

VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA INFORMAČNÍCH TECHNOLOGIÍ
ÚSTAV INTELIGENTNÍCH SYSTÉMŮ

FACULTY OF INFORMATION TECHNOLOGY
DEPARTMENT OF INTELLIGENT SYSTEMS

MODELOVÁNÍ SOCIÁLNÍ SÍŤE POMOCÍ MULTI-AGENTNÍCH SYSTÉMŮ

BAKALÁŘSKÁ PRÁCE
BACHELOR'S THESIS

AUTOR PRÁCE
AUTHOR

GÁBOR LELKES

BRNO 2013



VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
BRNO UNIVERSITY OF TECHNOLOGY



FAKULTA INFORMAČNÍCH TECHNOLOGIÍ
ÚSTAV INTELIGENTNÍCH SYSTÉMŮ

FACULTY OF INFORMATION TECHNOLOGY
DEPARTMENT OF INTELLIGENT SYSTEMS

MODELOVÁNÍ SOCIÁLNÍ SÍTĚ POMOCÍ MULTI-AGENTNÍCH SYSTÉMŮ

MULTI-AGENT SYSTEMS FOR SOCIAL NETWORK MODELLING

BAKALÁŘSKÁ PRÁCE

BACHELOR'S THESIS

AUTOR PRÁCE

AUTHOR

GÁBOR LELKES

VEDOUCÍ PRÁCE

SUPERVISOR

Ing. JAN SAMEK, Ph.D.

BRNO 2013

Abstrakt

Práce poskytuje čitateli úvod do témat sociálních sítí a multi-agentních systémů. Jejím cílem je popsat návrh a implementaci funkčního modelu sociální sítě jakožto multi-agentního systému postaveného na frameworku Jason a na závěr zhodnotit tuto snahu.

Abstract

This thesis introduces the reader to topics of social networks and multi-agent systems. It's goal is to describe design and implementation of a functional model of social network as a multi-agent system built on Jason framework, and, in the end evaluate this effort.

Klíčová slova

Agent, model, sociální síť, Jason, Java, AgentSpeak.

Keywords

Agent, model, social network, Jason, Java, AgentSpeak.

Citace

Gábor Lelkes: Modelování sociální sítě pomocí multi-agentních systémů, bakalářská práce, Brno, FIT VUT v Brně, 2013

Modelování sociální sítě pomocí multi-agentních systémů

Prohlášení

Prohlašuji, že jsem tuto bakalářskou práci vypracoval samostatně pod vedením pana Ing. Jana Samka Ph.D.

.....
Gábor Lelkes
14. května 2013

Poděkování

Rád by som sa poďakoval vedúcemu mojej práce, pánovi Ing. Janu Samkovi, Ph.D. za jeho trpezlivosť a ochotu, ktorú pri konzultáciách prejavoval.

© Gábor Lelkes, 2013.

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

1 Úvod	3
1.1 Štruktúra práce	3
2 Multi-agentné systémy	4
2.1 Agent	4
2.2 Agentný systém	4
2.3 Vlastnosti agentov	5
2.3.1 Autonómia	5
2.3.2 Reaktivita	5
2.3.3 Proaktivita	5
2.3.4 Schopnosť učiť sa	5
2.4 Racionálny agent	5
2.5 BDI model	6
2.6 Implementácia multi-agentných systémov	6
2.6.1 Architektúra IRMA	7
2.6.2 Architektúra FIPA	7
2.6.3 JADE	7
2.6.4 Jazyk AgentSpeak	7
2.6.5 Jason	7
3 Sociálne siete	8
3.1 Pojmy v sociálnych sieťach	8
3.1.1 Aktér	8
3.1.2 Binárny vzťah	8
3.1.3 Klika	8
3.1.4 Komunita	9
3.2 Tvorba vzťahov	9
3.3 Webové sociálne siete	9
3.4 Príklady sociálnych sietí	10
3.4.1 Trieda na základnej škole	10
3.4.2 Základná škola	10
3.4.3 Základné školy v jednom meste	10
3.4.4 Facebook	10
3.4.5 Google+	11
3.4.6 Twitter	11
3.4.7 LinkedIn	11
3.5 Model sociálnej siete	11
3.6 Analýza sociálnej siete	12

4	Návrh modelu sociálnej siete	13
4.1	Fyzický svet	13
4.1.1	Mentálny stav agenta	14
4.2	Spoločenský svet	15
4.2.1	Nadväzovanie kontaktu	15
4.2.2	Komunikácia	16
4.3	Predpokladaný výstup	17
5	Implementácia	18
5.1	Java	18
5.1.1	View	19
5.1.2	Model	19
5.1.3	Controller	19
5.1.4	Telo agenta	20
5.1.5	Rozhodovacia logika agenta	20
5.1.6	Interné akcie	20
5.2	AgentSpeak	21
5.2.1	Stretnutie	21
5.2.2	Rozhovor	21
6	Analýza výsledného modelu	23
6.1	Vzorová sieť z Facebooku	23
6.2	Výstup modelu - všetky známosti	24
6.3	Výstup modelu - priatelia	24
6.4	Štatistické porovnanie	25
7	Záver	27
A	Obsah CD	30

Kapitola 1

Úvod

Pre ľudí a zvieratá s určitou mierou inteligencie je typické a prirodzené vytvárať medzi sebou sociálne väzby a takto sa zoskupovať do sociálnych skupín. V priebehu 20. storočia sa veda postupne začala zaoberať analýzou takýchto skupín a vzniklo označenie sociálna sieť. Predovšetkým v oblasti informatiky a informačných technológií sa zas začal rozvíjať relatívne nový smer zaoberajúci sa agentnými a multi-agentnými systémami, teda takými systémami, v ktorých sa pohybuje jeden a viac samostatných prvkov. Na prvý pohľad nie príliš spolu súvisiace témy sa v skutočnosti skvele dopĺňajú, keďže sociálna sieť je multi-agentným systémom, a naopak, dá sa pomocou multi-agentných systémov jednoducho modelovať, keď zoberieme do úvahy že agentom sa stáva človek.

Táto bakalárska práca sa zaoberá práve týmto spojením, teda modelovaním sociálnej siete pomocou multi-agentného systému, kde výsledkom je model použiteľný na experimentovanie bez potreby skutočnej sociálnej siete, alebo na nadväzujúci výskum v oblasti skúmania dôvery v počítačových systémoch.

1.1 Štruktúra práce

V prvých dvoch kapitolách sa zoznámime s pojmami zásadnými pre túto prácu - multi-agentnými systémami a sociálnymi sieťami. V tretej kapitole nasleduje návrh modelu, ktorý čerpá z oboch tém a v kapitole štvrtej je popísaná implementácia programu, ktorý tento model realizuje. Na záver je prichystaná analýza a validácia tohoto modelu.

Kapitola 2

Multi-agentné systémy

Táto kapitola si kladie za úlohu uviesť čitateľa do oblasti multi-agentných systémov, čo si vyžaduje vysvetliť niektoré základné pojmy, v prvom rade, čo to vlastne ten agent je.

