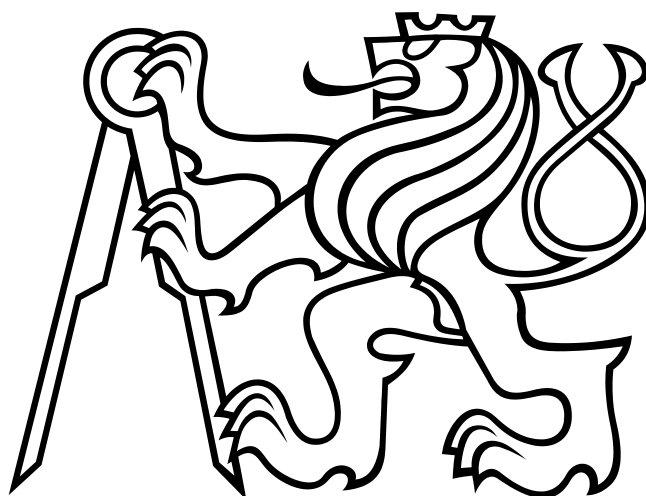


ČESKÉ VYSOKÉ UČENÍ TECHNICKÉ V PRAZE
FAKULTA ELEKTROTECHNICKÁ

BAKALÁŘSKÁ PRÁCE



Vojtěch Křížek

**Agentní modelování nelegitimního chování
v námořním provozu**

Katedra kybernetiky

Vedoucí bakalářské práce: Ing. Michal Jakob, Ph.D.

Praha, 2010

ZADÁNÍ BAKALÁŘSKÉ PRÁCE

Student: Vojtěch Křížek

Studijní program: Elektrotechnika a informatika (bakalářský), strukturovaný

Obor: Kybernetika a měření

Název tématu: Agentní modelování nelegitimního chování v námořním provozu

Pokyny pro vypracování:

1. Seznamte se s modelováním chování agenta pomocí konečných stavových automatů.
2. Seznamte se se současnými strategiemi pirátů v oblasti Somálska a Adenského zálivu a vypracujte jejich přehled. Zjistěte, jaká data reálného světa jsou potřeba pro vytvoření modelů pirátského chování.
3. Namodelujte jednotlivé pirátské strategie pomocí konečných stavových automatů formálně, za použití jazyka UML.
4. Seznamte se s multi-agentní platformou AgentC a závislými moduly.
5. Naimplementujte jednotlivé modely chování a vytvořte vhodné scénáře pro jejich demonstraci.
6. Vyhodnoťte jednotlivé modely chování z hlediska úspěšnosti na jednotlivých scénářích.

Seznam odborné literatury:

- [1] Russel, S. and Norvig, P.: Artificial Intelligence: A Modern Approach, Englewood Cliffs, New Jersey, 1995.
- [2] Adamczyk, P.: The Anthology of the Finite State Machine Design Patterns, The 11th Cons. on Pattern Languages of Programs, 2004.
- [3] Tsvetkova B.: Securitizing Piracy Off the Coast of Somalia, Central European Journal of International and Security Studies, 2008.
- [4] Vybrané články z Kuala Lumpur Conference on Piracy and Crimes on Sea, 2009.

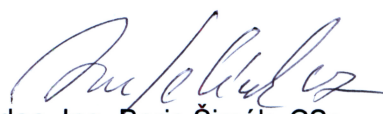
Další literatura bude specifikována vedoucím práce dle potřeby.

Vedoucí bakalářské práce: Ing. Michal Jakob, Ph.D.

Platnost zadání: do konce zimního semestru 2010/2011


prof. Ing. Vladimír Mařík, DrSc.
vedoucí katedry





doc. Ing. Boris Šimák, CSc.
děkan

V Praze dne 12. 1. 2010

Prohlášení

Prohlašuji, že jsem svou diplomovou práci vypracoval samostatně a použil jsem pouze podklady (literaturu, SW, projekty atd.) uvedené v příloženém seznamu.

V Praze dne 25. 5. 2010



.....

podpis

Děkuji Centru agentních technologií za možnost podílet se na vývoji projektu AgentC, zejména pak Ing. Michalu Jakobovi, Ph.D. a Ing. Ondřejovi Vaňkovi za cenné rady, připomínky a pomoc při tvorbě této práce. Také děkuji za poskytnutí L^AT_EX šablony pro sazbu této práce Ing. Janu Faiglovi.

Abstrakt

Tato bakalářská práce se zabývá implementací nelegitimního chování lodí v námořním provozu pomocí agentních modelů, které jsou vytvořeny pomocí konečných stavových automatů. Dále je v práci vypracován přehled problematiky pirátských lodí v okolí pobřeží Somálska z hlediska použití získaných dat pro implementaci agentních modelů. Je zde navrženo několik různých pirátských strategií s ohledem k získaným datům. Jednou z navržených strategií je i učící se strategie na bázi herního algoritmu multi-armed bandit. Tyto strategie jsou zahrnuty v jednotlivých agentních modelech lodí. Tyto modely jsou popsány za použití jazyka UML a jsou naimplementovány v jazyce Java v rámci multi-agentní platformy AgentC. Součástí práce jsou také simulace naimplementovaných strategií pirátských lodí a jejich vyhodnocení vůči reálné situaci v oblasti Somálska.

Abstract

The thesis deals with an implementation of illegitimate activities of vessels in the maritime traffic using agent models and finite state machines. An informal overview of recent pirate activity off the coast of Somalia is presented. From this overview, a formal model of the pirate activity is drawn and it is used as a source of data for the implementation of several different pirate models. Moreover, an adaptive pirate strategy based on the multi-armed bandit algorithm is designed. The models are described using UML language and they are implemented using Java language as the part of the AgentC platform. Simulations of implemented strategies of pirate vessels and their evaluation towards real-world situation around Somalia are presented.

Obsah

Úvod	1
1 Přehled problematiky	2
1.1 Úvod do problematiky	2
1.2 Popis oblasti	3
1.2.1 Definice	3
1.2.2 Neformální popis	3
1.2.3 Formální popis	6
1.3 AgentC	7
1.3.1 Zavádění modulů pro simulaci a tok dat	8
1.3.2 Běh simulace	9
1.3.3 Prostředí (Environment)	11
1.3.4 Agenti	11
1.3.5 Zdroje dat	12
1.3.6 Vizualizace v Google Earth	12
2 Agentní modely	14
2.1 Jednotlivé strategie a FSM modely	14
2.1.1 Strategie No Info	15
2.1.2 Strategie Radar, Mothership + Radar	15
2.1.3 Strategie AIS, Mothership + AIS	15
2.1.4 Strategie Learnable	16
2.1.5 Strategie Bandit (multi-armed bandit)	16
2.2 UML diagramy FSM modelů	18
2.2.1 Strategie No Info a strategie Bandit	18
2.2.2 Strategie Radar, Mothership + Radar	19
2.2.3 Strategie AIS, Mothership + AIS	21
2.2.4 Strategie Learnable	21

3 Implementace	23
3.1 Popis implementace	23
3.2 Implementace strategií	25
4 Vyhodnocení	26
4.1 Popis simulace a scénářů	26
4.1.1 Nastavené konstanty pro simulace	26
4.2 Testování a výsledky	28
4.2.1 Získané grafy	28
4.3 Diskuze	39
5 Závěr	41
Příloha A: Obsah CD	44

Seznam tabulek

1.1	Formalizovaná data	6
1.2	Vzdálenost horizontu v závislosti na výšce pozorovatele	7
4.1	Použité konstanty pro simulace	28
1	Adresářová struktura na CD	44

Seznam obrázků

1.1	MSPA a IRTC koridor v Adenském zálivu [14]	4
1.2	Cesta uplutá pirátskými loděmi (červeně) [11]	4
1.3	Rub letáku Somali Pirate Tactics [15]	5
1.4	Znázornění pro odvození vzdálenosti horizontu	7
1.5	Schéma vnitřní struktury platformy AgentC [7]	9
1.6	UML diagram FSM transportní lodě	12
1.7	UML diagram FSM armádní lodě	12
2.1	UML diagram strategií No Info a Bandit	18
2.2	UML diagram strategie Radar	19
2.3	UML diagram strategie Mothership + Radar	20
2.4	UML diagram strategie AIS	20
2.5	UML diagram strategie Mothership + AIS	21
2.6	UML diagram strategie Learnable	22
3.1	UML diagram vybraných základních a zděděných tříd [[4]]	24
4.1	Graf útoků na transportní lodě v závislosti na počtu pirátských lodí .	29
4.2	Graf úspěšných útoků v závislosti na počtu pirátských lodí	30
4.3	Graf útoků na transportní lodě v závislosti na jejich rychlosti	30
4.4	Graf útoků v závislosti na vzdálenosti horizontu (výšce pirátské lodi)	31
4.5	Počet útoků v čase v závislosti na vzdálenosti horizontu (No Info strategie)	32
4.6	Útoky během dne (No Info strategie, 50 lodí)	32
4.7	Celkový počet útoků v čase v závislosti na oblasti (No Info strategie)	33
4.8	Celkový počet útoků v čase v závislosti na oblasti (Bandit strategie) .	33
4.9	Uplutá vzdálenost v závislosti na počtu pirátských lodí	34
4.10	Uplutá vzdálenost v čase v závislosti na oblasti (No Info strategie) . .	35
4.11	Uplutá vzdálenost v závislosti na rychlosti transportních lodí	35

4.12	Uplutá vzdálenost v závislosti na vzdálenosti horizontu	36
4.13	Max. procentní hodnota v čase v závislosti na oblasti (Bandit strategie)	37
4.14	Aktuální max. procentní hodnota v závislosti na rychlosti transp. lodí (Bandit strategie)	37
4.15	Aktuální procentní hodnota v závislosti na rychlosti transportních lodí (Learnable strategie)	38
4.16	Aktuální max. procentní hodnota v závislosti na vzdálenosti horizontu (Bandit strategie)	38
4.17	Aktuální procentní hodnota v závislosti na vzdálenosti horizontu (Learn- able strategie)	39
4.18	Vizualizace simulace strategie Bandit pro 10 lodí v Google Earth . . .	40

Úvod

Motivací této práce je pomoci při vývoji projektu, který má za úkol zlepšit odhalování nelegitimního chování v námořním provozu. Výskyt nelegitimního chování, pojmenovávaného jako pirátství, v poslední době velice narůstá zejména v politicky nestabilních regionech. Také začíná více zasahovat jak do běžné lodní dopravy surovin a výrobků, tak také do turistické dopravy a to zejména ve formě přepadení lodí a čekání na výkupné. Proto je pomoc při pochopení chování a formy útoků pirátských lodí pro současné ochranné armádní složky v těchto exponovaných oblastech velice vítaná.

Úkolem práce je seznámení se s modelováním chování agentů pomocí konečných stavových automatů. Dále pak získání dat a zpracování přehledu o současných strategiích pirátů v oblasti Adenského zálivu a Somálské pánve. Také je třeba zjistit potřebná reálná data pro vypracování těchto strategií a modelů. Potom je úkolem namodelovat tyto pirátské strategie pomocí konečných stavových automatů za použití jazyka UML. Pro úspěšnou implementaci je také potřeba se seznámit s multi-agentní platformou AgentC a závislými moduly. Velmi důležitým bodem práce je také implementace formálních modelů v jazyce UML do projektu AgentC a to včetně potřebných scénářů, které se použijí pro vyhodnocení. Závěrečným bodem práce bude již zmíněné vyhodnocení jednotlivých naimplementovaných modelů z hlediska jejich úspěšnosti v závislosti na zvoleném scénáři a v porovnání se získanými reálnými daty.

Výsledkem práce jsou hotové modely agentů simulující nelegitimní chování na moři a to jak ve formě UML diagramů, tak ve formě zdrojového kódu, který je součástí projektu AgentC. Dále pak sumarizovaná data o chování pirátů z různých otevřených zdrojů. Zároveň jsou součástí získaná data simulací naimplementovaných modelů pro vyhodnocení úspěšnosti jednotlivých scénářů.

V kapitole 1 je uveden přehled problematiky, používané definice, neformální a formální popis problému a popis multi-agentní platformy AgentC. V kapitole 2 jsou popsány jednotlivé implementované strategie a jsou zde jejich UML diagramy. Kapitola 3 obsahuje popis implementace navržených strategií a konečných stavových automatů v jazyce Java. V kapitole 4 je popsán způsob simulace strategií a jsou zde shrnuta získaná nasimulovaná data. Poslední kapitolou 5 je závěr.

Kapitola 1

Přehled problematiky

Obsahem této kapitoly bude úvod do problematiky, popis zpracovávané oblasti, a multi-agentní platformy AgentC, na které je implementace této práce postavena.

1.1 Úvod do problematiky

Nelegitimní chování na moři je rozšířený jev a zahrnuje několik druhů tohoto chování. Tyto druhy je možné rozdělit na problematiku pašování drog, nelegálního rybaření, obchodu s lidmi, problematiku vypouštění nelegálního odpadu do moře a také problematiku pirátství. V této práci se zaměřím na poslední jmenovaný problém pirátství, zejména pak na oblast Adenského zálivu a Somálské pánve, čili na oblast okolo pobřeží Somálska, která je v poslední době i velice mediálně známá.

V oblasti Adenského zálivu proplouvají transportní lodě, které hlavně do Evropy přiváží suroviny a výrobky z Asie. Tato cesta je totiž nejkratší, a proto také nejvýhodnější. Zároveň se v této oblasti rybaří kvůli obživě místních lidí. Rybáři zde mají plno přístavů po obou pobřežích tohoto zálivu.

Z důvodu občanské války v Somálsku a nyní již několik let nefungujícího politického systému v zemi jsou obyvatelé Somálska velmi chudí lidé, a tak se někteří rozhodli si vydělávat pirátstvím. Výkupné jim totiž vynese dost peněz na obživu a na život nad místní poměry. Útoky jsou velice podobné a probíhají způsobem, že si piráti v moři naleznou loď, na kterou se pokusí zaútočit. V případě úspěchu s ní odplují k somálskému přístavu a zde čekají na výkupné, které se prakticky vždy dostaví. V případě zajmutí armádní lodí jim nehrozí téměř žádný trest, protože by je museli jednak přistihnout při útoku a také pro piráty je i toto zajmutí a držení v cele daleko lepší z hlediska jejich životní úrovně než být v Somálsku. Většina zajmutých pirátů je obvykle po pár dnech stejně propuštěna, protože nejsou prakticky postižitelní [13].

1.2 Popis oblasti

V této sekci bych se chtěl zaměřit na definování pojmů, které budu dále v práci používat, a na popis problematiky nelegitimního chování plavidel na moři. Toto chování bych chtěl rozebrat jak z neformální roviny popisu, tak z formálního hlediska, ze kterého získané parametry chování lodí budu dále v práci používat a to zejména pro implementaci jednotlivých modelů agentů.

1.2.1 Definice

Agent je dle [12] cokoliv, co umí vnímat okolní prostředí pomocí senzorů a podle vnitřního vyhodnocení vyvolávat akci pomocí akčních členů.

Konečný stavový automat (FSM¹) je podle [2] uspořádaná šestice

$$M = \langle X, S, Z, \omega, \delta, S_0 \rangle$$

kde X je konečná množina všech vstupních vektorů, S je konečná množina všech vnitřních stavů, Z je konečná množina všech výstupních vektorů, δ je přechodová funkce (zobrazení $\delta : X \times S \rightarrow S$), ω je výstupní funkce (zobrazení $\omega : X \times S \rightarrow Z$ nebo $\omega : S \rightarrow Z$) a S_0 je počáteční stav ($S_0 \in S$).

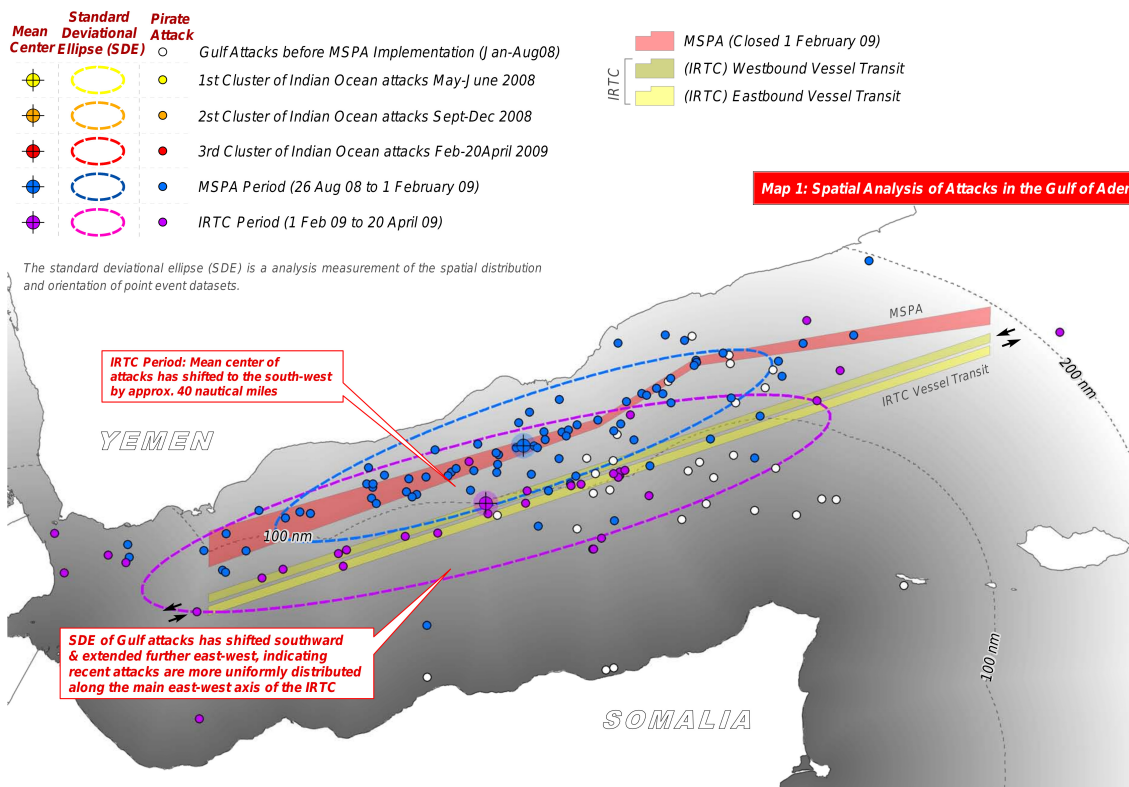
V této práci budu dále používat ekvivalent mezi pojmy nelegitimně se chovající plavidlo a pirát. Bude-li to zapotřebí, tak toto označení upřesním.

