se nepřekračuje zadaný limit počtu stejně vygenerovaných entit za sebou. Pokud bychom tento limit překročili, náhodně vybíráme jinou entitu ke generování.

Suť vytváříme obdobným způsobem, jako tvorbu klády, a to za pomoci funkce „unitutil_create“ z knihovny „libunitutil“. Pokud střela dopadla na budovu, vytváříme suť u nejbližší hrany, a to náhodně vybráním místa generování, které leží kolmo na tuto hranu mimo zastavěnou plochu budovy. Jakmile střela dopadla mimo budovu v blízkosti některé z hran, generujeme suť mezi danou hranou a místem dopadu střely.



Obr. 24: Ukázka vygenerované suti při dopadu střely na budovu

8 Závěr

V diplomové práci jsem uvedl všeobecný základ a úvod do modelování a simulace. Tento obor je velmi rozsáhlý a stále dochází k rozvoji, proto jsem jednu z částí – konstruktivní simulaci, popsal detailněji v jejich základech a principech. Snažil jsem se přiblížit modelování a simulaci z pohledu využití pro vojenské účely, jelikož má práce směřuje tímto směrem. V návaznosti na to jsem popsal, jak se různé druhy simulací aktuálně využívají ve vojenství.

Dále jsem uvedl a popsal simulační systém OneSAF Testbed Baseline (OTBSAF), který aktuálně využívá armáda ČR pro výcvik jednotlivců, velitelů a štábů. Pro tento systém jsem implementoval dva navržené modely – model psychologických efektů a model generování suti a klád při bojové činnosti dělostřelectva.

Model psychologických efektů se snaží do systému OTBSAF vnést tzv. měkký faktor simulace při bojové činnosti dělostřelectva, kdy simulované entity a jednotky při dopadu dělostřeleckých granátů v jejich blízkosti se začnou chovat odlišně v závislosti na daných podmínkách. V současné době tento model řeší pouze základní chování, které je možno v systému OTBSAF u entit ovlivnit. V budoucnu vidím možnost vylepšení tohoto modelu o přidání dalších různých reakcí na dělostřeleckou palbu, případně vylepšení modelů chování pokud se entita nachází ve skupině, či je skryta v lese, případně budově.

Model generování suti a klád přibližuje systém OTBSAF více realitě, kde se počítá s následky dělostřelecké činnosti při dopadu na různý terén, popř. zastavěnou plochu. Model je velice komplexní a dovoluje přidávat i v budoucnu nové entity generované jako suť, nebo označovat různé další entity jako budovy z kterých se následně zásahem může generovat suť. Případná další vylepšení tohoto modelu vidím v započítání okolních budov do procesu tvorby suti.

Oba výše zmíněné modely byly otestovány a v současné době již jsou plně nasazeny a využity v Centru simulačních a trenažerových technik na Univerzitě obrany v Brně, kde se s nimi setkávají velitelé čet, rot a štábů při různých plánovaných cvičeních.

Literatura

- [1] „Autor neznámý“. *Simulation* [online]. Poslední modifikace: 3. ledna 2008. Dostupné na URL: <http://en.wikipedia.org/wiki/Simulation> [cit. 2009-5-21]
- [2] Rybár, M a kolektiv. *Modelovanie a simulácia vo vojenstve*. MO SR. Bratislava 2000
- [3] Kmec, M. Rybár, M. *Metódy a prostriedky konštruktívnej simulácie bojovej činnosti*. Otázky taktiky č. 2/1998. MO SR. Bratislava
- [4] VR Group, a.s. Toužimská 583, Praha 9. *Pozemní vojsko – schéma taktického výcviku* [online]. Dostupné na URL: <http://www.vrg.cz/sch-t-cz.html> [cit. 2009-5-21]
- [5] VR Group, a.s. Toužimská 583, Praha 9. *Pozemní vojsko – schéma cvičení s počítačovou podporou (CAX)* [online]. Dostupné na URL: <http://www.vrg.cz/sch-cax-cz.html> [cit. 2009-5-21]
- [6] VR Group, a.s. Toužimská 583, Praha 9. *Pozemní vojsko – rekonfigurovatelné virtuální simulátory* [online]. Dostupné na URL: <http://www.vrg.cz/prod1-vs2-cz.html> [cit. 2009-5-21]
- [7] „Autor neznámý“. *Standardizované protokoly informačních technologií pro distribuovanou interaktivní simulaci (DIS)*. Český obranný standard. Úřad pro obrannou standardizaci, katalogizaci a státní ověřování jakosti. Praha 2005
- [8] Ternion Corporation. *What Is FLAMES* [online]. Dostupné na URL: <http://www.ternion.com/> [cit. 2009-5-21]
- [9] VT MÄK, 68 Moulton Street, Cambridge. *VR-Forces* [online]. Dostupné na URL: <http://www.mak.com/products/vrforces.php> [cit. 2009-5-21]
- [10] U.S. Joint Forces Command, 1562 Mitscher Ave, Norfolk. *Joint Conflict and Tactical Simulation (JCATS)* [online]. Dostupné na URL: http://www.jfcom.mil/about/fact_jcats.htm [cit. 2009-5-21]
- [11] Shimamoto, Faith. *Simulation Warfare Is No Video Game* [online]. Dostupné na URL: <https://www.llnl.gov/str/Shimamoto.html> [cit. 2009-5-21]