2.1 Agent

Agentom rozumieme aktívny autonómny prvok vyskytujúci sa v určitom systéme, s ktorým sa vzájomne môžu ovplyvňovať. Definície z rôznych zdrojov sa mierne líšia, avšak väčšina sa zhoduje na tom, že agent je schopný (po tom čo bol vytvorený) určitého stupňa autonómie, teda samostatného konania. Agent vníma svoje okolie, a je schopný ho ovplyvňovať.

Tento pojem a teória okolo neho (obzvlášť multi-agentné systémy) sú využiteľné v širokom spektre vedných odborov, počnúc informatikou, cez biológiu až po sociológiu, kde agentom môžeme označiť program, robota, mravca v mravenisku alebo aj človeka. Čo sa týka tejto práce, ďalej sa bude zaoberať len agentami z pohľadu informatiky, konkrétne softwarovými agentami.

Úlohou agentov a ich praktickým využitím v reálnom živote je zastupovať človeka pri každodenných činnostiach, uľahčiť mu život napríklad v podobe programu pre vyhľadanie najkratšej cesty do cieľa, plánovanie schôdzok či automatické draženie v aukcii, ale aj pri činnostiach, ktoré sú pre človeka príliš náročné a vyžadujú strojovú presnosť, ako príklad sa dá uviesť software lietadla dohadujúci letovú dráhu s iným lietadlom.

Zmyslom života agenta je dosiahnuť nejaký dopredu stanovený cieľ, ktorý väčšinou predstavuje uvedenie svojho okolia (systému) do určitého stavu. Na to aby tento cieľ dosiahol musí ovplyvňovať a meniť systém, v ktorom sa nachádza, až kým nie je „spokojný“. K tomu využíva rôzne svoje schopnosti ako plánovanie postupu, sledovanie okolia a následné spracovanie vstupov, komunikácia s inými agentami a podobne. V časti 2.3 sú stručne zhrnuté jednotlivé vlastnosti, ktorými môže agent oplývať. [18, 19]

2.2 Agentný systém

Agentný systém je taký, v ktorom sa vyskytuje agent. Tento agent, ovplyvňuje systém a je systémom ovplyvňovaný. My sa budeme v tejto práci zaoberať multi-agentnými systémami, čo sú také, v ktorých sa agentov vyskytuje viac. Systémy môžeme deliť podľa ich vzťahu k okoliu na otvorené, relatívne uzatvorené a uzatvorené. Otvorené sú schopné komunikovať s okolím, relatívne uzatvorené len s jeho podmnožinou. Uzatvorené systémy s okolím

komunikovať nedokážu, interakcia prebieha len v rámci systému medzi jednotlivými prvkami. Ďalej môžeme agentné systémy rozdeliť podľa chovania prvkov na diskkrétne, spojité, deterministické a stochastické. [19]

2.3 Vlastnosti agentov

Nielen softwaroví agenti, ale aj agenti iných druhov, sa vo všeobecnosti dajú deliť podľa toho akými vlastnosťami disponujú. Nás predovšetkým zaujíma to, ako komunikujú so svojim okolím, a aké vnútorné procesy v nich prebiehajú. Tieto vlastnosti sa dajú navzájom kombinovať a tým vzniká určitá hierarchia medzi týmito typmi. [18, 19]

2.3.1 Autonómia

Základnou vlastnosťou všetkých agentov je schopnosť samostatného konania. Tá podmieňuje definíciu agenta a určuje smer jeho využitia. Autonómia umožňuje agentovi zastupovať človeka a pracovať bez nutnosti jeho dohľadu.

2.3.2 Reaktivita

Hneď po autonómii je to najdôležitejšia vlastnosť agentov, schopnosť reagovať na zmeny a podnety z okolia. Keď reaktívny agent zaregistruje takýto podnet, vyhodnotí ho, a primerane situácii mu odpovedá. Týmto sa už dá simulovať určitá forma inteligentného správania v prípade, že je množina prednastavených odpovedí adekvátne množine možných podnetov, pričom stále je zachovaná potreba agenta ovplyvňovať okolie tak, aby dosiahol svoj cieľ.

2.3.3 Proaktivita

Jedná sa o schopnosť agenta plánovať postup svojich akcií už dopredu tak, aby dosiahol svoj cieľ. Na rozdiel od agenta reaktívneho, proaktívny agent nečaká pasívne na podnety z okolia, ale ak má možnosť z vlastnej iniciatívy ho ovplyvňuje. Takýto agent sa dá označiť aj pojmom deliberatívny.

2.3.4 Schopnosť učiť sa

Dôležitou vlastnosťou agentov je schopnosť zbierať informácie zo svojho prostredia a vytvárať si z nich bázu znalostí. Na základe týchto poznatkov si potom agent môže upraviť svoj aktuálny plán na dosiahnutie cieľa, prípadne zmeniť samotný cieľ. Takýto agent, ktorého môžeme nazvať kognitívnym, nemusí byť nutne proaktívny, môže len zbierať informácie z prostredia a analyzovať ich.

2.4 Racionálny agent

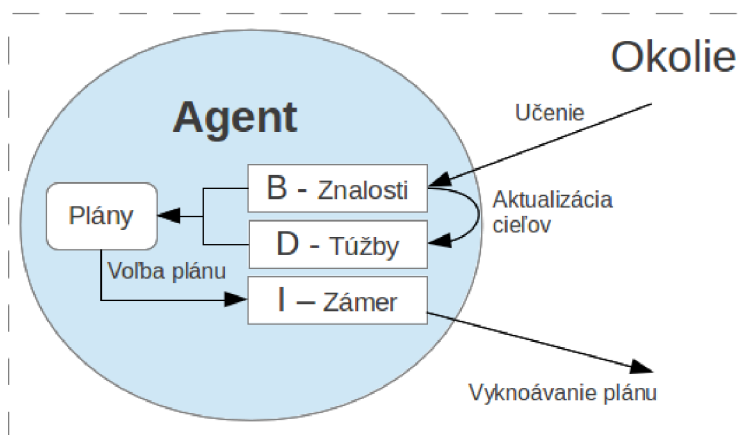
Keď agent nadobudne všetky uvedené vlastnosti, stáva sa agentom racionálnym. Takýto agent je schopný vnímať svoje okolie, učiť sa, teda vytvárať si bázu znalostí, a na základe týchto znalostí potom plánovať nasledujúci postup. V prípade že sa zmení prostredie, alebo dostane podnet z prostredia, adekvátne reaguje. Tieto činnosti vykonáva úplne samostatne za účelom dosiahnutia svojho cieľa.

2.5 BDI model

BDI (Belief-Desire-Intention, v preklade presvedčenie-túžba-zámer) model je určený pre návrh inteligentných agentov. Bol vytvorený Michaelom Bratmanom a implementuje jeho teóriu praktického uvažovania (practical reasoning).

Presvedčenie agenta je reprezentované bázou znalostí, ktorou agent disponuje (hovoríme o presvedčení pretože takáto znalosť nemusí byť faktom v prípade že má agent obmedzený výhľad). Túžby sú množina stavov systému, ktoré by chcel agent v konečnom dôsledku dosiahnuť, kdežto zámer je jeden z týchto stavov (prípadne viac), ku ktorému sa agent práve rozhodol približovať vykonávaním vhodne zvoleného plánu z báze plánov, čím vzniká určitá forma záväzku voči vykonaniu tohoto plánu.