1.2.2 Neformální popis

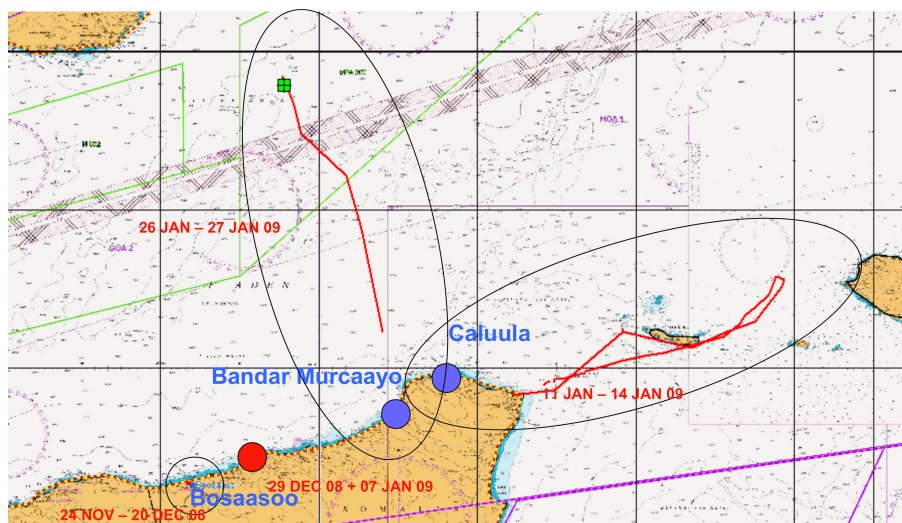
Informace o pirátských lodích a jejich chování na moři byly získávány z různých internetových zdrojů. Z těchto zdrojů byly nejužitečnější zejména armádní prezentace a články, a pak také konference zabývající se touto problematikou.

V oblasti Adenského zálivu se používá pro plutí transportních lodí nový koridor se jménem *IRTC* (Internationally Recognised Transit Corridor) [14]. Tento koridor oproti dřívějšímu *MSPA* (Maritime Security Patrol Area) koridoru [6], který byl uzavřen 1. února 2009 [14], má oddělené části pro jednotlivé směry plutí (viz obr. 1.1) a zároveň poskytuje přímou cestu bez nutnosti zatačet. Jelikož je tento nový koridor přibližně uprostřed Adenského zálivu, tak nevadí rybářům z Jemenu ani ze Somálska. Také to znesnadňuje případné pirátské útoky z Jemenu [9]. Tímto koridorem ročně propluje okolo 20 000 lodí a také se touto cestou transportuje 7% světové produkce ropy, což dělá tuto oblast pro piráty velice „atraktivní“.

¹FSM – Finite State Machine



Obrázek 1.1: MSPA a IRTC koridor v Adenském zálivu [14]

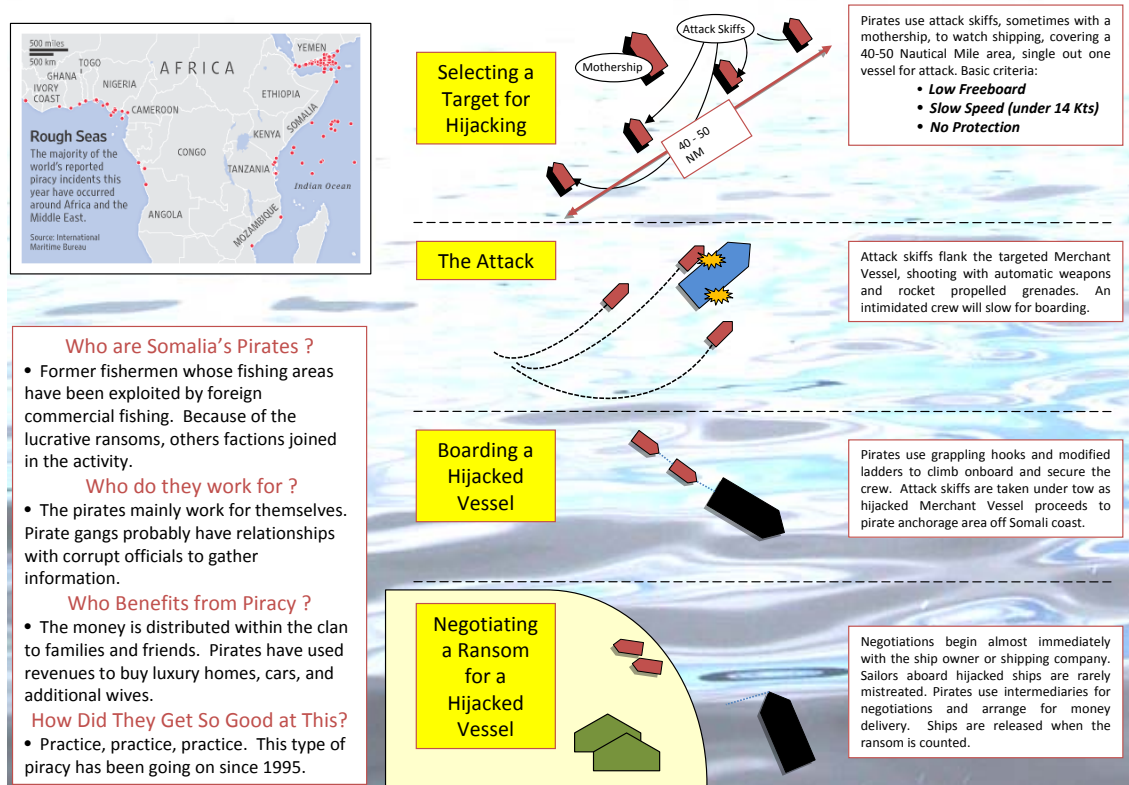


Obrázek 1.2: Cesta uplutá pirátskými loděmi (červeně) [11]

Somálští piráti jsou lidé, kterým je mezi 17 až 30 lety, a jsou obvykle velice špatně vzdělaní a nezaměstnaní. Pocházejí zejména z regionů u severního a východního pobřeží. Příčina tohoto jejich chování je pravděpodobně v nefungujícím politickém systému Somálska, nefunkční justici a také v úplatných odpovědných lidech, kteří by toto měli postihovat. Zároveň je to také důsledek občanské války a vysoké míry chudoby a s tím také spojené nezaměstnanosti. Kvůli těmto důvodům se dávají

někteří lidé kvůli vidině vysokého zisku na pirátství [13]. Tento zisk podle odhadů může dosahovat asi 15 tisíc dolarů za úspěšný útok a výkupné pro jednoho piráta [5].

V nalezených zdrojích byla získána také data o pirátských lodích a jejich způsobu přepadávání lodí. Piráti obvykle používají buď jednu menší loď nebo až dvě menší lodě [11] o rychlosti maximálně 25 uzlů [8], nebo *mateřskou loď* s třemi až pěti čluny, které mají rychlost až 30 uzlů [5]. Piráti atakují zejména transportní lodě o rychlosti menší než 15 uzlů, s nízkou polohou paluby a s malým zabezpečením. Většina útoků je za dne (cca 90%) a nejvíce obvykle při rozbřesku a za úsvitu. Útok trvá v Adenském zálivu cca 30 minut a v Somálské pánvi cca 1 hodinu. Tento rozdíl je dán rizikem, že je může dopadnout armádní loď. Při útoku piráti používají obvykle žebříky pro vylezení na palubu lodí. Také jsou ozbrojeni a to obvykle samopaly AK-47 (Kalašnikov) a i protitankovými střelami a ručními granáty [13]. Podařilo se rovněž nalézt zdroj, kde je vizualizace upluté cesty pirátskými loděmi ze zabavených GPS zařízení (obr. 1.2). Zde je vidět, jak dlouho piráti plují po určité trajektorii a také směr plutí. Způsob útoku pirátů na transportní loď je pak možné vidět na obr. 1.3. Zde je znázorněno napadení transportní lodě pomocí člunů, které jsou koordinovány z mateřské lodi.



Obrázek 1.3: Rub letáku Somali Pirate Tactics [15]

Jeden z největších problémů pirátství jsou náklady pro výkupné a s tím spojené lepší vyzbrojení pirátů. Výkupné je v průměru cca 2 miliony dolarů za jednu zajatou loď. Jen mezi lednem a listopadem roku 2008 bylo pirátům v Somálsku zapláceno

na výkupném okolo 30 milionů dolarů. Toto má také za následek zvyšování pojistek pro lodí, které touto oblastí proplouvají, u pojišťoven. Jeden velmi známý případ výkupného bylo za propuštění ropného tankeru Sirius Star v lednu roku 2009. Zde bylo zapláceno 3 miliony dolarů a to zejména kvůli tomu, že tanker vezl ropu v celkové hodnotě 100 milionů dolarů [3].

Celkové *riziko útoku* na proplouvající transportní loď je podle statistik méně než 1% pro Adenský záliv a asi 1,4% pro Somálskou pánev. Podle webových stránek <http://www.icc-ccs.org/>, kde jsou shromažďovány jednotlivé provedené útoky, toto číslo odpovídá, protože jsou zde zveřejněny asi 3 nové útoky během týdne. V případě útoku na loď je riziko přepadení lodí podle zdroje [14] 20,8% v Adenském zálivu a 27,3% v Somálské pánvi (z dat mezi 1. 1. až 20. 4. 2009). Také se podle tohoto zdroje zvyšuje vzdálenost útoků na loď od pobřeží Somálska a to až k 700 km v Indickém oceánu (aktuální číslo k dubnu 2009).

1.2.3 Formální popis

V této sekci zformalizuji získaná data z předchozí kapitoly. Tato data budou v dalších kapitolách použita pro implementaci agentních modelů pirátů a rovněž pro další odkazování na jejich hodnoty.

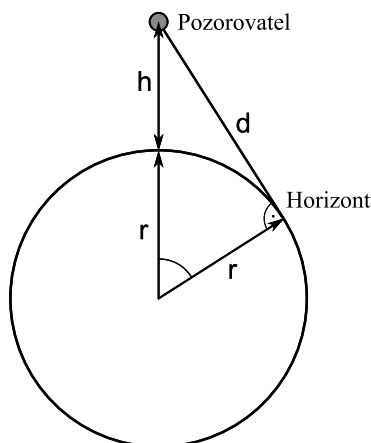
Data byla uspořádána přehledně do tabulky 1.1. V této tabulce jsou uvedeny hodnoty, které byly získány analýzou nalezených dat v předchozí sekci 1.2.2 (Neformální popis).

Tabulka 1.1: Formalizovaná data

Jméno	Hodnota	
Počet lodí v Adenském zálivu ročně	20 000 lodí	
Rychlost pirátských lodí	25-30 uzlů	
Počet útoků za dne / Počet útoků celkem	90%	
Počet plavidel	– lodí	1-2 menší lodě
	– člunů	3-5 člunů + mateřská loď
Délka útoku	v Adenském zálivu	30 minut
	v Somálské pánvi	1 hodina
Riziko útoku	v Adenském zálivu	1%
	v Somálské pánvi	1,4%
Úspěšnost útoku	v Adenském zálivu	20,8%
	v Somálské pánvi	27,3%

Důležitá hodnota je také vzdálenost horizontu, která byla použita pro určení maximální viditelné vzdálenosti lodi (hodnota je zjednodušená tím, že se zanedbává výška pozorované lodi). Vypočtené hodnoty pro typické výšky lodí jsou v tabulce 1.2. Pro spočítání hodnot v této tabulce byl odvozen vzorec (1.1) pomocí Pythagorovy věty podle obr. 1.4. Pro použití tohoto vzorce je nutné použít poloměr Země $R = 6\,378$ km.

$$\begin{aligned}
 d &= \sqrt{(h+r)^2 - r^2} \\
 d &= \sqrt{h^2 + 2hr + r^2 - r^2} \\
 d &= \sqrt{h(h+2r)}
 \end{aligned}
 \tag{1.1}$$



Obrázek 1.4: Znázornění pro odvození vzdálenosti horizontu

Tabulka 1.2: Vzdálenost horizontu v závislosti na výšce pozorovatele

Výška pozorovatele (m)	Vzdálenost horizontu (km)
1,8 (člověk, člun)	4,8
4 (rybářská loď)	7,1
15 (mateřská loď)	13,4

1.3 AgentC

Tato práce byla postavena na multi-agentní platformě AgentC [7]. Schéma platformy je možné vidět na obr. 1.5. Z tohoto schématu je patrné, že architektura tohoto systému je postavena z několika modulů. Moduly jsou rozděleny do těchto skupin:

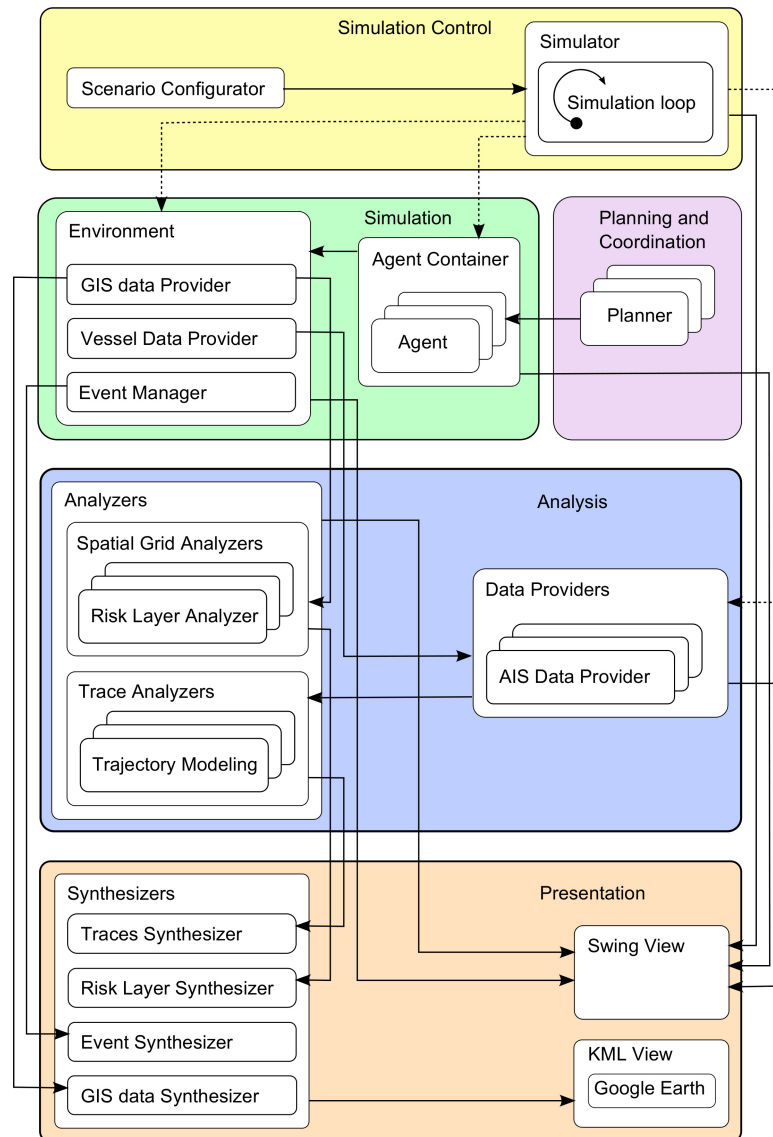
- **Řízení simulace (Simulation Control)** – moduly této skupiny se starají o řízení, inicializaci a spouštění simulace a všech potřebných procesů pro běh simulace.
- **Simulace (Simulation)** – zde jsou moduly starající se o reprezentaci a provoz simulovaného modelu, všech simulovaných plavidel a simulačního prostředí, ve kterém jsou tyto modely a tato plavidla provozovány.
- **Analýza (Analysis)** – obsahuje moduly, které pomocí analýz zpracovávají data, která jsou získána pomocí simulace. Také je v této skupině obsažen modul Zdrojů dat (Data Providers), které má za úkol poskytování dat, které odpovídají reálnému prostředí a to tím způsobem, že přidávají šum a výpadky různých hodnot.
- **Plánování a koordinace (Planning and Coordination)** – zde jsou obsaženy moduly, které zajišťují složitější a komplexnější koordinaci a plánování oproti základnímu chování plavidel, které je implementováno přímo ve skupině Simulace.
- **Vizualizace (Presentation)** – obsahuje moduly, které vizualizují stav simulace a analýzy. Důležitou částí této skupiny je také modul Synthesizer, který kombinuje různé datové vstupy za účelem vizualizace v aplikaci Google Earth a to pomocí souborů ve formátu KML. Také tato skupina obsahuje rozhraní pro kontrolu simulace a zobrazování aktuálních hodnot, které je realizováno pomocí Swing knihoven jazyka Java.

1.3.1 Zavádění modulů pro simulaci a tok dat

Výše uvedené moduly se zavádí pomocí Groovy² skriptů. Ty umožňují změnu parametrů bez kompilace zdrojového kódu do byte-kódu. V těchto skriptech je možné měnit počet kroků simulace, délku pauzy mezi kroky a další proměnné ovlivňující běh simulace. Také se tyto skripty starají o zavádění modulů zmíněných výše. Zejména jsou to moduly pro prostředí (Environment), pro Agent Container (obsahuje jednotlivé agenty), Synthesizer (stará se o tvorbu datových souborů KML pro Google Earth) a další moduly, které jsou pro danou simulaci potřebné.