- [12] Institute for Creative Technologies. *OneSAF Objective System* [online]. Dostupné na URL: http://ict.usc.edu/projects/onesaf_objective_system/ [cit. 2009-5-21]
- [13] Science Applications International Corp, 12901 Science Drive, Orlando. *Software Architecture Design and Overview Document (SADOD) for OTBSAF*.
- [14] Smith, Joshua. *LibTask – LibTask Programmer’s Reference Manual*
- [15] Dr. Braudaway, Wesley. Buettner, Cedric. Chamberlain, Forrest. Dr. Evans, Alan. Longtin, Michael. Smith Joshua. Stanzone, Tom. *LibCTDB – Compact Terrain DataBase Library*
- [16] Science Applications International Corp, 12479 Research Parkway, Orlando. *One Semi-Automated Forces (OneSAF) Testbed Baseline (OTB) Advanced Developer’s Course*.
- [17] Smith, Joshua. *LibReader – LibReader Programmer’s Reference Manual*

Seznam příloh

Příloha 1. Příklad konfiguračního souboru pro knihovnu „libvpsychoeffect“ implementující model psychologických efektů při bojové činnosti dělostřelectva.

Příloha 2. Příklad konfiguračního souboru pro knihovnu „libartyrubble“ implementující model generování suti a klád při bojové činnosti dělostřelectva.

Příloha 3. CD se zdrojovými kódy knihoven „libvpsychoeffect“ a „libartyrubble“ implementující model psychologických efektů a model generování suti a klád při bojové činnosti dělostřelectva.

Příloha 1

Příklad konfiguračního souboru pro knihovnu „libvpsychoeffect“ implementující model psychologických efektů při bojové činnosti dělostřelectva:

```
(
; =====Delostrelecky granat=====
  (munition_CZ_152_EOF_ShKH77NkP_NP
    (range      50   75   100  150)
    (prob       1.0  0.8  0.6  0.4)
  )
; =====Minometny granat=====
  (munition_CZ_120_OF_NP6
    (range      50   75   100  150)
    (prob       1.0  0.8  0.6  0.4)
  )
; =====Raketometna munice=====
  (munition_CZ_M_21_OF
    (range      75   100  125  175)
    (prob       1.0  1.0  0.8  0.6)
  )
; =====Letecke pumy=====
  (munition_US_Maverick
    (range      75   100  125  175)
    (prob       1.0  1.0  0.8  0.6)
  )
)
```

Příloha 2

Příklad konfiguračního souboru pro knihovnu „libartyrubble“ implementující model generování suti a klád při bojové činnosti dělostřelectva:

```
(config
  (munition_rubble
    ; =====Delostrelecky granat=====
    (munition_CZ_152_EOF_ShKH77NkP_NP
      (range          0    5    10   15)
      (prob            1.0  0.75 0.4  0.15)
      (num_entity      5    3    1    1)
    )
  )
  (rubble_entity
    (structure_CZ_Hromada_dratu      10)
    (structure_CZ_Zborcena_zed       15)
    (structure_CZ_Skruz               10)
    (structure_CZ_Skrin              10)
    (structure_CZ_Stavebni_sut       10)
    (structure_CZ_Stavebni_sut_V     15)
    (structure_CZ_Traverza           10)
    (structure_CZ_Zlomeny_panel      10)
    (structure_CZ_Tram               10)
  )
  (stem_entity
    structure_CZ_Klada
  )
  (building_rubble
    (max_entity                100)
    (prob_vehicle_shoot        0.5)
    (max_same_gen_entity       3)
  )
  (forest_rubble
    (range_in_forest           15)
    (prob_in_forest            0.25)
  )
)
```

```
(range_to_forest          5)
(prob_to_forest          0.10)

(next_near_forest        50)
)
(building_entity
(structure_CZ_Admin_budova structure_CZ_Cerpaci_stanice
 structure_CZ_Chata structure_CZ_Jaderna_elektrarna)
)
)
```