Základom BDI modelu je BDI agent, racionálny agent zameraný na splnenie svojho cieľa, ktorého rozhodovanie je riadené určitou logikou a zároveň môže byť obmedzené zdrojmi, ktoré má agent k dispozícii (rozhľad, čas apod.). Funkciou modelu je potom na základe dostupných znalostí zvoliť pre agenta taký plán k dosiahnutiu cieľa, ktorý bude pre neho najvýhodnejší. Agent si pri rozhodovaní musí zároveň rozdeliť čas na rozhodovanie a vykonávanie plánu (tzn. je tu riešená aj dimenzia času). Samotné vykonávanie BDI model už nerieši, to je ponechané na návrhárovi multiagentného systému.



Obrázek 2.1: Cyklus rozhodovania BDI agenta.

Je vhodné zaviesť aj pojem agentná architektúra - v oblasti softwaru je to framework, v ktorom agentný systém beží. V tomto prípade je teda BDI architektúra framework, ktorý umožňuje vytvorenie a spustenie multi-agentného systému založeného na BDI agentoch. [5, 10]

2.6 Implementácia multi-agentných systémov

Pre vytvorenie softwarového multi-agentného systému sa bežne používajú programovacie jazyky. Autor systému, programátor, vytvorí pomocou konštrukcií daného programovacieho jazyka požadované typy agentov, teda určí ich reakcie na podnety, ich ciele, a spôsoby akými ich dosiahnu. Tieto jazyky umožňujú aj navrhnutie prostredia, v ktorom budú agenti pôsobiť, a jeho integráciu do iných softwarových produktov. Existuje viacero rôznych jazykov, architektúr a frameworkov pre vytváranie multiagentných systémov, táto práca sa však zaoberá len platformou Jason založenej na jazyku AgentSpeak.

2.6.1 Architektúra IRMA

Intelligent Resource-bounded Machine Architecture navrhnutá skupinou Bratman, Israel, Pollack, je prvou architektúrou postavenou na BDI modele. Kladie dôraz na fakt, že agent sledujúci svoje dlhodobé ciele je viazaný prostriedkami ktoré má k dispozícii a tým je ovplyvňované aj jeho rozhodovanie. Agent je obmedzený len na svoje znalosti a potrebuje čas na vytvorenie plánu ďalšieho postupu, pričom sa môže stať že kým tento plán vytvorí, podmienky v prostredí sa zmenia a tento plán nemusí byť už vôbec použiteľný. [15, 19]

2.6.2 Architektúra FIPA

V roku 1996 vznikla v Švajčiarsku nezisková organizácia FIPA (Foundation for Intelligent Physical Agents), ktorej cieľom bolo vytváranie štandardov v oblasti agentov a multi-agentných systémov. Spoločne tieto štandardy vytvárajú abstraktnú architektúru FIPA, ktorá obsahuje princípy spoločné pre väčšinu vytvorených agentných systémov. [7, 19]

2.6.3 JADE

JADE (Java Agent DEvelopment Framework) je softwarový framework pre vývoj multi-agentných systémov. Je založený na architektúre FIPA a implementovaný v jazyku Java. Špecifikom tohoto frameworku je že jednotliví agenti v rámci systému sa nemusia nachádzať na jednom fyzickom stroji, ale môžu byť rozložení na viacerých strojoch, ktoré nemusia ani zdieľať rovnaký operačný systém. Konfiguráciu takéhoto systému je navyše možné meniť aj počas behu. [1]

2.6.4 Jazyk AgentSpeak

Logický programovací jazyk, ktorý vytvoril v roku 1996 Anand Rao. Pôvodne sa jednalo len o abstraktný programovací jazyk označený AgentSpeak(L). Je určený pre logické programovanie agentných systémov modelu BDI. [10, 3]

2.6.5 Jason

Jason je platforma pre vývoj multi-agentných systémov. Jazyk, ktorý Jason interpretuje je rozšírením jazyka AgentSpeak(L). Je implementovaný v jazyku Java, čo mu zaručuje multiplatformovosť, a zároveň sa jedná o open-source produkt, distribuovaný pod licenciou GNU LGPL. Poskytuje užívateľské rozhranie a mnohé nástroje pre vývoj multi-agentných systémov, umožňuje vkladať programy vytvorené v AgentSpeaku do prostredí vytvorených v Jave, čím uľahčuje prepojenie týchto modelov s okolím. Sociálna sieť navrhnutá v tejto práci bude implementovaná práve za využitia platformy Jason. [10]

Kapitola 3

Sociálne siete

Prvý krát pojem sociálna sieť použil vo svojom článku o pobyte na nórskom vidieku J.A. Barnes [20], ktorý tam pozoroval spoločenské javy na miestnom obyvateľstve. Sociálna sieť je definovaná takto:

Sociálna sieť je spoločenská štruktúra vytvorená z množiny aktérov (jednotlivci či skupiny) a množiny vzťahov medzi týmito účastníkmi, ako sú priateľstvo, príbuzenstvo, spoločný záujem, nenávisť a podobne. [4]

Toto je obecná definícia sociálnej siete avšak v rôznych publikáciách sa môžu definície líšiť. [11]

3.1 Pojmy v sociálnych sieťach

Nasleduje zoznam základných rozpoznateľných elementov v sociálnych sieťach. [14, 17]

3.1.1 Aktér

Aktér je základná jednotka sociálnej siete, vytvára väzby s inými aktérmi. Môže sa jednať o jednotlivca alebo o skupinu, pričom v rámci jednej siete sa môžu nachádzať aktéri rôznych typov. Aktér so sebou môže niesť svoje vlastnosti - jeho atribučné dáta. Príkladom aktéra je človek v spoločnosti alebo firma v obchodnej skupine.

3.1.2 Binárny vzťah

Väzba medzi práve dvoma aktérmi. Väzby obecné - relačné dáta - patria všetkým účastníkom danej väzby. Takýto vzťah môže byť orientovaný, kedy chová jeden aktér vzťah k druhému ale ten mu to neopätuje, alebo neorientovaný, kedy je vzťah vzájomný. Vzťah môže byť ohodnotený váhou (hodnota z určitého rozsahu), vyjadrujúcou napríklad intenzitu pocitov jedného aktéra k druhému, alebo ich vzájomnú blízkosť.

3.1.3 Klika

Klika (z francúzskeho clique) je taká skupina aktérov, v ktorej je každý jeden aktér vo vzťahu s ostatnými. Takéto skupiny, s najmenej tromi aktérmi, väčšinou nenaberajú veľké rozmery s ohľadom na veľkosť celej siete. Vhodným príkladom je niekoľko zamestnancov v malej kancelárii, každý každého pozná.

3.1.4 Komunita

Komunita (príp. cluster) je obecné taká skupina aktérov, v ktorej je viac vzťahov vo vnútri skupiny, ako vzťahov smerujúcich mimo. Vytváranie komunit je v sociálnych sieťach veľmi bežné, je pre ne typické že aktéri v rámci komunity majú väčšinou spoločné vlastnosti alebo spoločnú oblasť záujmov.