Zavedená simulace se provádí pomocí časových skoků (tiků), které mají předem danou délku. Skupina Řízení simulace přidělí prostředí (Environment) a agentovi z Agent Container daný časový úsek, po který tento agent v daném stavu provádí určitou akci (typicky plutí na danou pozici či zjišťování lodí v okolí). Provedená akce provede obvykle změnu polohy nebo jiného stavu lodi a tato změna se zapíše do prostředí (Environment). Změna polohy je zapsána do Vessel Data Provider (zdroj

²Groovy – dynamický programovací jazyk pro Java VM, <http://groovy.codehaus.org/>



Obrázek 1.5: Schéma vnitřní struktury platformy AgentC [7]

dat spravující lodě a jejich upluté trajektorie), aby mohla být později použita např. pro vizualizaci na mapě. Také prostředí (Environment) obsahuje Event Manager, který spravuje události, jež lodě vytvoří a ty zalogue a také podle nich udělá předem určenou akci (např. přejde do jiného stavu). Z prostředí čerpá také modul Zdrojů dat (Data Providers), který v případě změny v prostředí změní svoji vnitřní reprezentaci simulace a upozorní další moduly na změnu (zejména ze skupiny Analýza).

1.3.2 Běh simulace

V této sekci budou detailněji popsány moduly skupin Simulace a Řízení simulace, jejich funkce a závislosti na dalších modulech.

Parametry simulace

Hlavními parametry pro tyto moduly jsou délka simulačního kroku (simulation step size) a prodleva mezi simulačními kroky (simulation step delay). Parametr délka simulačního kroku udává kolik času je přiděleno na jeden krok simulace (je to virtuální čas a vůbec nesouvisí s reálnými hodinami). Má-li tento parametr větší hodnotu, pak je výsledek simulace „hrubší“. Je-li tomu naopak, tak je výsledkem detailnější průběh simulace. Parametr prodleva mezi simulačními kroky má pouze roli pro vizualizaci, protože po dobu, která odpovídá nastavené hodnotě tohoto parametru, simulace neprovádí. Je-li tento parametr větší, tak můžeme sledovat průběh simulace „krok po kroku“, v opačném případě se nám vykreslí v závislosti na našem hardwaru třeba jen každý desátý krok simulace. Poslední parametr, který je zejména určen pro automatizované simulace, je délka simulace (simulation duration). Tento parametr udává kolik kroků se má provést, než se simulace sama ukončí.

Virtuální hodiny simulace jsou aktualizovány v modulu prostředí (Environment) a to od nastavené počáteční hodnoty přičítáním délky simulačního kroku. Tímto způsobem může agent zjistit virtuální čas při provádění akce např. pro zjištění západu slunce.

Scénáře

Scénáře se generují pro transportní, armádní, pirátské a rybářské lodě resp. agenty. Při generování těchto scénářů se zároveň vytváří a registrují v prostředí jednotlivá plavidla. U transportních plavidel se inicializuje scénář pro tuto loď, přiřadí se název, počáteční a koncový přístav, cesta kudy bude loď plout a rychlost jejího plutí. U armádních lodí se jen nastavuje název a dva body, mezi kterými bude tato loď hlídkovat. Pirátský scénář pak obsahuje název, domovský přístav, počet plavidel, rychlost a oblast, která je agentovi přidělena. U rybářské lodě se nastaví oblast pro rybaření a parametry této lodě.

Cyklus simulace

V každém kroku simulace se provádí následující sekvence:

1. Modul prostředí (Environment) zaktualizuje svůj simulační čas.
2. Agent Container je upozorněn a sekvenčně posílá informaci o novém kroku každému agentovi. Každý jednotlivý agent spotřebuje celý nebo část tohoto intervalu na provedení akce (plutí, hledání lodí, ...). Vygenerované události (Events) jsou posílány do modulu Agent Container, který je přeměruje ke správnému agentovi a také zapíše do prostředí. V případě, že loď nespotřebuje celý časový interval, tak buď použije zbytek času na provedení dalšího stavu, nebo po tuto dobu nic neprovádí.

3. Modul Zdrojů dat (Data Providers) je upozorněn na nová data v prostředí (vygenerovaná v tomto kroku) a zpracuje je a když jsou tato data dostupná, tak je pošle modulu Analýza pro další zpracování.

Rozhraní pro řízení simulace

Z grafického rozhraní Simulátoru je možné upravovat hodnoty simulace za běhu. Lze simulaci zastavit a znovu spustit, zvětšit či zmenšit prodlevu mezi kroky simulace a také prodloužit nebo zkrátit délku simulačního kroku. Zároveň v tomto rozhraní je možné vidět aktuální krok simulace, vygenerované události a další informace užitečné jak pro vývoj, tak pro běh simulace.

1.3.3 Prostředí (Environment)

Model prostředí (Environment) obsahuje simulovanou reprezentaci reálného světa. Poskytuje zdroje dat a ukládá akce vykonané agenty.

Prostředí obsahuje následující moduly:

- **Zdroj dat GIS** – poskytuje geografická data dalším modulům. Jsou to zejména tato data: somálské přístavy, rybářské oblasti, oblasti působení pirátů, IRTC koridor a mapu útoků na transportní loď.
- **Zdroj dat lodí** – poskytuje databázi informací o lodích jako např. rychlost, rozměry, identifikaci lodě, atd. Také poskytuje dynamicky aktualizované informace o trajektoriích lodí.
- **Historie** – uchovává všechny události, které byly vygenerovány v průběhu simulace.
- **Zobrazení** – poskytuje vizualizaci statických informací plavidel do uživatelského rozhraní.

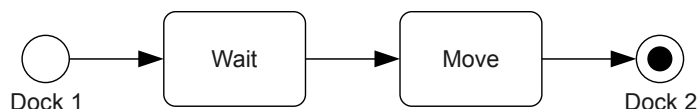
1.3.4 Agenti

Chování agentů je modelováno pomocí konečných stavových automatů (FSM). Více o implementaci FSM je v sekci 3.1 (Popis implementace).

V platformě AgentC jsou implementovány FSM pro transportní loď, pro rybářské lodě, pro piráty a pro armádní lodě. V této práci byly používány pouze FSM pro transportní, pirátské a armádní lodě.

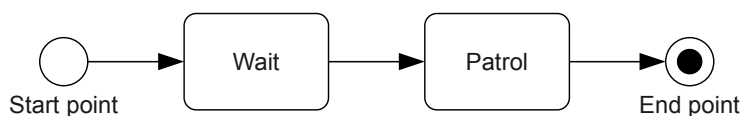
Schéma FSM transportní lodě v UML je na obr. 1.6. Z něj je možné vidět, že na začátku loď čeká (stav Wait), až bude moct vyplout. To je z důvodu, že se zavádí

všechny lodě najednou a takto se rozloží rovnoměrně v čase. Po ukončení čekání loď přejde do stavu Move, kdy již pluje po předem vygenerované trajektorii do druhého přístavu.



Obrázek 1.6: UML diagram FSM transportní lodě

Schéma FSM armádní lodě v UML je na obr. 1.7. FSM armádní lodě je v principu totožné s transportní lodí až na to, že místo stavu Move je zde stav Patrol. Ve stavu Patrol loď pluje po vygenerovaných trajektoriích tak, aby minimalizovala riziko útoku piráta na transportní loď.



Obrázek 1.7: UML diagram FSM armádní lodě

1.3.5 Zdroje dat

Zdroje dat (Data Providers) jsou použity pro reálnější získávání dat o jiných plavidlech. Poskytují obvykle zašuměný nebo hrubší obraz o situaci na moři oproti modelu prostředí (Environment), kde jsou uložena přesná data. Tato „horší“ data jsou určena zejména pro zjišťování vhodných transportních lodí, na které by mohli piráti útočit. Data jsou pak zatížena např. výpadky informace.

1.3.6 Vizualizace v Google Earth

Pro vizualizaci simulovaných dat se používá aplikace Google Earth³ od stejnojmenné firmy Google. Tato aplikace umožňuje 3D zobrazení planety Země s možností přiblížení jednotlivých oblastí, až i dokonce jednotlivých lidí například na náměstích větších měst. Pro zobrazení různých informací na mapě se používá značkovací jazyk KML (Keyhole Markup Language) [10]. Tento jazyk je založen na jazyku XML. Distribuce souborů v jazyce KML se provádí buď v nekomprimované podobě s příponou souboru .kml nebo v komprimované podobě pomocí algoritmu ZIP a pak má tedy soubor příponu .kmz.

³Google Earth – <http://earth.google.com/>

Pro získávání automaticky aktualizovaných dat do aplikace Google Earth se pak používá interní jednoduchý web-server v projektu AIS, který na vyžádání aplikace sestaví požadovaný KML soubor a zašle ho aplikaci, která ho vykreslí. Toto se děje přes standardní implementaci HTTP protokolu. Data jsou tedy periodicky aktualizována tímto způsobem s uživatelsky definovaným časovým intervalem. Je zde možné zobrazit vrstvu lodí (pozice lodí a jejich trajektorie), vrstvu událostí, vrstvu risk-mapy a vrstvu GIS, která obsahuje informaci o přístavech.

Kapitola 2

Agentní modely

V této kapitole jsou nejdříve v první sekci popsány jednotlivé strategie, které byly implementovány. V poslední sekci jsou tyto strategie zformalizovány pomocí UML diagramů.

2.1 Jednotlivé strategie a FSM modely

Agentní modely pirátských strategií byly v této práci vytvořeny pomocí konečných stavových automatů (FSM). Konečný stavový automat je popis chování programu pomocí konečného počtu stavů, přechodů mezi stavy a akcemi, které vyvolávají změny stavu modelu. Takovýto popis také musí mít počáteční a koncový stav (definice FSM – viz sekce 1.2.1).

Modelovány byly následující pirátské strategie:

- **No Info** – Strategie neinformovaného hledání lodí, kdy pirát má pouze dalekohled pro hledání transportních lodí.
- **Radar** – Tato strategie rozšiřuje předchozí typ přidáním radaru.
- **Mothership + Radar** – Strategie s mateřskou lodí a radarem používá při útoku rychlejší čluny, které vyplují z mateřské lodi. Strategie je jinak totožná se strategií Radar.
- **AIS** – Piráti v této strategii používají Zdroj dat (sekce 1.3.5 na str. 12) a pomocí analýzy se snaží vytipovanou loď přímo přepadnout bez náhodného hledání.
- **Mothership + AIS** – Totožná strategie s AIS, avšak piráti používají mateřskou loď a čluny.

- **Learnable** – Tato strategie rozšiřuje strategii No Info tím, že se pirát učí najít místa vhodná pro další útok a to způsobem, že se vrací na úspěšná místa útoků.
- **Bandit** – Strategie Bandit implementuje do strategie No Info herní algoritmus multi-armed bandit [16]. Zde se již pirát nezaměřuje na úspěšná místa útoků, ale na větší oblasti.

2.1.1 Strategie No Info

Pirátská strategie No Info pracuje na náhodném principu hledání lodi bez jakýchkoliv informací, které se získávají jinak než dalekohledem. Piráti plují po náhodné trajektorii a v průběhu plutí sledují okolí vymezené vzdáleností horizontu při pozorování dalekohledem. Zpozorují-li v tomto okolí transportní loď, tak na ní začnou útočit. Pokud piráti zaregistrují během útoku, že je v blízkosti armádní loď, tak se budou snažit utéct pryč a útok přerušit. V případě úspěšného útoku transportní loď odvezou do svého přístavu a zde počkají na výkupné. Bude-li útok neúspěšný, tak se vrátí do stavu, kdy budou plout a hledat další loď. Toto plutí může ukončit buď čas strávený na moři a s tím spojené spotřebování paliva, západ slunce, jelikož v noci neútočí nebo špatné počasí, které se však v oblasti Somálska moc nevyskytuje.

Nalezení transportní i armádní lodi je zatíženo určitým šumem, kdy se třeba pozorovatel může přehlédnout. Tento šum je pak zahrnut do pravděpodobnosti zpozorování lodi. Dále pak je pirátská loď úspěšná s určitou pravděpodobností, která je rovněž brána v úvahu. Poslední uvažovaná pravděpodobnost je spojená s nočním hledáním lodí, kdy v noci mají piráti stížené pozorovací schopnosti.

2.1.2 Strategie Radar, Mothership + Radar

Strategie Radar a Mothership + Radar rozvíjejí předchozí strategii No Info tím, že přidávají do stavu, kdy loď pluje a hledá dalekohledem transportní loď, hledání lodí radarem. V případě nalezení lodě na radaru se k ní začnou přibližovat. Až jí mají na dohled, tak přejdou do útoku. V případě, že piráti používají mateřskou loď, tak při útoku používají rychlejší čluny, které se odpoutají od mateřské lodi. V případě útěku se ale nejdříve musí spojit s mateřskou lodí z důvodu malého množství paliva pro motory člunů.

Opět jsou zde zahrnuty stejné pravděpodobnosti jednotlivých akcí. Navíc je zde zahrnut šum radaru, kdy piráti nemusí úspěšně zpozorovat transportní loď.

2.1.3 Strategie AIS, Mothership + AIS

Strategie AIS je pojmenována podle zdroje dat o poloze lodí, kterým je Automatický identifikační systém (Automatic Identification System). Piráti v této strate-

gii již neplují náhodně po moři, ale získávají informace o poloze, rychlosti a vlastnosti proplovajících lodí. V případě nalezení vhodné lodi pro přepadení vyplují na místo budoucího setkání, kde buď počkají, nebo loď už přímo zpozorují a potom přejdou do stavu útočení. V případě úspěchu je již postup stejný jako v předchozích strategiích. Avšak neuspějí-li, tak plují zpět domů, kde zase čekají na další vhodné plavidlo.

V této strategii je pro získávání dat použit velice zašuměný zdroj dat (sekce 1.3.5, str. 12). V tomto zdroji dat jsou výpadky jednotlivých hodnot i plavidel. Proto si musí piráti držet určitou malou historii, aby byli schopni zjistit vhodnou loď. Toto zjištění je také ještě zatíženo určitou chybou, která je modelována pravděpodobností pro zpozorování lodí.

Tato strategie opět zahrnuje útěk pirátů v případě zpozorování armádní lodě. Použití mateřské lodě se opět projeví jen u útoku a útěku pirátů. V této strategii je oproti předchozím strategiím výhoda, že nemusí plout za dne, protože nepotřebují loď hledat, jelikož jí mají již vytipovanou.

2.1.4 Strategie Learnable

Tato strategie rozvíjí typ No Info tím, že si piráti pamatují body, kde úspěšně či neúspěšně útočili. Zároveň v případě jakéhokoliv útoku navýší hodnotu pravděpodobnosti, se kterou se vrací na místa předchozích útoků. Neuspějí-li však, tak se tato pravděpodobnost exponenciálně snižuje.

Piráti nejdříve plují (objevují) jako ve strategii No Info. V případě nějakého útoku se jim navýší již zmíněná hodnota pravděpodobnosti. V dalším cyklu se již s touto pravděpodobností rozhodují mezi objevováním a zužitkováním (Exploration \times Exploitation). V případě zužitkování piráti plují v okolí již navštívených bodů, kde předtím útočili, a postupně je periodicky prochází.

Piráti pomocí této strategie by se měli přizpůsobovat trajektoriím transportních lodí a postupně plout jen na místa, kde se tyto lodě vyskytují a neplout jen náhodně.

2.1.5 Strategie Bandit (multi-armed bandit)

Strategie Bandit je podobná předchozí strategii Learnable, avšak zde je implementován herní algoritmus multi-armed bandit konkrétně ve variantě ϵ -decreasing strategy (je to jeden z typů hladové strategie ϵ -greedy) [16]. Teoretický popis této herní strategie je níže v následující podsekci.

Tento algoritmus také pracuje na principu objevování a zužitkování (Exploration \times Exploitation), ale s tím, že poměr mezi objevováním a zužitkováním se mění v čase a ne v závislosti na předchozích výsledcích. Prostor moře určený pro

piráta je v této strategii rozdělen na menší pole a každému poli je přidělena určitá pravděpodobnost definující míru úspěšnosti nalezení lodi pro útok. V případě, že pirát zužitkovává znalosti, tak pluje do prostoru, který má maximální pravděpodobnost z dané množiny. Je-li zde více prostorů s danou pravděpodobností, tak pluje do náhodného prostoru s maximální pravděpodobností.

Je-li pirát ve stavu objevování, tak v případě útoku na loď, nalezne příslušný prostor, který obsahuje místo útoku, a zvýší mu hodnotu pravděpodobnosti. Když je pirát ve stavu zužitkování znalostí, tak pluje do daného prostoru, kde hledá transportní loď. V případě, že za celý cyklus nenalezne žádnou loď, tak se exponenciálně sníží hodnota pravděpodobnosti daného prostoru. Nalezne-li loď, tak zvýšení pravděpodobnosti proběhne jako ve stavu objevování.

Objevování lodí, útěk před armádní lodí a způsob útoku se v této strategii jinak neliší od strategie No Info a to včetně příslušných pravděpodobností, které modelují různý šum a nejistoty.

Herní strategie multi-armed bandit

Herní strategie multi-armed bandit [16] je příklad obecného problému. Často se také tato strategie nazývá K -armed bandit a je podobná tradičnímu hernímu automatu, která představuje one-armed bandit strategii, ale má více herních pák. Každá páka má přiřazenu jednu distribuční funkci, která vyjadřuje pravděpodobnost úspěchu. Zpočátku hráč nemá žádnou informaci o rozložení pravděpodobností úspěchu přes všechny páky, tak je nejdříve testuje, aby získal znalosti, a pak se zaměří už jen na páky, které mu poskytují největší zužitkování znalostí.

Multi-armed bandit strategii si lze představit jako skupinu výnosových (*reward*) distribučních funkcí $B = \{R_1, \dots, R_K\}$, kde K je počet herních pák. Nechť μ_1, \dots, μ_K jsou průměrné hodnoty výnosových distribučních funkcí. Hráč aktivně zatáhne za jednu páku každé kolo a sleduje příslušný výnos. Jeho úkolem je maximalizovat součet těchto výnosů. Oželení (*regret*) ρ po T kolech je definováno jako rozdíl mezi součtem výnosů příslušných výnosových funkcí optimální strategie a součtu získaných výnosů $\rho = T\mu^* - \sum_{t=1}^T \hat{r}_t$, kde μ^* je průměrný optimální výnos, $\mu^* = \max_k \{\mu_k\}$ a \hat{r}_t je výnos v čase t .

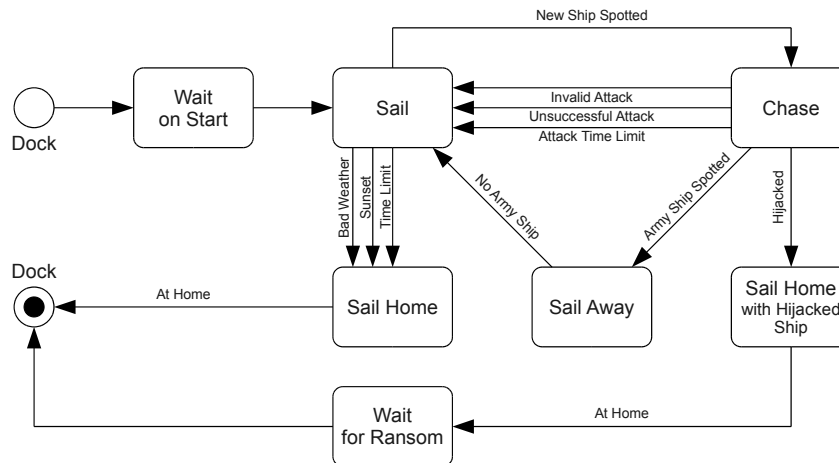
Zvolený typ varianty strategie multi-armed bandit ϵ -decreasing strategie je přirozená varianta ϵ -greedy strategie. Strategie ϵ -decreasing je založena na zmenšování hodnoty ϵ , kvůli asymptotickému přiblížení se optimální strategii. S pravděpodobností ϵ_t , kde t je číslo kola, je zatáhnuto za páku s maximálním získaným průměrem, v opačném případě je zatáhnuto za náhodou páku. Hodnota ϵ_t je definována jako $\epsilon_t = \min \left\{ 1, \frac{\epsilon_0}{t} \right\}$, kde $\epsilon_0 > 0$. Hodnota ϵ_0 je zvolena libovolně v závislosti na použití.

2.2 UML diagramy FSM modelů

V této kapitole více zformalizuji modely popsané v předchozí sekci Jednotlivé strategie a FSM modely (str. 14) a to pomocí UML diagramů konečných stavových automatů. Pro zjednodušení budu používat pouze stavy a přechody. Také zde již budu uvádět konkrétní hodnoty jednotlivých parametrů, které vychází ze sekce 1.2.3.

2.2.1 Strategie No Info a strategie Bandit

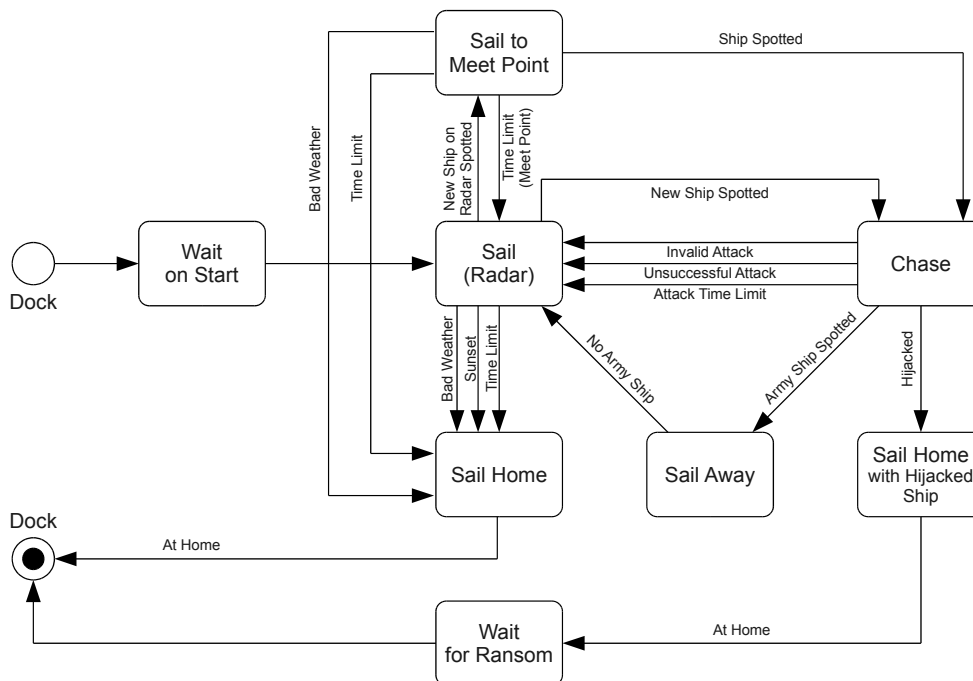
Tyto strategie jsou znázorněny na obr. 2.1. Strategie mají stejný diagram z důvodu stejného systému plutí. Liší se však v implementaci agentního modelu, což je detailně popsáno v sekci 3.2 (Implementace strategií).



Obrázek 2.1: UML diagram strategií No Info a Bandit

Na obrázku je vidět počáteční stav Wait on Start, v němž pirát čeká na vyplutí. Je to z důvodu, aby všechny lodě z přístavů nevypluly na ráz, ale aby se rozprostřely rovnoměrně v čase. Dále je zde stav Sail, kdy loď pluje po moři a hledá transportní loď k přepadení a to v okruhu 7 km. Z tohoto stavu vedou přechody Bad Weather, Sunset a Time Limit, které stavu určují, kdy má přejít do stavu Sail Home, a kdy začne plout domů. Tyto přechody jsou pro špatné počasí, západ slunce a také pro vyčerpání doby plutí, která je nastavena na 1 den pro Adenský záliv a na 3 dny pro Somálskou pánev. Také ze stavu Sail vede přechod New Ship Spotted, po kterém přejde pirát do stavu Chase (pronásledování) v případě, že spatří transportní loď. V případě neúspěšného pronásledování se pirát vrací do stavu Sail. Tento neúspěch nastane, když vyčerpá maximální čas útoku (Attack Time Limit), je-li útok neúspěšný (Unsuccessful Attack) nebo je-li útok neplatný (Invalid Attack). Účel neplatného útoku je popsán v sekci 3.2. Maximální čas útoku je nastaven na 30 minut v Adenském zálivu a na 1 hodinu v Somálské pánvi. Pokud piráti spatří armádní loď, tak přejdou do stavu Sail Army, ve kterém utíkají armádní lodi. Až jí nemají

na dohled, tak zase pirátská loď přejde do stavu Sail. Je-li pronásledování a útok úspěšný, tak pirát přejde do stavu Sail Home with Hijacked Ship, kde s uloupenou lodí odpluje domů, kde čeká na výkupné (stav Wait for Ransom). Tímto se uzavírá cyklus pirátské lodi.

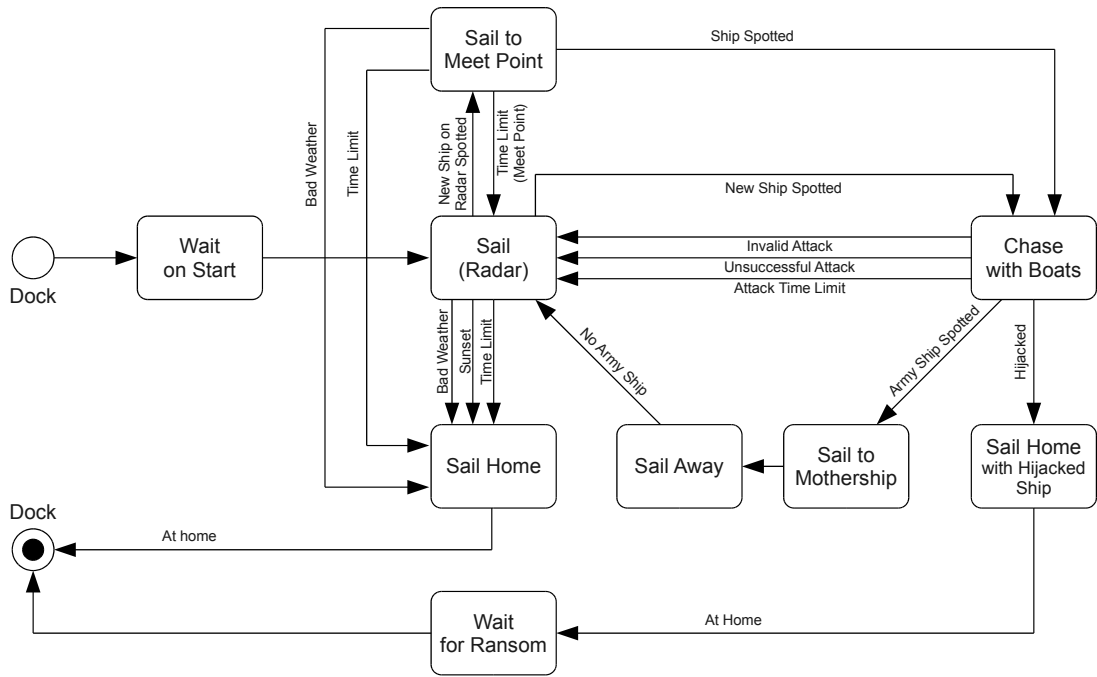


Obrázek 2.2: UML diagram strategie Radar

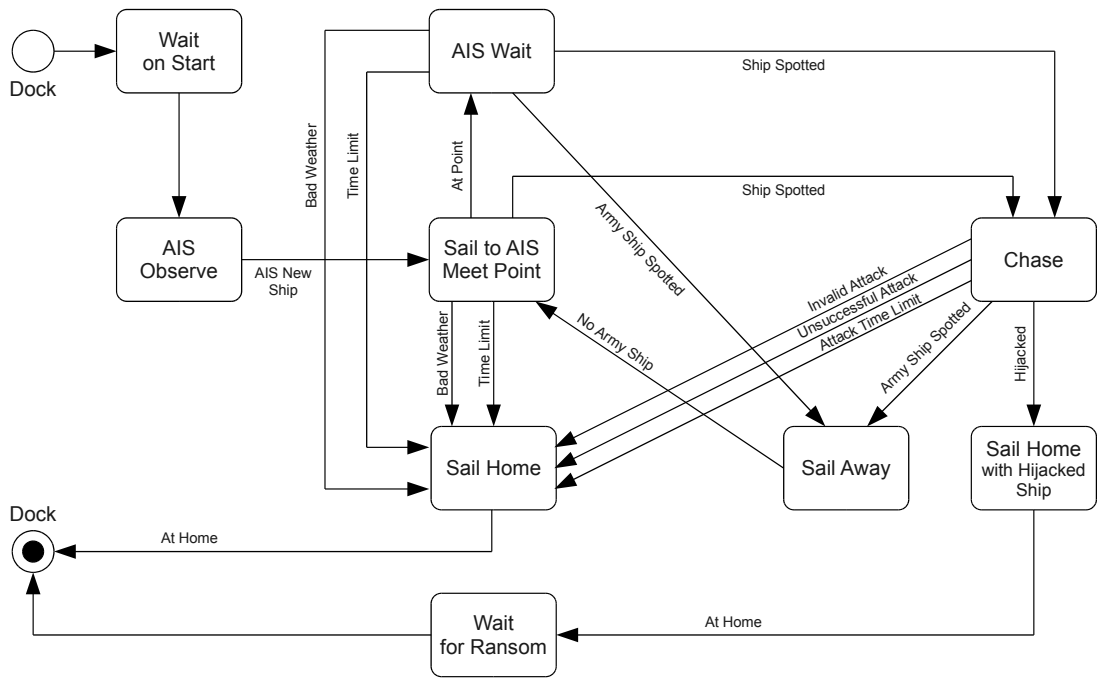
2.2.2 Strategie Radar, Mothership + Radar

Strategie Radar (viz obr. 2.2) má jádro stejné jako předchozí strategie No Info, avšak přibyl zde stav Sail to Meet Point (plutí na bod setkání), který reflektuje přítomnost radaru na lodi, a tedy i potřeby přiblížení lodi na dohled před případným útokem. Také jsou zde zahrnuty přechody prakticky stejné jako ve stavu Sail u strategie No Info. Stav Sail (Radar) je rozšířením stavu Sail u No Info diagramu o pozorování okolních lodí radarem. Pomocí radaru je pirát schopen vidět lodě v okruhu 70 km.

Model s pirátskou mateřskou lodí a radarem na obr. 2.3 se pak od strategie bez mateřské lodi liší jiným stavem pro pronásledování (Chase with Boats), který zahrnuje použití rychlejších člunů při útoku, a také se liší přidáním stavu Sail to Mothership, kdy čluny při spatření armádní lodi musí nejdříve připlout k mateřské lodi.

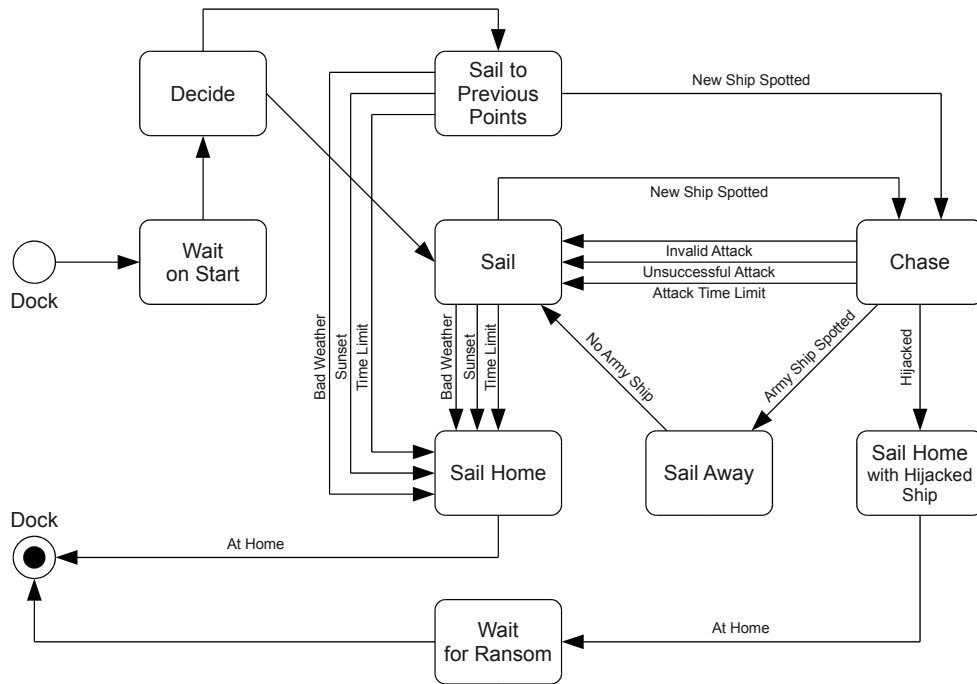


Obrázek 2.3: UML diagram strategie Mothership + Radar



Obrázek 2.4: UML diagram strategie AIS

strategiím a je podobný stavu Sail s tím rozdílem, že zde pirátská loď nepluje po náhodných trajektoriích, ale pluje v okolí bodů, kde již piráti útočili na transportní loď.



Obrázek 2.6: UML diagram strategie Learnable

Kapitola 3

Implementace

V této kapitole bych chtěl popsat implementaci agentních modelů pirátů podle UML diagramů z předchozí kapitoly v programovacím jazyce Java do platformy AgentC.