3.2 Tvorba vzťahov

Základným stavebným kameňom bežnej sociálnej siete je človek (prípadne skupina ľudí, organizácia), nazvime ho aktér. Jeho existencia v sieti by však nemala zmysel, keby v nej bol sám, preto vytvára väzby s inými aktérmi. Takýto vzťah umožňuje jeho účastníkom vymieňať si medzi sebou informácie a tým sa vzájomne ovplyvňovať. Môže vzniknúť čisto náhodne, alebo skrz spoločnú známosť dvoch aktérov, no väčšinou je založený na ich blízkosti či už geografickej, názorovej, alebo v podobe rodinných vzťahov. Aktér má tzv. sociálny dosah, čo v spoločenskej dimenzii predstavuje jeho okolie, v ktorom je najvyššia pravdepodobnosť že vytvorí nový vzťah. Toto okolie je kombináciou geografickej a spoločenskej blízkosti. Napríklad športovo založený aktér vytvorí nový vzťah pravdepodobnejšie s futbalistom bývajúcim na druhej strane ulice, ako s krčmárom z vedľajšej dediny. Tým vznikajú určité skupiny na miestach s vysokou hustotou podobne zameraných užívateľov, komunity. [13, 16]

3.3 Webové sociálne siete

V dnešnej dobe pojem sociálna sieť u väčšiny populácie vyvolá predstavu o internetovej stránke, na ktorú sa denno-denne pripájajú a trávajú pri nej čas. Pre túto prácu sú významné v tom zmysle, pretože samy o sebe poskytujú určitú úroveň abstrakcie nad sociálnymi sieťami v skutočnom svete, a je jednoduché z nich získať dáta o užívateľoch a vzťahoch medzi nimi, ktoré budú použité pre porovnanie s výstupmi výsledného modelu za účelmi jeho validácie.

Webovú sociálnu sieť (službu) môžeme definovať ako sieť prepojených (súvislý graf) verejných a poloverejných profilov, kde profil predstavuje užívateľov, skupinu, či udalosť, a prepojenie predstavuje vzájomné vzťahy medzi nimi. Realizáciou takýchto sociálnych sietí sú webové stránky určené na nadväzovanie a udržiavanie kontaktov medzi ľuďmi či organizáciami. Webové sociálne siete sú tvorené nielen profilmi a vzťahmi, ale aj obsahom, ktorým na stránky prispievajú. [8]

Svojej prudko rastúcej popularite v posledných rokoch vďaka zvýšenej dostupnosti internetu, a taktiež cenovej dostupnosti zariadení schopných pripojiť sa na internet, či už do pevnej siete alebo bezdrátovo. Sociálne siete sa stali dôležitým miestom pre stretávanie sa ľudí, komunikáciu, organizáciu rôznych podujatí či zdieľanie obsahu medzi užívateľmi. Človek dostáva priestor na vyjadrenie svojich názorov v trvalej a všetkým dostupnej forme. Takáto sociálna interakcia skrz internet prebieha na úkor reálneho osobného kontaktu, avšak väčšinou názor je, že výhody prevažujú nad nevýhodami.

3.4 Príklady sociálnych sietí

Nasleduje niekoľko príkladov skutočných a webových sociálnych sietí.

3.4.1 Trieda na základnej škole

Bežná sociálna sieť, skupina detí, ktoré sa stretávajú každý deň, jedná sa o kliku, keďže je predpoklad že každý pozná každého. Sieť je otvorená, to znamená že deti majú vzťah aj s inými ľuďmi mimo triedy. Sieť však nevznikla na základe spoločných záujmov, ale na základe podobného veku, geografickej blízkosti (deti chodia do najbližšej školy v okolí) a povinnej školskej dochádzky. Vzťahy sú vzájomné, a orientované len v tom zmysle že jedno dieťa môže mať k druhému ako druhé k nemu.

3.4.2 Základná škola

Založená na predchádzajúcom príklade, v tejto sieti sa takýchto tried nachádza viac. Deti v rámci tried vytvárajú vzťahy s deťmi z iných tried, avšak už sa nejedná o kliku, nie každý je vo vzťahu s každým. Vyskytujú sa tu aj neopätované orientované vzťahy, kedy osoba A predáva informácie o osobe B osobe C, takže osoba C sa dozvie o B ale pri tom B nepozná C, spája ich len spoločná známosť s osobou A. V prípade že má jedno dieťa veľa vzťahov a zároveň na neho smeruje veľa takýchto orientovaných jednostranných vzťahov, môžeme hovoriť o jeho popularite.

V kontexte základnej školy sa objavuje aj nový typ aktéra - učiteľ. Skôr ako s jednotlivými žiakmi je vo vzťahu s triedou ako celkom a s ostatnými učiteľmi.

3.4.3 Základné školy v jednom meste

Každá škola v tejto sieti tvorí komunitu, keďže deti medzi školami tvoria vzťahy len sporadicky oproti vzťahom v rámci ich školy.

3.4.4 Facebook

V roku 2012, len 8 rokov po svojom založení, mala sociálna sieť Facebook vyše jednu miliardu aktívnych užívateľov, vďaka je bezpochyby najrozšírenejšou sociálnou sieťou. Jedná sa o webovú službu, na ktorej si ľudia vytvoria profil, naplnia ho svojimi zážitkami, formujú priateľstvá, tvoria skupiny a komunikujú. Obzvlášť jednoduchá je pri takom množstve aktívnych užívateľov organizácia rôznych udalostí a záujmových skupín.

Zo vzťahov prevládajú jednoduché binárne neorientované, kde väčšinou musia užívatelia potvrdiť žiadosť o vzťah od iného užívateľa, výnimku čo sa týka orientovanosti tvoria profily verejných užívateľov (celebrity, firmy a pod.), kde iniciátor vzťahu len dá najavo že sa mu daný profil páči. Je možné v obmedzenej miere prideliť vzťahu váhu aj explicitne, ustanovením vzťahu ako rodinného alebo partnerského, inak vzťahy ohodnocuje vnútorná implementácia Facebooku na základe vzájomnej aktivity účastníkov a následne zobrazuje prednostne obsah tých užívateľov, s ktorými máme najlepší vzťah. Z pohľadu tejto práce sa naskytuje možnosť modelovať sociálnu sieť ako podmnožinu Facebooku, teda že by sa zvolili niektoré interakcie prebiehajúce medzi užívateľmi, ktoré by agenti mohli vykonávať aby vytvárali sociálnu sieť. Voľbu tejto možnosti podporuje jej prehľadný návrh, bohaté skúsenosti s jej používaním, ale hlavne ľahká dostupnosť štatistických údajov pre jej modelovanie. Modelovanie Facebooku ako celku a všetkých jeho možností by bolo príliš obsiahle a značne by prekročilo rámec bakalárskej práce.

3.4.5 Google+

Sociálna sieť od vyhľadávacieho giganta Google bola spustená v roku 2011. V princípe poskytuje tie isté možnosti ako Facebook, a niečo navyše k tomu. Google si od nej sľubovalo, že zoberie výrazný podiel užívateľov Facebooku, ale po počiatočnom návale ju veľa ľudí opustilo. Napriek tomu sa táto sieť chváli takmer štvrtí miliardou aktívnych užívateľov, čo sa však dá pripísať hlbokjej integrácii s ostatnými široko využívanými službami od Google. Navyše v porovnaní utrpel Google+ ďalšiu porážku, kde zatiaľ čo na Facebooku trávia užívatelia mesačne v priemere 7,5 hodiny, na Google+ strávia len niečo okolo 3 minút.