3.1 Popis implementace

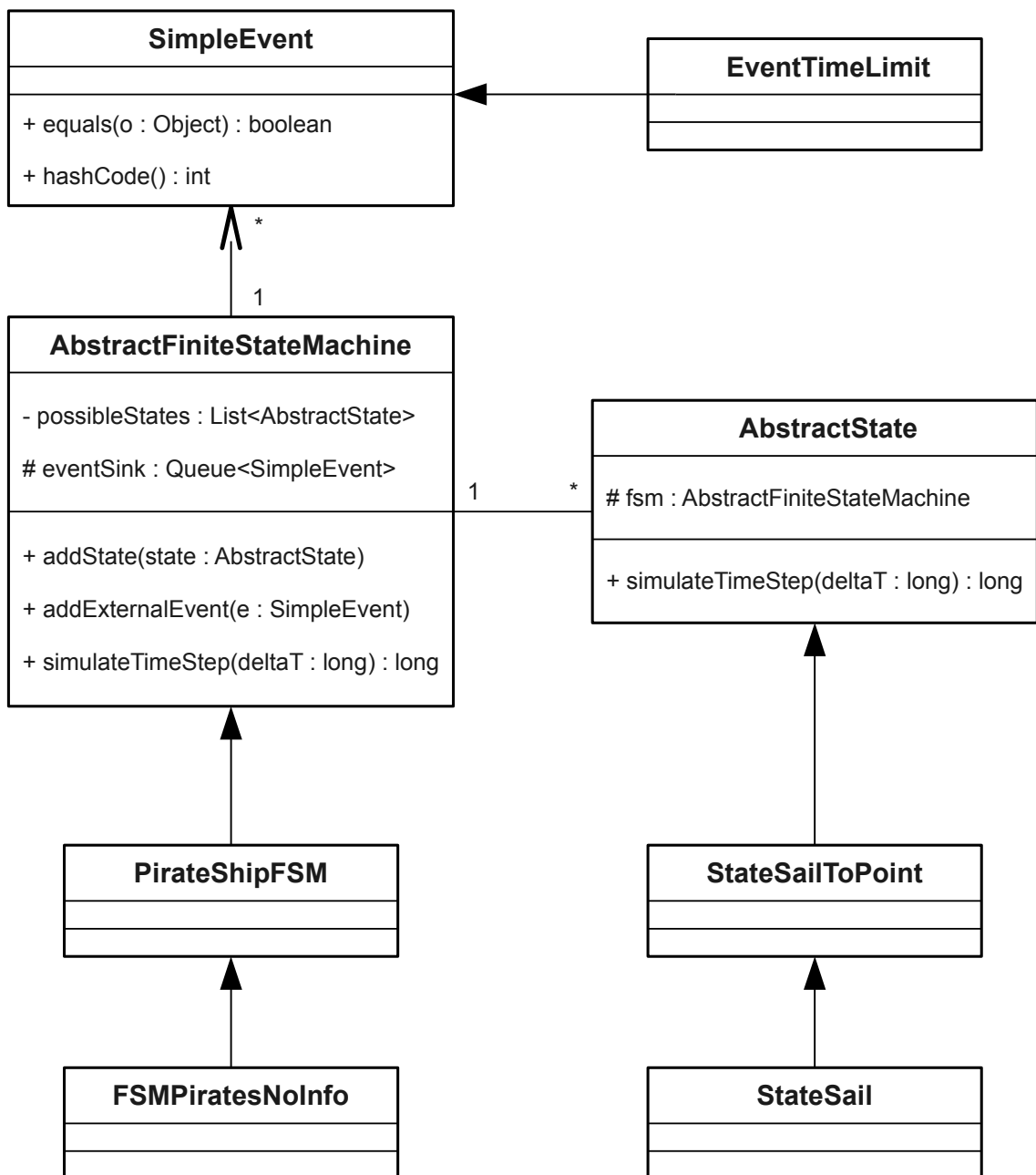
V této sekci bych chtěl popsat princip implementace konečných stavových automatů (FSM) v jazyce Java a zejména v souvislosti s objektově-orientovaným programováním.

Jak již bylo zmíněno v sekci 1.3.2 (Běh simulace), tak simulace probíhá pomocí přidělování časových úseků jednotlivým agentům. Tito agenti jsou implementováni pomocí tříd, které jsou zděděny od třídy `AbstractFiniteStateMachine`. Tato třída přesměruje získaný časový interval aktuálnímu stavu, který je reprezentován třídou zděděnou od třídy `AbstractState`. Pokud se nevyčerpá celý zadaný interval, tak se jeho zbytek přidělí následujícímu stavu za předpokladu, že do něj agent přejde. Přechody ve FSM jsou implementovány jako zděděné třídy událostí od třídy `SimpleEvent` a stavy propojují pomocí mapy, kde jsou uvedeny přechody z daného stavu a cíl, kam směřují, a také je zde mapa stav-stav pro implicitní přechody bez vyvolání akce (události). K přechodu ze stavu dojde buď nevyčerpáním časového intervalu (implicitní přechod) nebo vyvoláním akce (události – zděděná třída od `SimpleEvent`).

Třída implementující stavy FSM je zděděna od třídy `AbstractState`, jak už bylo zmíněno výše, a proto implementuje metodu `long simulateTimeStep(long deltaT)`, ve které se vykonává vlastní činnost ve stavu (např. plutí lodě). Dále se pak při volání této metody provádí instrukce, jejichž vyhodnocení může mít za následek přechod do dalšího stavu (např. vypršení času, nalezení lodě, atd.). Stavy jsou v této práci velice často děděny pro jejich podobnou strukturu a chování.

Přechody a akce jsou v této práci implementovány pomocí tříd událostí zděděných od třídy `SimpleEvent`. Každá takováto třída implementuje metody `boolean equals(Object o)` a `int hashCode()`, které zaručují jednoznačné poznání stejných tříd tak, aby se jich dalo použít pro přechody mezi stavy a akce. Těmito událostmi se také přenáší data mezi stavy a také jsou použity pro statistiku a vizualizaci dat.

Výše uvedeným děděním základních abstraktních tříd z projektu `AgentC` je zajištěna integrace této práce do stávající struktury projektu. Toto je znázorněno pomocí UML diagramu popsanych a zděděných tříd na obr. 3.1.



Obrázek 3.1: UML diagram vybraných základních a zděděných tříd [[4]]

3.2 Implementace strategií

V této sekci se zaměřím detailně na implementaci modelů pirátských strategií a zejména jejich strukturu, která není z UML diagramů patrná.

Všechny třídy, které implementují pirátskou strategii, jsou zděděny od třídy `PirateShipFSM`, která je ještě zděděna od třídy `AbstractFiniteStateMachine`. Ve třídě `PirateShipFSM` jsou definovány konstanty dle formalizovaných dat ze sekce 1.2.3, proměnné nezávislé na strategii a metody, které slouží k nastavování a získávání různých hodnot. Proměnné nezávislé na strategii jsou takové hodnoty, které jsou vhodné pro všechny pirátské strategie. Mezi ně patří zejména různé aktuální časové hodnoty limitů, definovaná oblast pro působení piráta a také například identifikační údaj (ID) unesené lodi.

Strategie `Learnable` a `Bandit` používají podobné nastavení hodnot, o které se zvýší nebo sníží pravděpodobnost definující míru úspěšnosti nalezení lodi pro útok. Tato hodnota je definována jako celé číslo vyjadřující procentní pravděpodobnost. Při neúspěšném útoku se navýší tato hodnota o 5 procentních bodů, v případě úspěšného útoku pak o 10 procentních bodů. Není-li během cyklu proveden žádný útok, tak se tato hodnota vynásobí konstantou 0,9. Tyto hodnoty byly experimentálně stanoveny a ověřeny.

Vnitřní rozdíl v implementaci strategií `No Info` a `Bandit` spočívá v nastavení oblasti pro působení piráta ve třídě `PirateShipFSM`. Ve strategii `Bandit` se narozdíl od strategie `No Info` provede po dokončení cyklu FSM ve stavu `Dock` vyhodnocení předešlé úspěšnosti či neúspěšnosti útoku. V závislosti na výsledku je buď nastavena pro hledání transportních lodí celá oblast (objevování), nebo je nastavena jen podoblast (zžitkování).

Přechod mezi stavy `Invalid Attack` je implementován pouze kvůli omezením modelu a to z důvodu, že je držena v simulačním systému databáze unesených plavidel, která má za úkol zamezit několikanásobnému přepadení stejné lodi po unesení. Jestliže se však dva piráti rozhodnou pronásledovat stejnou loď a první jí úspěšně přepadne, tak druhý pirát není ani úspěšný, ani neúspěšný, ale zároveň nemůže dál pokračovat v útoku. Proto se vyvolá akce `Invalid Attack`, po které pirát přejde do stavu, kdy může zase začít hledat další transportní loď k přepadení.

Kapitola 4

Vyhodnocení

V této kapitole bude popsán postup testování, hodnoty použité pro testování, výsledky a v závěru kapitoly diskuze nad získanými výsledky.

4.1 Popis simulace a scénářů

Simulace byla prováděna pro všechny pirátské strategie, které byly zaneseny do scénářů. Dále pak byly měněny hodnoty klíčových konstant a byl pozorován jejich vliv na výsledky testování. Měněné konstanty byly ze tří skupin: rychlost transportních lodí, vzdálenost horizontu (výška pirátské lodi) a počet pirátských lodí v regionu v okolí pobřeží Somálska. Pro účely testování byly postupně vyhodnoceny všechny kombinace nastavení konstant a scénářů.

Rychlost transportních lodí byla nastavována postupně na hodnoty 13, 16 a 19 uzlů. Vzdálenost horizontu byla nastavována na 7 km a na 14 km. Počet pirátských lodí v regionu byl nastavován postupně na hodnoty 1, 3, 6, 10, 25 a 50 lodí.

4.1.1 Nastavené konstanty pro simulace

V této podsekcí budou podrobně popsány nastavené konstanty použité při testování všech scénářů. Popsané konstanty jsou shrnuty přehledně v tabulce 4.1.

Při všech simulacích byl nastaven počet transportních lodí na 2000. Toto číslo odpovídá přibližně počtu lodí, které proplují Adenským zálivem za 1/10 roku. Tento počet se však v simulaci nevztahuje pouze k Adenskému zálivu, ale k celé simulované oblasti. Dále pak byly inicializovány dvě armádní lodě a to jen z demonstračního účelu, protože rozmisťováním těchto lodí kvůli minimalizaci pirátského útoku na transportní loď se zabývá jiná práce.

Pro nastavení pravděpodobností při generování pseudonáhodných čísel byly použity různé procentní hodnoty. Úspěšnost útoku byla nastavena na 20%, což je

hodnota, při které se počet útoků vztažený k počtu proplutých lodí přibližuje reálné hodnotě. Pravděpodobnost pronásledování lodí za dne byla nastavena na 90%, kde nezahrnutých 10% by mělo odpovídat šumu a špatnému odhadu při vytipování plavidel pro útočení. V noci klesá tato pravděpodobnost na 10% a to z důvodu, že se v noci uskutečňuje málo útoků a také, že se dá o hodně hůře vytipovat loď, kterou chce pirát pronásledovat. Dále pak byla používána pravděpodobnost rozpoznání armádní lodi, která chce omezit útok piráta, a ta byla nastavena na hodnotu 80%. Při zjištění přítomnosti armádní lodi totiž pirát začne „utíkat“. Několikrát se ale podle dostupných novinářských zdrojů stalo, že piráti byli dopadeni těmito ochrannými složkami. Poslední používaná pravděpodobnost se týkala úspěšnosti rozpoznání lodí pomocí strategie tipu na vhodnou transportní loď (AIS). Zde byla nastavena tato úspěšnost na hodnotu 90%, čili z 10% pirát nezačne pronásledovat vytipovanou loď.

Dále byly používány při simulaci různé maximální časové úseky pro jednotlivé stavy či strategie. Jedna z nejlépe určitelných hodnot byla doba maximální délky útoku na pronásledovanou loď. Tato doba byla dle zdrojů nastavena na 20 minut pro Adenský záliv a 1,5 hodiny pro Somálskou pánev. Pro maximální dobu plutí byly také nastaveny hodnoty rozlišené podle prostoru určeného pro výskyt pirátské lodi. Hodnoty byly nastaveny na 1 den pro Adenský záliv a 3 dny pro Somálskou pánev. Také byla nastavena doba, po kterou čeká loď na výkupné. Tato doba byla úmyslně nastavena na 3 dny a to z důvodu realizovatelnosti simulace, která běžela po časovou dobu odpovídající cca 35 dnům (50 000 minut). V reálné situaci je totiž doba čekání na výkupné i měsíc, což je ale doba téměř stejná jako délka simulace. Další časový limit je pro plutí za pronásledovanou lodí viděnou na radaru, či na kterou je získán tip pomocí AIS. Tento limit má délku 3 hodiny pro radar a délku plutí lodí pro strategii AIS. Poslední limitní hodnota je pro čekání na místě na loď, na kterou dostal pirát tip (AIS). Tato doba je 12 hodin. Tento stav se však v simulacích prakticky nepoužívá, a proto ani tato limitní hodnota.

Poslední velkou skupinou konstant byly hodnoty viditelné vzdálenosti pro různé druhy pozorování. Pro neinformovanou loď, která používá pro hledání vhodných transportních lodí k přepadení jen dalekohled, je tato hodnota zhruba 7 kilometrů. Má-li loď radar, tak může pozorovat případné objekty až do vzdálenosti 70 km. Pro scénář, kdy loď vyplouvá až na základě tipu, je tato vzdálenost 500 km. Poslední konstanta je pro nastavení, kdy má pirát přejít do stavu pronásledování ze stavu, kdy loď pronásleduje pomocí radaru. Tato vzdálenost byla nastavena na 700 metrů. Je to simulací zvolená nejvhodnější hodnota, při které simulace funguje rozumně.

Také byly používány další pomocné konstanty, které však sloužily jen pro ladění programu a pro různé pomocné operace. Například konstanta pro plánování cesty, která udává hodnotu po kolika krocích se tato cesta přeplánuje. Toto plánování probíhá jednou za 20 kroků simulace a to z důvodu náročnosti výpočtu a výpočetního omezení procesoru.

Tabulka 4.1: Použité konstanty pro simulace

Jméno konstanty		Hodnota
Počet lodí	– transportních	2 000 lodí
	– armádních	2 lodě
Úspěšnost	útoků	20%
	pronásledování za dne / v noci	90% / 10%
	rozpoznání armádní lodě	80%
	tipu na loď (AIS)	90%
Doba	útoků v Adenském zálivu / Somálské pánvi	20 min / 1,5 h
	plutí v Adenském zálivu / Somálské pánvi	1 den / 3 dny
	čekání na výkupné	3 dny
	pronásledování lodí (Radar, AIS)	3 h
Viditelná vzdálenost	dalekohledem	7 km
	radarem	70 km
	pro tip na loď (AIS)	500 km
	pro přechod z pronásledování do útoku	700 m

4.2 Testování a výsledky

V této sekci se zaměřím na průběh testování a na získaná data z proběhlých simulací.

Testování strategií bylo prováděno vzhledem k výpočetním možnostem a náročnosti běhu simulace průběžně skoro celý týden. Nejdélší doba běhu simulace byla u scénářů s 50 pirátskými loděmi. Simulace pro tuto hodnotu trvala téměř 1 hodinu (konfigurace PC: Intel Core 2 Duo 2,26 GHz, 2 GB DDR3 RAM, 32bit Java na 32bit GNU/Linuxu).

Všechny simulace proběhly 10 000 kroků, což odpovídá při délce kroku 5 min hodnotě 34 dnů 17 h 20 min.

4.2.1 Získané grafy

Simulací různých strategií byly získány grafy, které pokrývají všechny kombinace konstant, které byly měněny. Také pak byla provedena syntéza některých dat a vznikly grafy, které porovnávají získané hodnoty u jednotlivých strategií. Jelikož je získaných grafů velké množství, tak byly vybrány jen některé, jenž jsou nejreprezentativnější.

Počet útoků

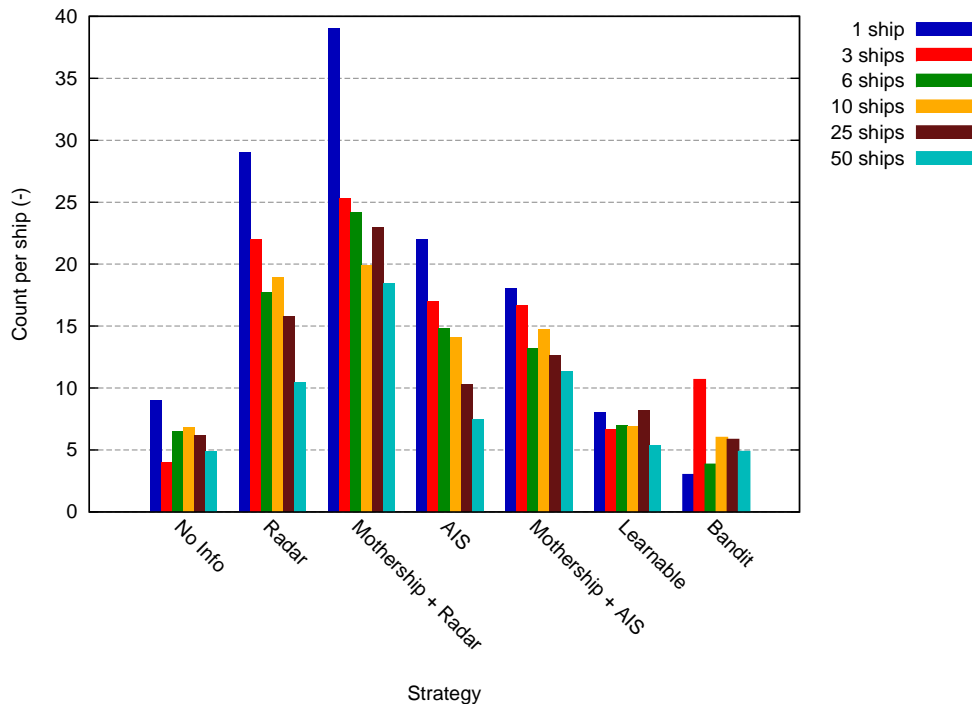
Největší skupinou získaných dat byly grafy pro počet útoků k nějaké veličině.

Z prvního grafu na obr. 4.1 lze vidět kolik útoků průměrně jeden pirát ve strategii provedl na transportní loď. Je zde také vidět, že zejména u strategie AIS a Mothership + AIS dochází k určitému poklesu celkového počtu útoků v případě, že je zde víc pirátských lodí. Toto je způsobeno tím, že více pirátů si vytipuje stejnou transportní loď, ale jen jeden na ní může zaútočit. Určité větší hodnoty u některých strategií jsou způsobeny tím, že se pirát či většina pirátů vyskytovala v Adenském zálivu, kde je větší koncentrace lodí.

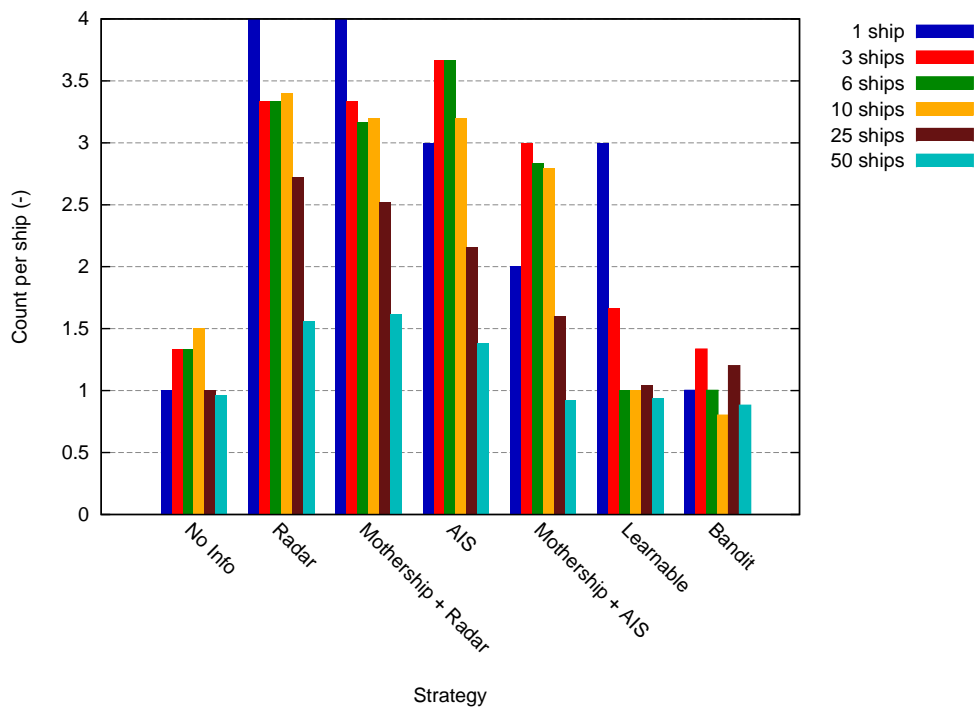
Na obr. 4.2 lze vidět průměrný počet úspěšných útoků jedné pirátské lodi. Oproti obr. 4.1 je zde občas jiný poměr kvantity útoků, což je způsobeno prvkem náhody při určování úspěšnosti útoku.

Graf útoků v závislosti na rychlosti transportních lodí je možné vidět na obrázku 4.3. Je zde vidět, že když se transportní loď zrychlí, tak na ně klesne počet útoků. Největší rozdíl je u strategie Mothership + Radar. U ostatních strategií je pokles patrný, avšak velmi malý. Opět jsou v grafu průměrná čísla na jednoho piráta.

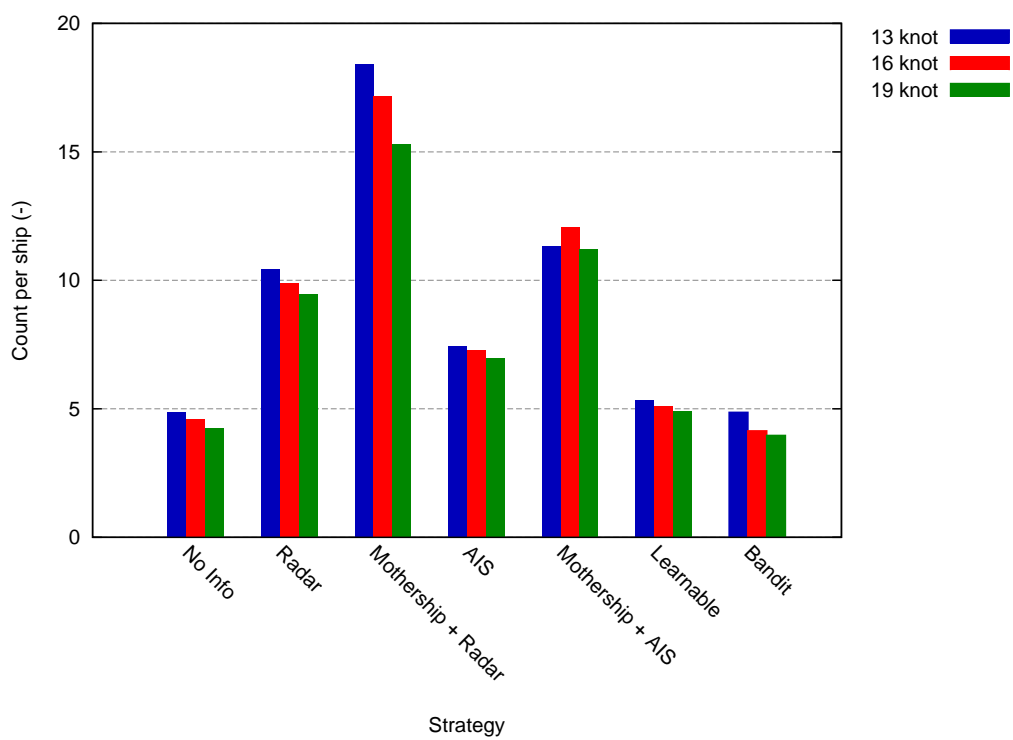
Na obr. 4.4 je pak možné vidět závislost počtu útoků na viditelné vzdálenosti dalekohledem a tedy i na výšce lodi. Zde je patrný rozdíl u strategií, které využívají dalekohled. Těmto pirátským strategiím stoupá počet útoků s viditelnou vzdáleností. U ostatních strategií je spíš rozdíl horší, což může být způsobeno nepřesnostmi v generování polohy lodí (mohou být jinde než pro menší vzdálenost).



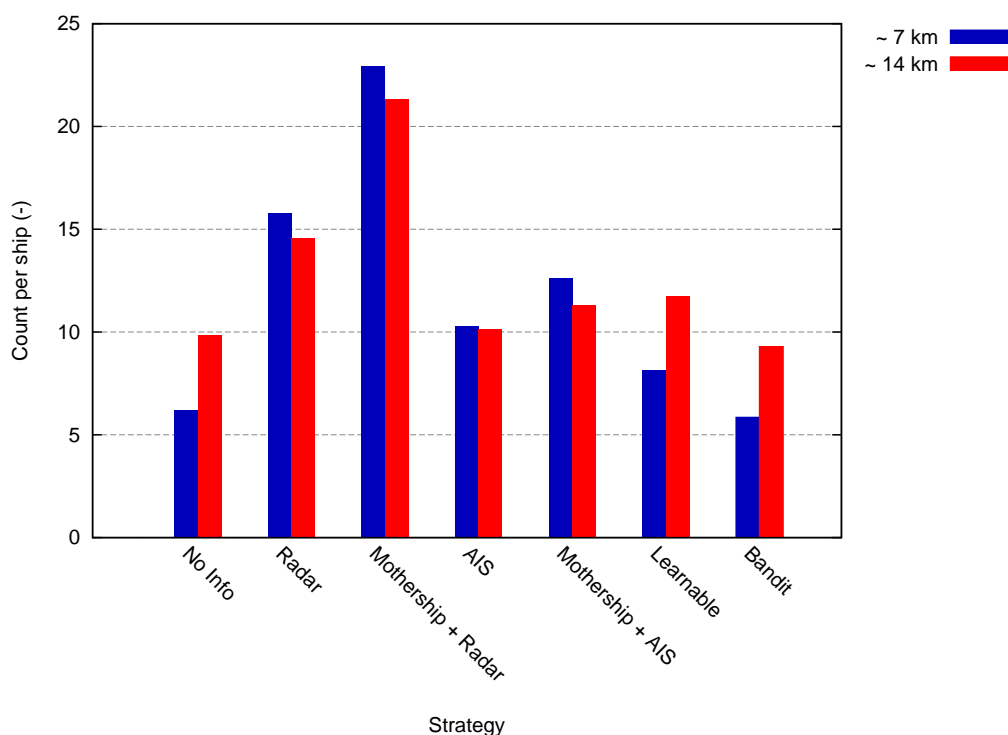
Obrázek 4.1: Graf útoků na transportní loď v závislosti na počtu pirátských lodí



Obrázek 4.2: Graf úspěšných útoků v závislosti na počtu pirátských lodí



Obrázek 4.3: Graf útoků na transportní lodě v závislosti na jejich rychlosti

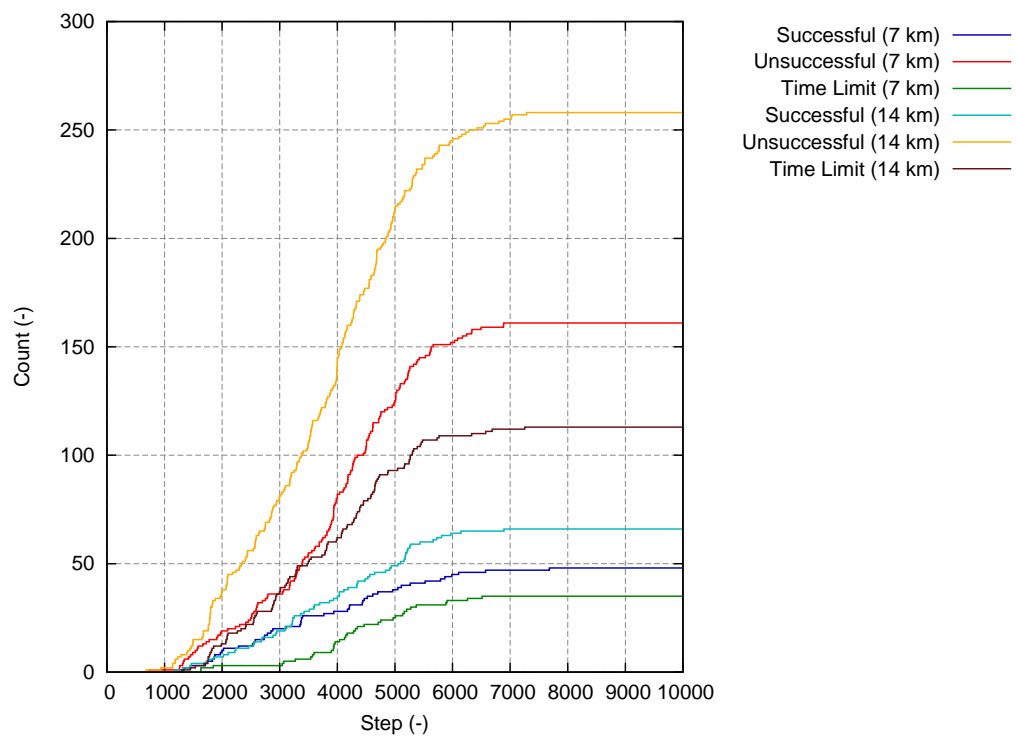


Obrázek 4.4: Graf útoků v závislosti na vzdálenosti horizontu (výšce pirátské lodi)

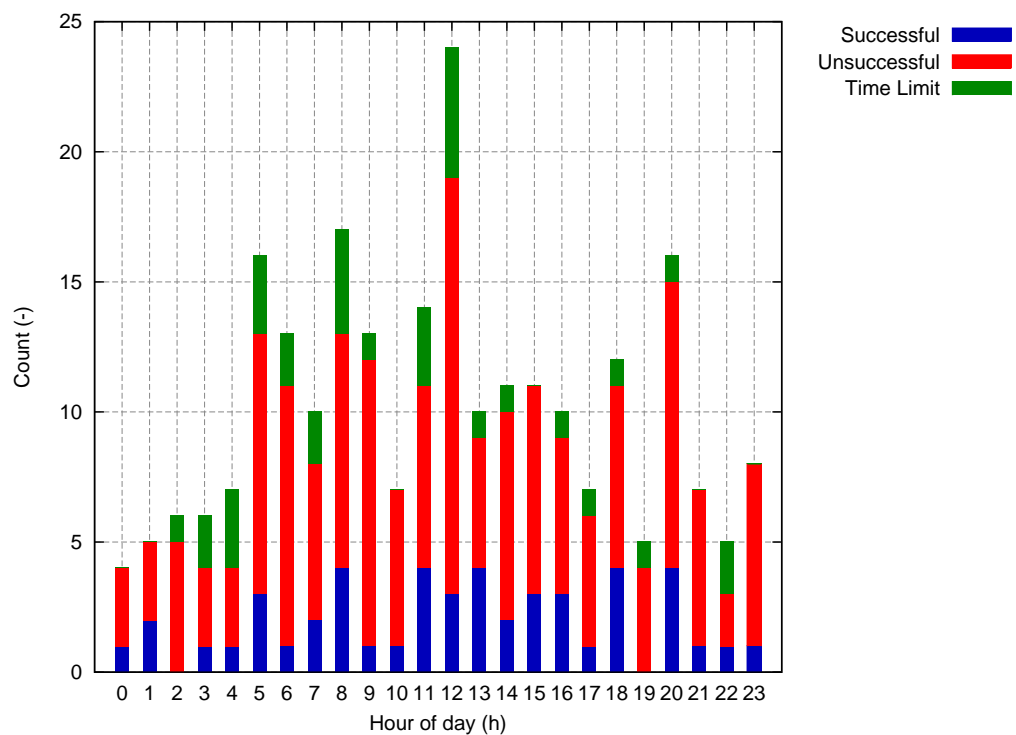
Na obr. 4.5 je vidět, jak se vyvíjí počet jednotlivých útoků (úspěšné, neúspěšné, neúspěšné kvůli vyčerpání časového limitu) v čase simulace a zároveň jsou zde vynešeny tyto průběhy pro vzdálenosti horizontu 7 km a 14 km. Lze vidět nárůst počtu všech typů útoků a největší rozdíl je v nárůstu neúspěšných útoků z důvodu vyčerpání časového limitu. Toto je způsobeno tím, že limit zůstal stejný, ale vzdálenost se prodloužila.

Útoky během dne při strategii No Info lze vidět na obr. 4.6. Z tohoto grafu je patrné, že nejvíce útoků je za denního světla. Toto rozložení je podobné u většiny strategií kromě AIS a Mothership + AIS, kde pirát může vyplout kdykoliv.

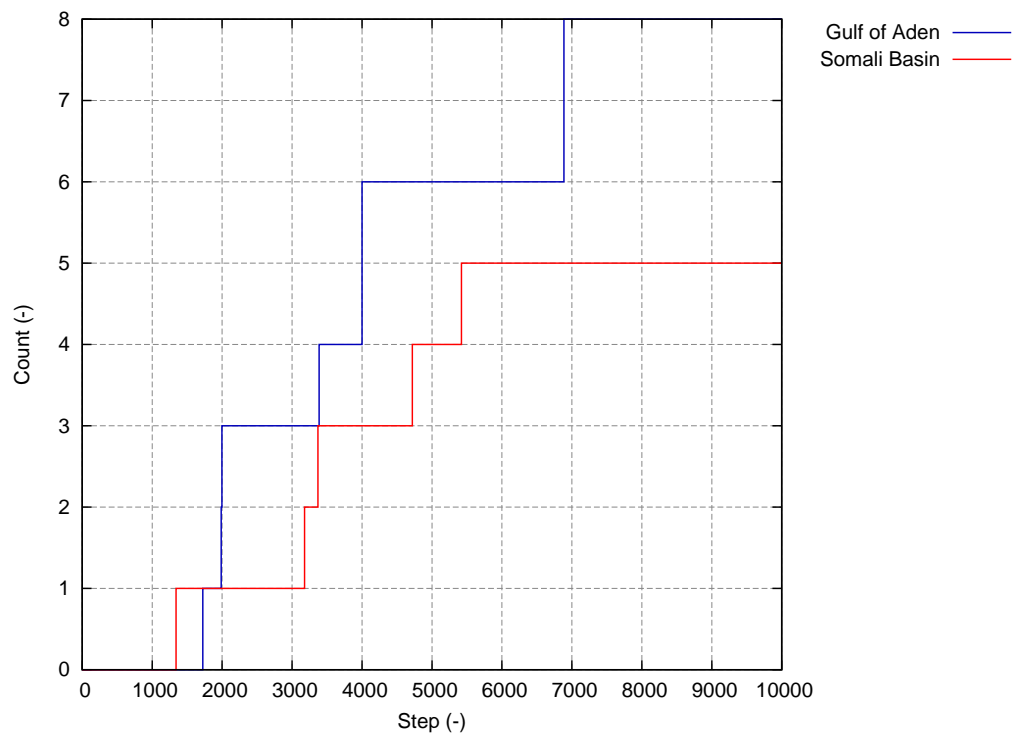
Na obr. 4.7 je vidět rozdíl v celkovém počtu útoků pirátů v čase a v závislosti na místě, kde pirát působí, ve strategii No Info. Je zde vidět předpokládaný rozdíl mezi Adenským zálivem a Somálskou pánví. Tento rozdíl se ještě několikanásobně zvětšuje ve strategii Bandit, což lze vidět na obr. 4.8. Toto je způsobeno tím, že pirát v Adenském zálivu pluje jen k jedné podoblasti, kde plují všechny lodě (IRTC koridor), kdežto v Somálské pánvi je provoz lodí rozprostřen.