Google+ však priniesol nové prístupy, zamerával sa hlavne na klasifikáciu vzťahov do tzv. kruhov (užívateľských skupín) a ich váhové ohodnocovanie, už spomínanú integráciu s ostatnými Google službami ako mail a online dokumenty, a posunul do popredia audiovizuálnu komunikáciu. Pre modelovanie je zaujímavý hlavne koncept kruhov.

3.4.6 Twitter

Odvožené od slova tweet (pípanie, zvuk ktorý vydávajú vtáci), Twitter je založený na tom, že užívateľ na svoj profil posiela veľmi krátke správy a ostatní užívatelia ich môžu odoberať a sledovať. Veľmi jednoduchý princíp, pričom sa tu vyvinulo akési merítko popularity, kde je podstatný počet odoberateľov tweetov.

Systém umožňuje vytvárať len vzťahy jednoduché a orientované, čo však nevyklučuje možnosť vytvoriť takýto vzťah obojsmerne. Zaujímavý je práve fenomén popularity, ktorý vznikol nepriamo, mimovoľne, pôsobením veľkého množstva užívateľov/agentov, čo nám ukazuje ďalšiu výhodu použitia multi-agentných systémov, že niekedy vzniknú stavy, ktoré pôvodný návrh nemusel predpokladať (tzv. emergent phenomena). [9]

3.4.7 LinkedIn

Sociálna sieť zameraná predovšetkým na pracovný styk užívateľov, profil je v podstate životopisom. Všetky schopnosti a kvalifikácie sú okamžite dostupné, podporené referenciami od bývalých zamestnávateľov, ktoré sa dajú ľahko dohľadať a overiť. Okrem štandardných možností ako komunikácia či vytváranie spojení, poskytuje tiež možnosť inzercie pracovných príležitostí.

3.5 Model sociálnej siete

Aby sme mohli simulovať fungovanie sociálnej siete, musíme najprv navrhnuť jej model. Model je napodobenina systému iným systémom. Webové sociálne siete samy o sebe predstavujú model ľudskej spoločnosti s určitou úrovňou abstrakcie, avšak pre účely tejto práce navrhne simulačný model na ešte vyššej úrovni abstrakcie, keďže skutočné siete sú tvorené skutočnými inteligentnými ľudskými bytosťami, ktorých správanie sa s dnešnými technológiami nedá ani zďaleka nahradiť umelou inteligenciou. Preto budú jednotliví ľudia - agenti - modelovaní podmnožinou ľudských vlastností, pričom možnosti interakcie medzi nimi budú zredukované len na jednoduché základné.

Model bude okrem entít popisovať aj vzťahy, ktoré medzi nimi môžu vznikáť. Čo je však najdôležitejšie, bude popisovať akým spôsobom tieto vzťahy vznikajú. Práve tento spôsob je kritický pre validitu celého modelu a tým aj pre zmyslupnosť tejto práce, pretože ak vzťahy nebudú vznikáť tak, ako sa to deje v reálnom živote (alebo aspoň podobne), bude výsledný model nepoužiteľný v ďalšej práci.

3.6 Analýza sociálnej siete

Pri rozbere sociálnych sietí niesú stredobodom záujmu vlastnosti entít, ktoré sa v sieti nachádzajú (atribučné dáta), ale vzťahy, ktoré medzi nimi existujú (relačné dáta). Typ vzťahu a to, akým spôsobom vzniká, charakterizujú sociálnu sieť.

Pre vyhodnotenie porovnávania tohoto modelu sociálnej siete so skutočnou sociálnou sieťou bude potrebné jasne stanoviť sledované charakteristiky, ktoré budeme považovať za smerodajné pri určovaní validity modelu. Analýza sociálnych sietí ako technika modernej sociológie nám poskytuje sadu možností a nástrojov vďaka ktorým môžeme tento model správne ohodnotiť. Nebudú nás zaujímať charakteristiky jednotlivých vrcholov hrán v rámci siete, ale vlastnosti siete ako celku [6]. Vhodne ju opisujú nasledujúce vlastnosti:

- Priemerný stupeň vrcholu - koľko má v priemere vrchol hrán

$$E_{avg} = \frac{1}{n} \sum_{i=1}^n E_i$$

kde E_{avg} je priemerný stupeň a E_i je stupeň konkrétneho uzlu.

- Hustota - pomer počtu existujúcich hrán ku počtu možných hrán

$$D = \frac{E_{existent}}{E_{all}}$$

kde D je hustota, $E_{existent}$ je počet všetkých existujúcich a E_{all} je počet všetkých možných hrán.

- Reciprocita - pomer neorientovaných hrán k orientovaným

$$R = \frac{E_{mutual}}{E_{oriented}}$$

kde R je reciprocita, E_{mutual} je počet všetkých vzájomných a E_{all} je počet všetkých jednosmerne orientovaných vzťahov.

- Globálny clustrovací koeficient - miera do akej majú vrcholy tendenciu vytvárať cluster (cluster vzniká keď sú dvaja susedia vrcholu tiež susedmi). Je to podiel všetkých počtu uzatvorených ciest v sieti s dĺžkou dva ku počtu všetkých ciest dĺžky dva v sieti, pričom uzatvorená cesta je to isté čo cluster.

$$C = \frac{P_{closed}^2}{P_{all}^2}$$

kde C je koeficient, P_{closed}^2 sú všetky uzavreté cesty dĺžky dva a P_{all}^2 sú všetky cesty dĺžky dva. [2]

- Homofília - tendencia vrcholov vytvárať medzi sebou hrany ak majú spoločné vlastnosti, základ pre vznik komunit. Komunitou v sieti rozumieme huste poprepájanú skupinu agentov, v ktorej je viac hrán s oboma koncami vo vnútri skupiny, ako hrán s jedným koncom mimo skupiny.

Pre analýzu boli použité nástroje Wolfram Mathematica 9 a Pajek.

Kapitola 4

Návrh modelu sociálnej siete

Model pozostáva z dvoch vrstiev. Jedna predstavuje fyzické umiestnenie agenta vo svete, to kde sa pohybuje a kde trávi čas, fyzický svet. Druhá vrstva predstavuje model interakcií medzi jednotlivými agentami, spôsob akým spolu komunikujú a vytvárajú vzťahy, nazvime ho spoločenským svetom.

4.1 Fyzický svet

Predstavuje priestor, v ktorom agenti žijú a pohybujú sa. Pozostáva z rôznych miest, avšak pre zachovanie jednoduchosti modelu boli zredukované na domy (obydlia), pracoviská a miesta pre zábavu. Pre napodobenie reality v tomto svete beží čas vo forme 24-hodinových dní. Agenti sa potom v rôznych dobách dňa presúvajú na rôzne miesta a tu trávajú určitý čas. Motiváciou k týmto presunom sú im potreby, ako napríklad potreba zarobiť si peniaze, potreba zabávať sa, alebo potreba odpočinúť si doma. V realite sú samozrejme potreby komplikovanejšie, avšak pre naše účely postačia tieto zjednodušené. Pre ich realizáciu stačia dve numerické hodnoty popisujúce agenta - peniaze a nálada.