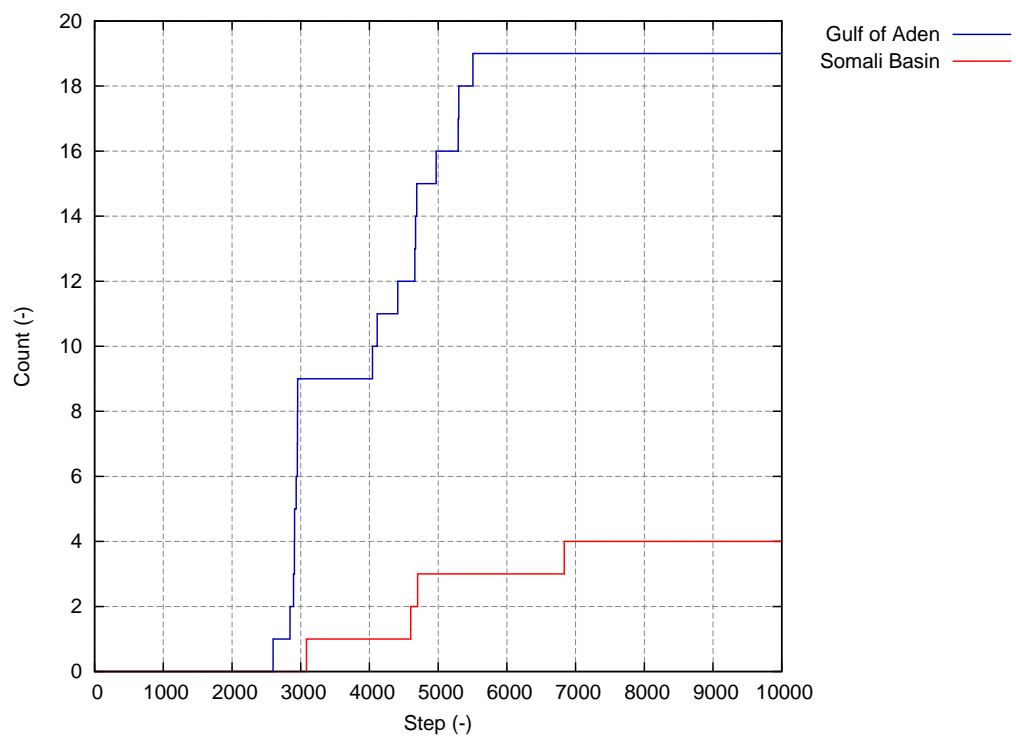
Obrázek 4.5: Počet útoků v čase v závislosti na vzdálenosti horizontu (No Info strategie)



Obrázek 4.6: Útoky během dne (No Info strategie, 50 lodí)



Obrázek 4.7: Celkový počet útoků v čase v závislosti na oblasti (No Info strategie)



Obrázek 4.8: Celkový počet útoků v čase v závislosti na oblasti (Bandit strategie)

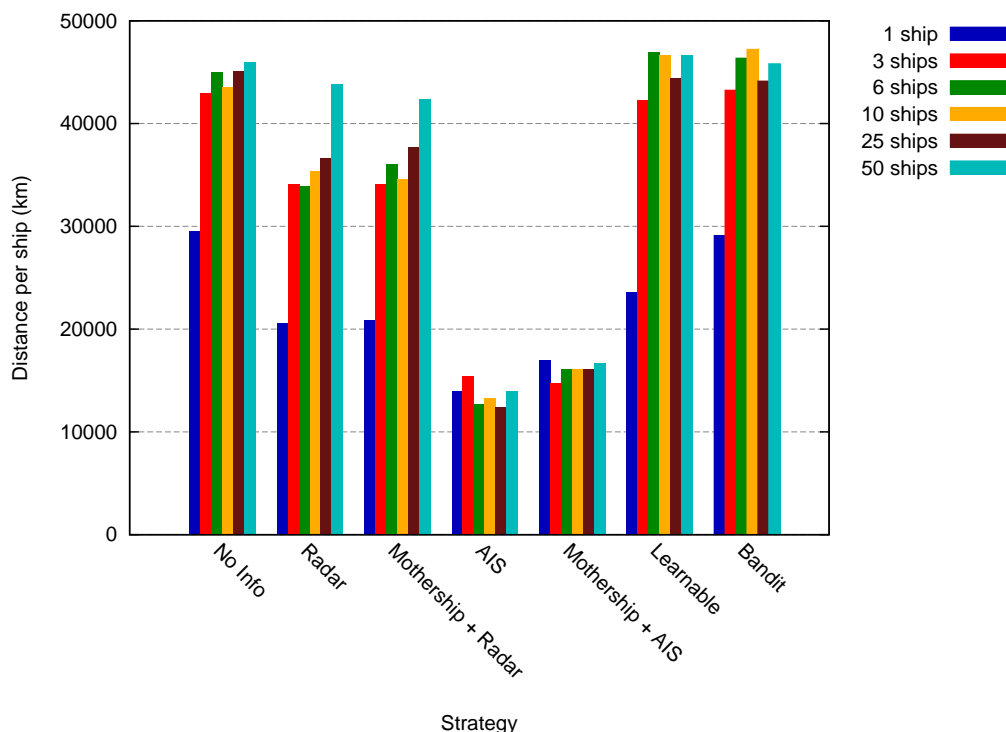
Uplutá vzdálenost

Další skupinou dat, která byla získána simulací jednotlivých strategií je uplutá vzdálenost a s tím spojená spotřeba paliva a tedy náklady na plutí pirátů po moři.

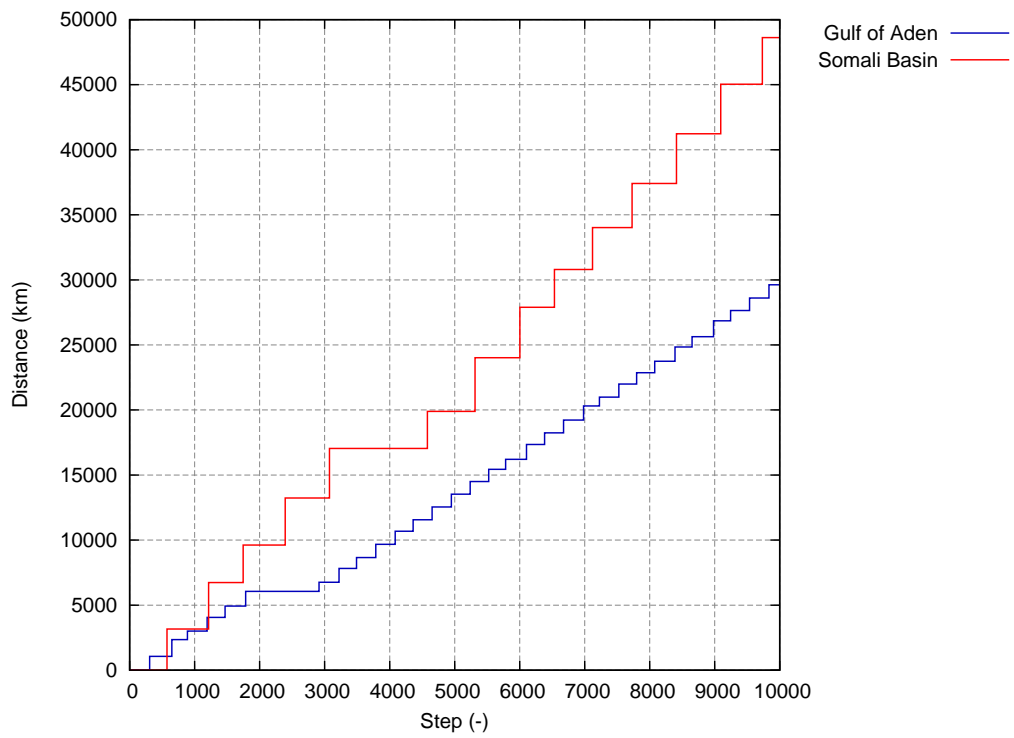
Na obr. 4.9 lze vidět graf průměrné upluté vzdálenosti jedné pirátské lodi v závislosti na počtu pirátských lodí při simulaci. Je zde vidět při porovnání s obr. 4.1 (Průměrný počet útoků při stejné závislosti), že uplutá vzdálenost souvisí s počtem útoků. U strategie AIS a Mothership + AIS je větší nepoměr oproti jiným strategiím, což je způsobeno principem této strategie. Opět je zde vidět, že jedna pirátská loď plula v Adenském zálivu a proto nemusela urazit tak velkou vzdálenost. Rozdíl uplutých vzdáleností v čase a v závislosti na místě, kde pirát operuje je vidět na obr. 4.10.

Graf průměrné upluté vzdálenosti na rychlosti transportních lodí je možné vidět na obr. 4.11. Zde je vidět, že rychlost transportních lodí nemá prakticky žádný vliv na uploutou vzdálenost.

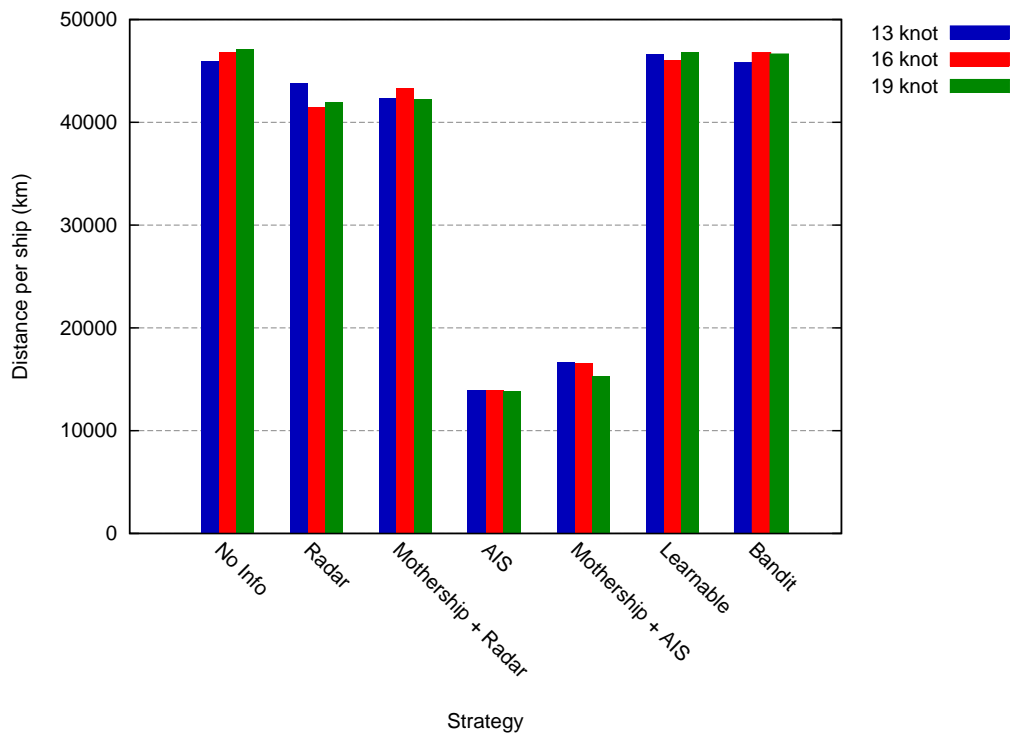
Na obr. 4.12 lze vidět graf průměrné upluté vzdálenosti pirátské lodi v závislosti na viditelné vzdálenosti (výšce lodi). Z tohoto grafu je patrné, že zvětšení vzdálenosti, kam pirát vidí, má pozitivní vliv na uploutou vzdálenost. Toto opět neplatí u strategie AIS a AIS + Mothership z důvodů, kde se tato veličina prakticky nepoužívá.



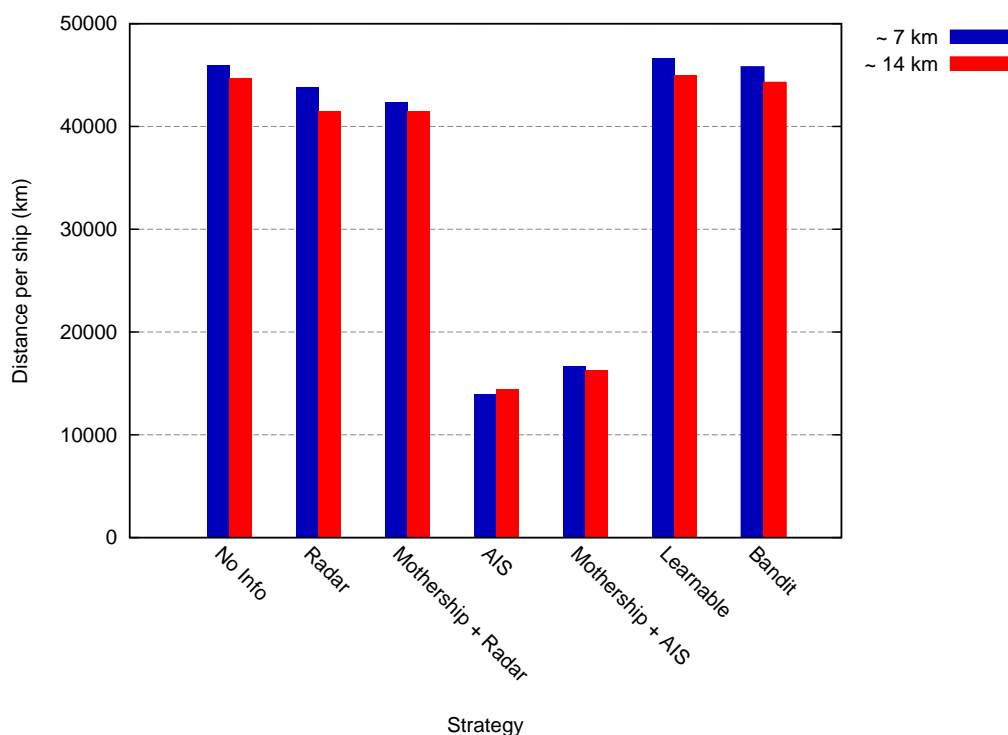
Obrázek 4.9: Uplutá vzdálenost v závislosti na počtu pirátských lodí



Obrázek 4.10: Uplutá vzdálenost v čase v závislosti na oblasti (No Info strategie)



Obrázek 4.11: Uplutá vzdálenost v závislosti na rychlosti transportních lodí



Obrázek 4.12: Uplutá vzdálenost v závislosti na vzdálenosti horizontu

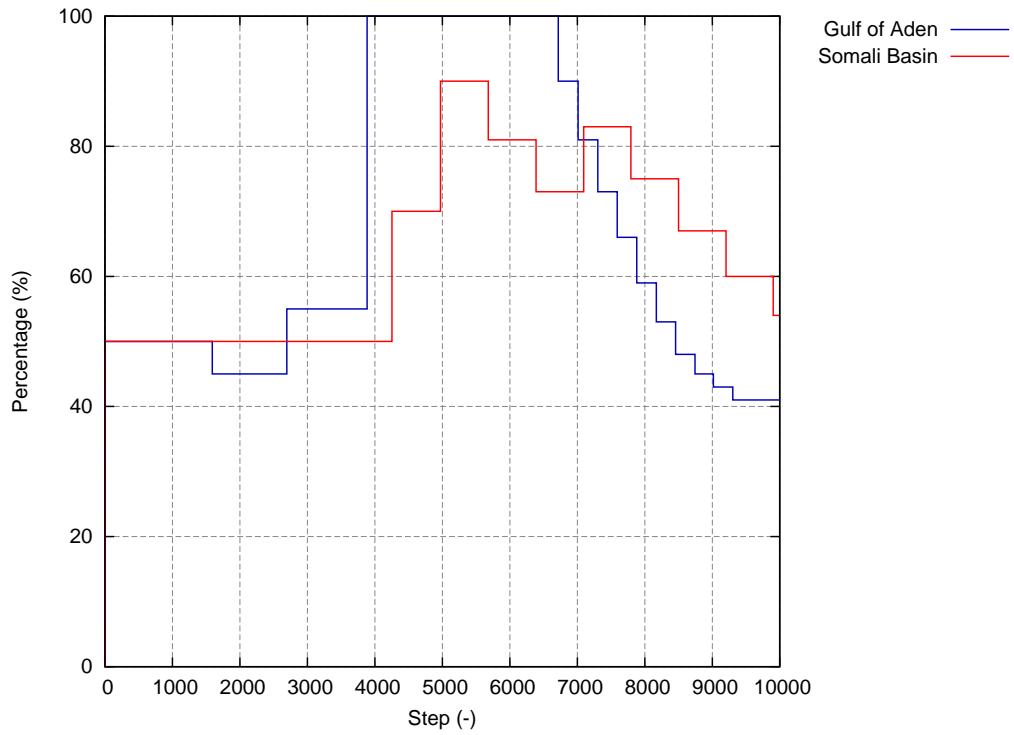
Procentní hodnoty

Poslední skupinou dat, která byla získána, jsou procentní hodnoty vnitřních proměnných agentů, které vyjadřují maximální hodnotu pravděpodobnosti výskytu transportní lodi nad všemi podoblastmi u strategie Bandit a nebo hodnotu pravděpodobnosti zužitkování předchozích znalostí u strategie Learnable.

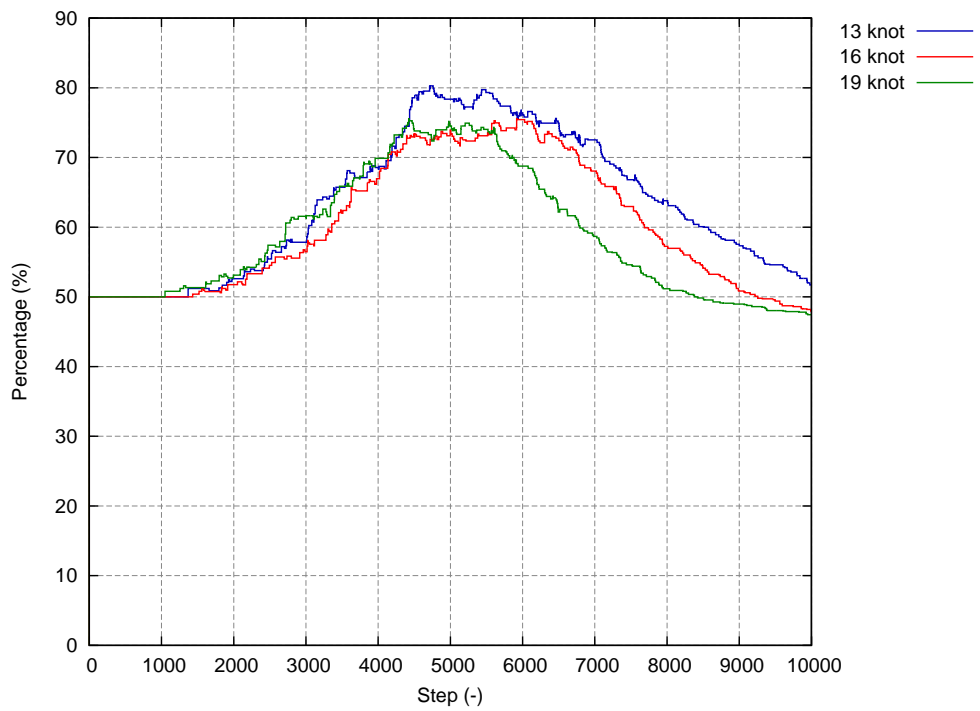
Na obr. 4.13 lze vidět rozdíl v maximální hodnotě pravděpodobnosti výskytu transportní lodi ve strategii Bandit a v závislosti na oblasti, kde se loď pohybuje. Zde je vidět strmý nárůst maximální hodnoty, a pak dlouhé zužitkování této znalosti vhodné podoblasti k útoku.