Peniaze agent získava v práci, na rôznych pracoviskách sú odlišné platy, takže zarábajú rôznym tempom. Agent peniaze míňa priebežne keď je vonku a zabáva sa. Aby mohol ísť vôbec von, musí disponovať určitou sumou peňazí, v prípade že tomu tak nieje, radšej ostane sedieť doma, odpočine si, na druhý deň zarobí ďalšie peniaze, a po práci pôjde von. Stav financií je vyjadrený celočíselne, podobne ako na účte v banke. Agent začína s nulou, mzdy sa pohybujú medzi hodnotou 20 a 50 (fiktívna mena)

```
1 int money = 0;
  int salary = random.integer(20..50);
3 double mood = random.float(0..1);
```

```
1 if (at_Home){
    mood += random.float(0..0.1); }
3 else if (at_Work){
    money += salary;
5    mood -= random.float(0..0.1); }
  else if (at_Fun){
7    money -= random.integer(50..80);
    if (money < 80) { goHome(); }
9 }
```

Nálada predstavuje duševný stav agenta, určuje, či sa mu po dlhom a ťažkom dni strávenom v práci bude vôbec chcieť ešte niekam ísť sa zabávať. Kým je agent doma, relaxuje a jeho nálada stúpa. Naopak, v práci jeho nálada klesá, dalo by sa povedať že sa prácou unavuje a večer už nemá energiu niekam ísť. Je vyjadrená reálnym číslom v rozmedzí od nuly po jedna. Na začiatku má agent náladu náhodne určenú v tomto rozsahu.

Obe tieto vlastnosti ovplyvňujúce fyzickú polohu agenta sú vyjadrené numerickými hodnotami, a takisto ako pri vytváraní fyzickej reprezentácie klienta (miesto kde býva, pracuje, jeho charakter) sú do značnej miery ovplyvňované náhodou, čiže rôzni agenti budú mať odlišné platy, rýchlosť obnovovania nálady a podobne.

Agenti pri výbere miesta kam sa pôjdu zabaviť nerozmýšľajú náhodne. Spočiatku tomu tak je, avšak v prípade že na niektorom takomto mieste založí nové priateľstvo s iným agentom, v budúcnosti bude vyššia pravdepodobnosť že si vyberie toto miesto, ako niektoré iné, keďže bude predpokladať že tu má väčšiu šancu naraziť na jemu podobných agentov (za predpokladu že oni rozmýšľajú rovnako ako on). Takéto správanie je prirodzené, a napomáha k vytváraniu komúnít v rámci sociálnej siete.

4.1.1 Mentálny stav agenta

Pri vytváraní simulačného modelu je možné zadať množinu tém, o ktorých sa agenti môžu medzi sebou rozprávať, takže každý agent je zároveň charakterizovaný aj mierou, do akej sa o tú ktorú tému zaujíma. Táto miera je vyjadrená reálnym číslom v rozmedzí nula až jedna, pričom čím je číslo vyššie, tým kladnejší vzťah má agent k téme. Mimo to, agenta popisuje aj jeho "dobrota", spôsob akým pristupuje k iným agentom, či je k ním milý alebo sa správa nepríjemne, taktiež vyjadrená reálnym číslom od nula po jedna. Táto vlastnosť však má svoje opodstatnenie len v spoločenskej vrstve, na pohyb agenta vo svete nemá žiadny vplyv.

Vlastnosť	Hodnota
Nálada	0.487
Dobrosť	0.098
Politika	0.135
Šport	0.864
Kultúra	0.934

Tabulka 4.1: Príklad mentálneho stavu agenta.

V tabuľke 4.1 sú uvedené duševné vlastnosti popisujúceho agenta, ktorý je vyslovene zlý, nezaujíma sa o politiku, ale naopak obľubuje šport a ešte viac kultúru.

Tu je na mieste podotknúť, že prepojenie medzi fyzickým svetom a sociálnym svetom je okrem osobných záujmov a výberu miesta pre zábavu minimálne. To znamená, že agenta nemusí zaujímať ako sa na dané miesto dostal, zaujíma ho len že koho tam stretol a komunikácia s ním. Fyzický pohyb je v úzadí, poskytujúci len základ pre založenie komunikácie na základe stretnutí dvoch agentov v priestore.

4.2 Spoločenský svet

Je tvorený agentami a všetkými vzťahmi, ktoré medzi jednotlivými agentami vznikajú. Môžeme si ho predstaviť ako graf s ohodnotenými orientovanými hranami, kde váha hrany vyjadruje, do akej miery jeden agent obľubuje druhého. Váha je vyjadrená numericky, reálnym číslom v rozsahu nula až jedna. Model poskytuje možnosť jednostranných priateľstiev, kde jeden agent chová kladný vzťah k druhému agentovi, ktorý mu to však už neopätuje. Takéto vzťahy sú typické aj pre skutočný svet.

4.2.1 Nadväzovanie kontaktu

Sociálna interakcia medzi agentami začína v okamžiku, kedy sa stretnú vo fyzickom svete a dostanú podnet z „očí“, že vidia nejakého iného agenta. V prípade, že sa s ním ešte nestretli, zapamätajú si že existuje a vytvoria si o ňom svoju mienku, prvý dojem. Keďže jeho charakter ešte nepoznajú a nemajú predsudky, ich názor naňho nieje vyhranený žiadnym smerom (vzťahu sa priradí stredná hodnota, tzn. 0.5). Ten sa bude vyvíjať až pri ďalších interakciách.

```
1 if (see(agentX) AND dontYetKnow(agentX)) {  
    remember(agentX); like(agentX, 0.5);  
3 }
```

V prípade, že agent je v dobrej nálade a má chuť sa rozprávať a byť milý, pozdraví druhého agenta a pokúsi sa o rozhovor (splnenie podmienky byť milý a mať náladu sa vyhodnotí tak, že sa získa aktuálna hodnota ukazateľa vlastnosti, vygeneruje sa náhodné číslo a pokiaľ je vyššie ako aktuálna hodnota, podmienka je splnená a agent je v nálade/milý).

Podmienkou pre začiatok hlbšej komunikácie však je, aby aj druhý agent zdieľal jeho náladu a chuť rozprávať sa. V prípade že tomu tak nieje, odmietne ponuku na rozhovor. Toto odmietnutie však nespraví dobrý dojem na prvého agenta, ktorému o tomto agentovi mierne klesne mienka (-0.05), avšak nezanevrie naňho a nezabráni mu to v budúcnosti pokúsiť sa o kontakt znova.

V prípade, že druhý agent okrem toho že nemá náladu na rozhovor je ešte aj zlá osoba, je vysoká pravdepodobnosť že iniciátora odmietne nevyberaným spôsobom, napríklad že ho hrubo pošle preč (-0.1). Takáto odpoveď už zanechá v prvom agentovi značne negatívne dojmy, a nabadúce si dobre rozmyslí, či sa druhému agentovi opäť pokúsi prihovoriť (úroveň vzťahu klesne pod hodnotu 0.3).

Agent	Hodnota vzťahu	Prístup
Agent A	0.846	nadviaže kontakt
Agent B	0.500	nadviaže kontakt, nová známosť
Agent C	0.261	nenadviaže kontakt
Agent D	0.579	nadviaže kontakt

Tabulka 4.2: Príklad vzťahov agenta.

V tabuľke 4.2 sú uvedené vzťahy, ktoré chová agent k ostatným agentom. Je vidieť že Agent A nemá príliš v obľube, preto s ním už v budúcnosti sám nenadviaže kontakt

4.2.2 Komunikácia

Ak druhý agent prvému agentovi jeho počiatočné nadšenie opätuje, môžu zahájiť rozhovor a vzájomne sa začať spoznávať. Prvá vec, ktorá je nutná pre konverzáciu, je výber témy. Agent, ktorý interakciu započal, je natoľko slušný že nechá druhého agenta aby vybral o čom sa budú rozprávať. Pripomeňme si, že množina tém je dopredu daná a každý agent má ku každej z nich dopredu určený vzťah. Pri výbere je vždy väčšia pravdepodobnosť, že agent si zvolí skôr tému ku ktorej má blízko, ako tému ktorú príliš neobľubuje.

Až je rozhodnutý, spýta sa prvého agenta na jeho postoj k danej téme. V závislosti od toho, do akej miery k nej zdieľajú náklonnosť, vyhodnotia konverzáciu. V prípade že sa im páčila, ich vzájomný vzťah jeden k druhému sa zlepši, v prípade že nie, bude tomu naopak. Obzvlášť veľký dôraz kladú agenti na výsledok rozhovoru (výpočet v 5.1.6) v prípade, že sa jedná o ich najobľúbenejšiu tému, ktorá pre nich veľa znamená. V tomto prípade sú značne citliví na negatívny postoj druhého agenta a naopak sú šťastní keď druhý agent ich nadšenie zdieľa. Po tejto zbežnej konverzácii sa agenti rozlúčia a každý si ide ďalej svojou cestou.

```
1 double conversation_success = talkAboutWithith(Topic, AgentX);  
2 if (Topic.isImportantForMe())  
3     like(AgentX, + 2 * conversation_success);  
4 else  
5     like(AgentX, + conversation_success);
```

Takto to pri ich opakovaných stretnutiach pokračuje a ich sympatie jeden k druhému rastú a klesajú, až kým nedosiahnú určitej hranice. V prípade, že má prvý agent k druhému po niekoľkých rozhovoroch vyslovene negatívny postoj, už sa mu nikdy viac neprihovori (môže však nastať špeciálny prípad že si ho tretí agent bude pochvaľovať a tým zvýši mienku prvého). Ak nastane opačný prípad (hodnota úrovne vzťahu stúpne nad 0.7) a prvý agent má druhého agenta rád, začne ho považovať za svojho priateľa (pozor, neznamená to automaticky že druhý agent považuje prvého za priateľa tiež). V rámci priateľstva potom bude k nemu pristupovať odlišným spôsobom. V závislosti od jeho dobroty a aktuálnej nálady si vyberá čo spraví.

Ak sa cíti byť dobrý a v nálade rozdáva v rámci konverzácie jednoduché komplimenty, na ktoré však musí mať náladu aj príjemca, keďže sa niekedy stane že napriek dobrému úmyslu si lichôtky vyloží nesprávne a urazí sa. Vo väčšine prípadov agent započne obyčajný rozhovor o niektorej z tém, ktorý bol popísaný vyššie, len s tým rozdielom, že pri opakovaní témy už nemá výsledok rozhovoru na vzájomný vzťah taký vplyv (úroveň vzťahu sa zmení len o desatinu pôvodnej hodnoty), pretože vzťah druhého agenta k danej téme je už dávnejšie známy.

Vo svete sa pohybujú aj agenti s menej vyvinutými morálnymi hodnotami. Keď majú zlú náladu a sú s priateľmi ktorým veria, začnú sa rozprávať o iných agentoch, teda podsúvať svoj názor na jedného agenta inému agentovi. V prípade že ich priateľ ohováraného agenta zatiaľ nepozná, má takáto informácia väčší vplyv na ich predstavu o ňom, ako keď ho poznajú, keďže o ňom majú utvorený už svoj vlastný názor.

```
1 if (feel_like_being_bad){  
2     Subject_agent = getRandomAgent();  
3     tell(Conversation_partner, howMuchILike(Subject_agent));  
4 }
```

Takto spolu rôzne dvojice agentov komunikujú, vytvárajú si názory jeden na druhého a vytvárajú vlastnú sociálnu sieť.

4.3 Predpokladaný výstup

Po správnej implementácii návrhu by výsledkom mal byť multi-agentný systém simulujúci správanie reálnych sociálnych sietí. To znamená, že po vytvorení simulačného prostredia a dostatočného množstva v ňom pohybujúcich sa agentov, títo agenti začnú vytvárať vzťahy a vymieňať si medzi sebou informácie. Takýto systém by mal ostať stabilný (nebudú sa kamarádiť všetci so všetkými) avšak mal by ostať zachovaný aj určitý spoločenský ruch, typický pre skutočné systémy.

Aby sme dokázali či výsledný systém spĺňa tieto požiadavky, bude potrebné porovnanie s reálnymi sociálnymi sieťami. To bude uskutočnené porovnávaním výstupov systému zo série experimentov s nazhromaždenými skutočnými štatistickými dátami. Smerodatné sú informácie ako počet vzťahov, ktoré si agent vytvorí, priemerná spoločenská vzdialenosť účastníkov vzťahu, podiel orientovaných a neorientovaných vzťahov, a podobne. Pre názornosť budú porovnané aj grafické zobrazenia agentov a ich vzťahov spoločenskom priestore. Dôraz bude kladený na zoskupovanie agentov podľa záujmov a na vytváranie vzťahov.

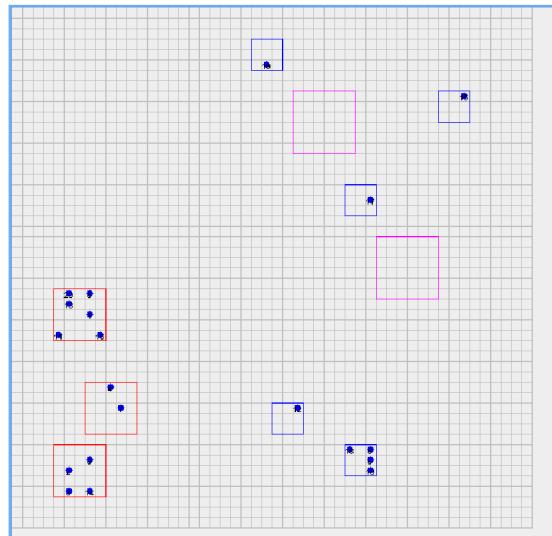
Kapitola 5

Implementácia

Výstup tejto práce je program napísaný v jazyku Java a jazyku AgentSpeak na platforme Jason. Pre vývoj bolo použité voľne dostupné prostredie Eclipse. Na vstupe programu je konfiguračný súbor, v ktorom je určené parametre simulácie ako veľkosť sveta, počet miest, množstvo agentov, dĺžka simulácie a sada výstupných súborov, do ktorého sa na konci zaznamená vytvorená sociálna sieť.

5.1 Java

Program nemá grafické užívateľské rozhranie s výnimkou možnosti zobrazenia fyzického sveta vo forme dvojrozmernej mriežky, v ktorej sú znázornené jednotlivé miesta (ako farebné štvorce v závislosti od typu) a pohybujúci sa agenti (vo forme očíslovaných bodov, ktoré časom menia svoju polohu). Toto zobrazenie nemá takmer žiadny praktický význam, slúži skôr ako názorná ukážka toho že svet agentov žije. Je vytvorené pomocou štandardného balíka Java.awt, určeného pre tvorbu jednoduchej grafiky a užívateľských rozhraní.