V grafech na obr. 4.14 a 4.15 je vidět vývoj procentních hodnot u strategií Bandit a Learnable v závislosti na rychlosti transportních lodí. Z těchto grafů je vidět, že při zvětšení rychlosti transportních lodí musí pirát více hledat, aby stihl loď napadnout.

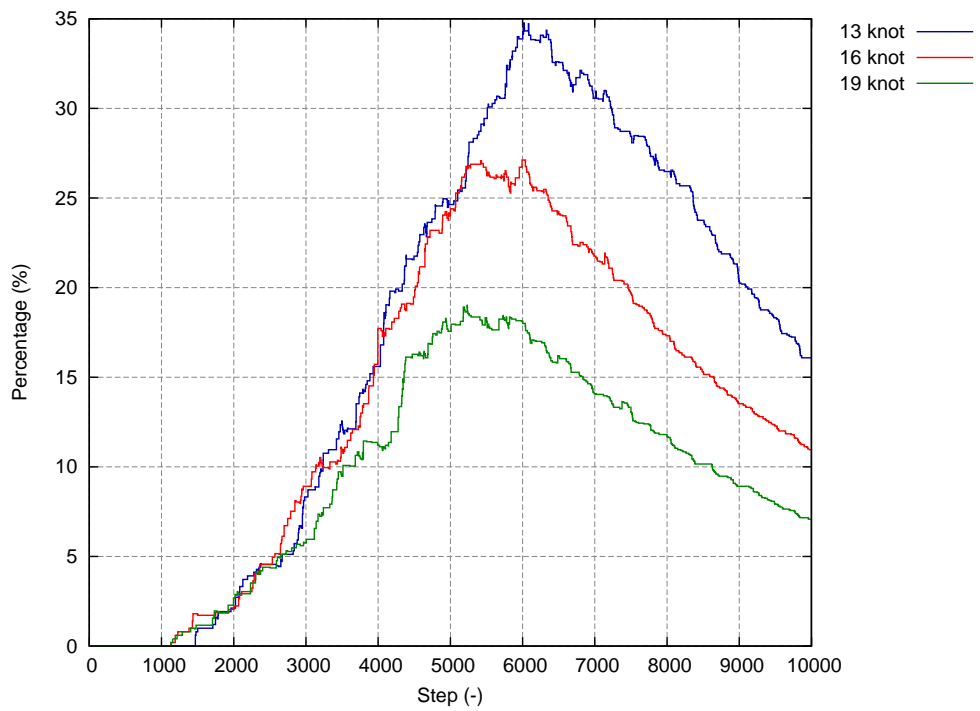
Podobný graf jako v předchozím odstavci je na obr. 4.16 a 4.17, avšak je v závislosti na vzdálenosti horizontu, a tedy i výšce pirátské lodi. Zde je patrný pozitivní nárůst zvětšení viditelné vzdálenosti na procentních hodnotách strategií Bandit a Learnable.



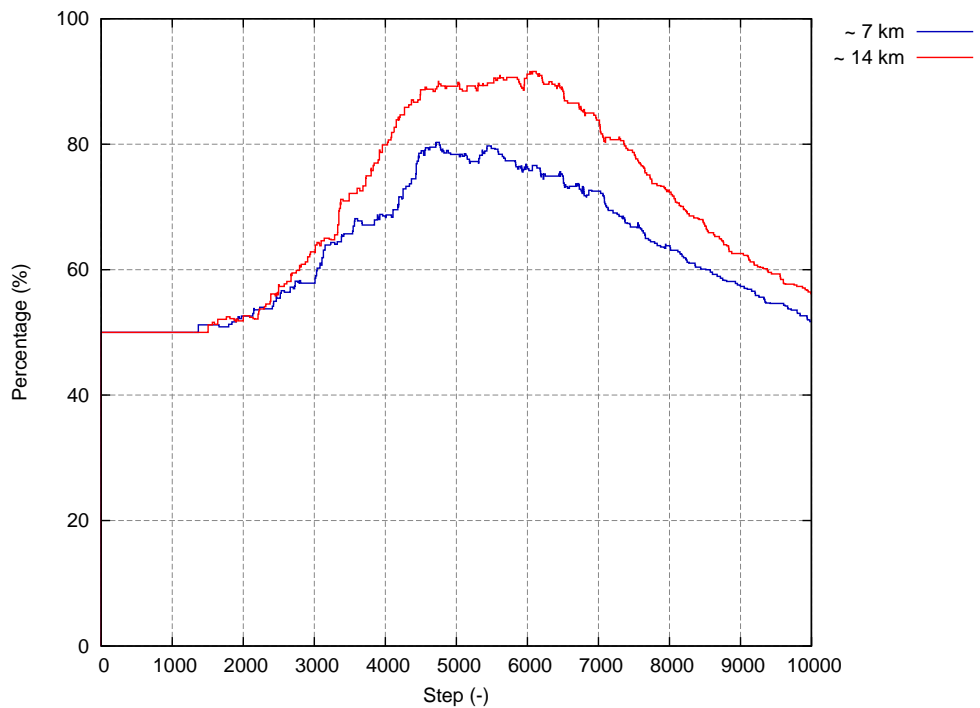
Obrázek 4.13: Max. procentní hodnota v čase v závislosti na oblasti (Bandit strategie)



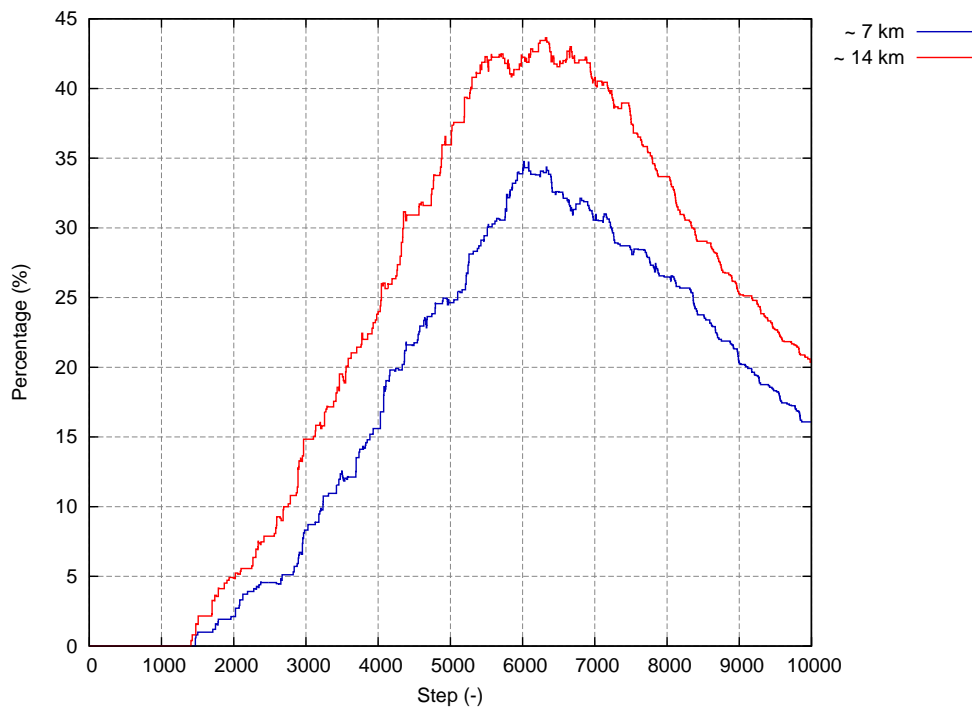
Obrázek 4.14: Aktuální max. procentní hodnota v závislosti na rychlosti transp. lodí (Bandit strategie)



Obrázek 4.15: Aktuální procentní hodnota v závislosti na rychlosti transportních lodí (Learnable strategie)



Obrázek 4.16: Aktuální max. procentní hodnota v závislosti na vzdálenosti horizontu (Bandit strategie)



Obrázek 4.17: Aktuální procentní hodnota v závislosti na vzdálenosti horizontu (Learnable strategie)

4.3 Diskuze

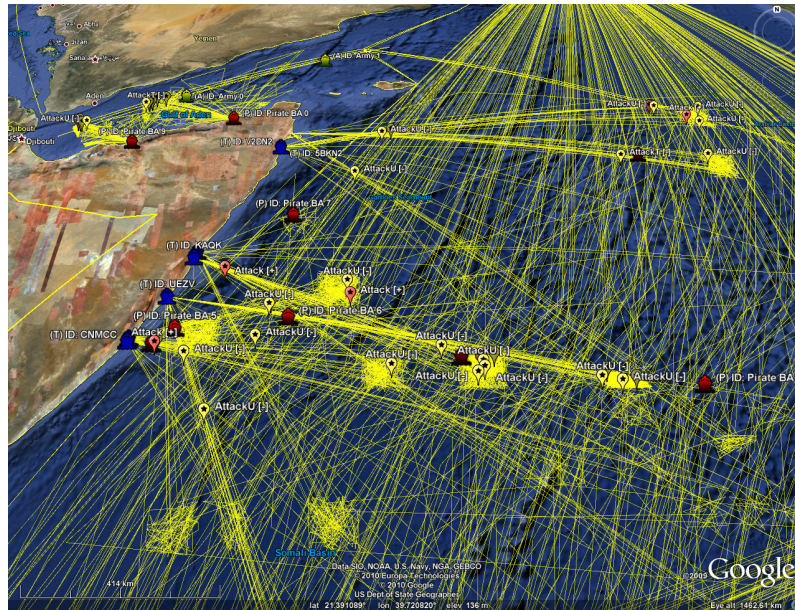
V této sekci shrnu a okomentuji získané výsledky ze simulací a zaměřím se na jejich přesnost vzhledem k reálné situaci.

Získané hodnoty pomocí simulace různých strategií většinou odpovídají reálné situaci v okolí pobřeží Somálska. Hodnoty, které situaci moc neodpovídají, jsou většinou způsobeny nastavením modelu, kde je obtížné simulovat větší časové intervaly než přibližně měsíc z důvodu výpočetní náročnosti. Toto se projevuje zejména u hodnot útoku a upluté vzdálenosti, které jsou o trochu větší než v reálné situaci a to z důvodu, že piráti čekají na výkupné po uloupení lodi jen 3 dny. Další výraznější odchylení od reality je také u upluté vzdálenosti lodí, která je na hranici možností, což je způsobeno tím, že většina lodí po západu slunce odpluje do přístavu a hned po půlnoci zase pluje bez delšího čekání na další cestu.

Při pohledu na grafy je možné zjistit, že strategie Learnable je lepší než No Info, což je způsobeno učením piráta. U strategie Bandit jsou někdy výsledky horší než u No Info, což může být způsobeno relativně kratší dobou simulace, kdy pirát nestačil získat dostatek znalostí. U strategií s radarem (Radar a Mothership + Radar) je vidět velké zlepšení oproti strategii No Info. V případě použití mateřské lodi se tento rozdíl ještě zvětší. Velmi zajímavé výsledky má také strategie AIS a Mothership + AIS,

kdy pirát má také velký počet útoků, ale v porovnání s ostatními strategiemi má nejvyšší účinnost v poměru počet útoků na uploutou vzdálenost. V této strategii totiž pirát upluje jen polovinu toho, co v jiných strategiích.

Procentní hodnoty u učících se strategií jsou také podle očekávání a vliv na ně je úměrný jako na počet útoků ve strategii No Info. Za zmínku také stojí očekávatelný rozdíl mezi loděmi v Adenském zálivu a Somálskou pánví a to jak v počtu útoků, tak v upluté vzdálenosti.



Obrázek 4.18: Vizualizace simulace strategie Bandit pro 10 lodí v Google Earth

Kapitola 5

Závěr

V práci byla získána data a vypracován přehled o současných pirátských strategiích v oblasti Adenského zálivu a Somálské pánve. Dále byly navrženy vhodné pirátské strategie. Ty byly namodelovány pomocí konečných stavových automatů za použití jazyka UML. Implementace strategií probíhala v rámci multi-agentní platformy AgentC. Naimplementované strategie byly otestovány a vyhodnoceny z hlediska úspěšnosti a blízkosti vůči reálné situaci v oblasti Somálska.

Pro přesnější implementaci a možnost zhodnocení naimplementovaných strategií by bylo potřeba přesnějších a důležitějších dat, které v současné době nejsou veřejně dostupné. Dnes je možné obvykle získat neúplné informace, které je nutné skládat dohromady, avšak jejich syntéza obtížně nahradí data, která jsou získaná například z GPS zařízení či satelitů.

Jelikož je problematika této práce velice zajímavá, tak bych se chtěl tímto tématem nadále zabývat a zhodnotit získané znalosti v dalších pracích. Také bych chtěl, bude-li to možné, dále rozvíjet naimplementované strategie v závislosti na nově získaných informacích a vývoji ve sledované oblasti.

Literatura

- [1] Adamczyk, P.: *The Anthology of the Finite State Machine Design Patterns* [online]. 2003, [cit. 10. 4. 2010].
URL <http://hillside.net/plop/plop2003/Papers/Adamczyk-State-Machine.pdf>
- [2] Bayer J., Hanzálek Z., Šusta R.: *Logické řízení*. ČVUT, druhé vydání, srpen 2008, ISBN: 978-80-01-04106-2.
- [3] BBC: *Saudi tanker 'freed off Somalia'* [online]. 2009, [cit. 10. 4. 2010].
URL <http://news.bbc.co.uk/2/hi/africa/7820311.stm>
- [4] Bell, D.: *UML basics: The class diagram* [online]. 2004, [cit. 14. 5. 2010].
URL <http://www.ibm.com/developerworks/rational/library/content/RationalEdge/sep04/bell/>
- [5] Gilpin, R.: *Counting the Costs of Somali Piracy* [online]. United States Institute of Piece, 2009, [cit. 10. 4. 2010].
URL http://www.usip.org/files/resources/1_0.pdf
- [6] Global Executive Outreach: *Maritime Security Patrol Area* [online]. 2009, [cit. 10. 4. 2010].
URL <http://www.geo-ops.com/securitynews/44-maritimesecuritypatrolarea.html>
- [7] Jakob M., Vaněk O., Urban S., Benda P., Pěchouček M.: *Adversarial Modeling and Reasoning in the Maritime Domain (Year 1 Report)*. Technická zpráva, Agent Technology Center, Department of Cybernetics, FEE, CTU in Prague, 2009.
URL <http://agents.felk.cvut.cz/cgi-bin/docarc/public.pl/document/278/main.pdf>
- [8] Maritime International Secretariat Services Limited: *Best Management Practices to Deter Piracy in the Gulf of Aden and off the Coast of Somalia* [online]. 2009, [cit. 10. 4. 2010].
URL <http://www.marisec.org/piracy-gulf-of-aden-indian-ocean-industry-best-management-practice>
- [9] North of England P&I Association: *Piracy – Gulf of Aden* [online]. 2010, [cit. 10. 4. 2010].

- URL <http://www.nepia.com/cache/files/1576-1266327081/LPBriefing-Piracy-GulfofAden.pdf>
- [10] Open Geospatial Consortium: *KML* [online]. [cit. 10. 4. 2010].
URL <http://www.opengeospatial.org/standards/kml/>
- [11] Papaioannou C. A.: *Operation ATALANTA* [online], The "Athena '09" Crisis Management International Conference, 1. den, 7. přednáška. 2009, [cit. 16. 4. 2010].
URL http://www.geetha.mil.gr/index.asp?a_id=2956
- [12] Russell, S.; Norvig, P.: *Artificial Intelligence: A Modern Approach*. Prentice Hall, třetí vydání, 2009, 1132 s., ISBN: 978-0136042594.
- [13] Tsvetkova, B.: *Securitizing Piracy Off the Coast of Somalia* [online]. 2009, [cit. 4. 10. 2010].
URL http://www.cejiss.org/assets/pdf/articles/vol3-1/Tsvetkova-Piracy_in_Somalia.pdf
- [14] UNOSAT: *Spatial Analysis of Somali Pirate Activity in 2009* [online]. 2009, [cit. 10. 4. 2010].
URL http://unosat-maps.web.cern.ch/unosat-maps/S0/Piracy/2009/UNOSAT_Somalia_Pirates_Analysis_Q1_2009_23April09_v1.pdf
- [15] U.S. Maritime Administration: *Somali Pirate Tactics (Pamphlet)* [online]. 2008, [cit. 11. 4. 2010].
URL <http://www.skuld.com/upload/News%20and%20Publications/Publications/Piracy/Somali%20Pirate%20Tactics%20Pamphlet%202008.pdf>
- [16] Vermorel, J.; Mohri, M.: Multi-Armed Bandit Algorithms and Empirical Evaluation [online]. *Lecture Notes in Computer Science*, ročník 3720, 2005: s. 437–448, [cit. 18. 4. 2010].
URL <http://www.cs.nyu.edu/~mohri/pub/bandit.pdf>

Obsah CD

Příložené CD obsahuje zdrojový kód bakalářské práce, spustitelný kód včetně projektu AgentC, vygenerované grafy ze simulací ve formátu PDF a samotný text bakalářské práce také ve formátu PDF. V následující tabulce je popsána struktura CD.

Cesta	Popis
<code>src/</code>	zdrojový kód bakalářské práce
<code>bin/</code>	spustitelný kód bakalářské práce vč. projektu AgentC
<code>charts/</code>	vygenerované grafy, které se nevešly do sekce 4.2.1
<code>thesis.pdf</code>	text diplomové práce

Tabulka 1: Adresářová struktura na CD