Obrázek 5.1: Grafické znázornenie agentov vo svete

Samotný program je postavený na architektúre **Model-View-Controller**, ktorý od seba oddeluje riadiacu logiku, dátový model a užívateľské rozhranie.

5.1.1 View

View predstavuje grafickú časť, ktorá je tvorená triedou **WorldView** v súbore `WorldView.java`. Dedí triedu **GridWorldView** frameworku Jason, ktorá je prichystaná práve pre jednoduché vykresľovanie agentov v dvojrozmerných svetoch (mriežkach). Táto jednoduchá funkcionálna vo veľkom množstve prípadov multi-agentnému systému dokonale postačuje, čo je aj náš prípad. Je mierne upravená funkcia vykresľujúca objekty tak, aby mohla vykresliť okrem agentov aj jednotlivé miesta. Tie sú potom podľa typu odlíšené farebne a veľkosťou.

5.1.2 Model

Je riešený v súbore `WorldModel.java`. Trieda sa nazýva **WorldModel** a dedí triedu **GridWorldModel** frameworku Jason, ktorá je kompatibilná s triedou **GridWorldView**, takže o vykresľovanie sa takmer netreba starať, stačí agentom prideliť miesto. Táto trieda zastrešuje dátový model fyzického sveta. Pri stvorení sveta (big bang) sa vygenerujú polohy dopredu určeného množstva domov, pracovísk a klubov (miest kam sa chodia agenti zabávať) podľa možnosti tak, aby sa neprekrývali. Následne sa umiestnia všetci agenti do svojich náhodne vybraných domovov pričom model zabezpečí, aby žiadny dom neprekročil svoju kapacitu. Model neskôr už len udržiava informácie o svetovom čase, polohe miest a polohe agentov, a prijíma požiadavky od fyzických reprezentácií agentov na presun do toho ktorého miesta. V prípade že dané miesto nieje plne obsadené agenta presunie, inak mu nahlási chybu, a agent musí zmeniť plány a ostať tam kde bol.

5.1.3 Controller

Časť Controller je implementovaná v triede **World** v súbore `World.java`. Rozširuje triedu **Environment** frameworku Jason a predstavuje most medzi fyzickým svetom a spoločenským svetom, vykonáva totiž akcie spúšťané agentami v časti kódu písanom v `AgentSpeaku`.

Pri spustení celého programu sa celý zoznam parametrov z konfiguračného súboru predáva jej inicializačnej funkcii, ktorej teda prípadne úloha postupne vytvorí a inicializovať všetky ostatné časti programu. Vytvorí nové objekty triedy **WorldModel** a triedy **WorldView**, ktorej predá ukazateľ na `WorldModel` aby mohla vykresľovať jeho obsah. Následne vytvorí určené množstvo agentov, vloží ich do modelu a prvý krát aktualizuje ich vnemy, aby videli kto sa na začiatku nachádza v ich okolí, a spustí jedného agenta dozorca, ktorý má na starosti aktualizáciu svetového času a ukončenie celej simulácie v prípade splnenia podmienky dosiahnutia časového limitu určeného v konfiguračnom súbore.

Po rozbehnutí sveta čaká program na podnety k vykonaní nejakej akcie od agentov. To sa deje zakaždým keď dostáva od dozorca hodinový tik (dozorca spí 500 milisekúnd, kratšie intervaly spôsobujú chybné výstupy), ktorý znamená že čas vo svete sa posunul o jednu hodinu a agenti podľa toho môžu zmeniť svoju polohu a potom získať vnemy zo svojho okolia. Najprv vymažú svoje hodinu staré a už neaktuálne vnemy, získajú vnemy nové a vykonajú jeden krok svojej rozhodovacej logiky (popísané v 5.1.5). Medzi tieto vnemy patrí ich aktuálna nálada a pohľad do ich okolia (osem-okolie v rámci dvojrozmernej svetovej mriežky), či sa tam nenachádza žiadny iný agent. Objekt typu **World** dostáva podnety aj od agentov - osôb, ktoré posielajú žiadosti o zaznamenanie aktuálneho vzťahu k inému agentovi, alebo o zaevidovanie (prípadne vyradenie z evidencie) iného agenta ako priateľa. Pri ukončení behu programu sa zavolá funkcia, ktorá otvorí výstupné súbory a pre každého agenta do nich zapíše jeho vzťahy, podľa požadovaných parametrov.

5.1.4 Telo agenta

Samotná entita agenta vo fyzickom svete je reprezentovaná triedou **MyAgent** v súbore `MyAgent.java`. Táto udržiava informácie o tom, ako sa agent volá, kde býva, kde pracuje, kam rád chodí zabávať sa, či je dobrý alebo zlý, v akej je nálade, o čo sa zaujíma, ktorých ďalších agentov pozná a ktorých z nich považuje za priateľov, a čo má agent najbližšie v pláne. Pri vytvorení nového objektu typu **MyAgent** sa všetky tieto prvky inicializujú a niektorým sa priradí náhodná hodnota v rámci ich oboru hodnôt. To zahŕňa aj určenie do akej miery sa agent zaujíma o témy v zozname tém v triede **Topic** (kde sú ako verejný statický zoznam, aby sa cez ne neskôr dalo pohodlne iterovať a indexovať).

5.1.5 Rozhodovacia logika agenta

Rozhodovacia logika agenta je riadená v krokovej funkcii, kde zistí čas z modelu a v prípade že na túto hodinu mal naplánovanú zmenu polohy, pokúsi sa ju vykonať (čo sa mu vo veľkej väčšine prípadov podarí). Ďalej sa mu v závislosti od miesta kde sa nachádza upravuje nálada a stav jeho financií. Keď sa presunie do práce alebo do klubu, určí v rámci určitého rozmedzia čas, ktorý tam pobudne. V prípade že je v klube stav jeho účtu klesne pod určitú úroveň odchádza domov. Čo sa týka klubov, pre každý z nich má numerický ukazateľ obľúbenosti, ktorý je spočiatku na hodnote jedna. Zakaždým keď získa nového priateľa agenta v niektorom z klubov, táto hodnota sa inkrementuje. Ak sa potom bude rozhodovať, do ktorého klubu zájde, bude si vyberať náhodne pomocou funkcie rozloženia hustoty pravdepodobnosti, teda vyberie náhodne jeden z klubov, a jedno číslo v rozmedzí nula až hodnota obľúbenosti najobľúbenejšieho baru. V prípade že obľúbenosť tohoto klubu je vyššia ako toto číslo, navštívi tento klub, inak pokračuje touto metódou v novom cykle. Týmto sa docielu zvýšenie pravdepodobnosti, že bude navštevovať klub, v ktorom má najviac priateľov.

V triede **MyAgent** prebieha aj zaznamenávanie známostí a priateľov, konkrétne sa ukladá úroveň na akej agent druhého agenta obľubuje. Sú tu dva zoznamy, v jednom sú len agenti ktorých spoločenská reprezentácia agenta považuje za priateľov, a v druhom sú všetci o ktorých agent vie že existujú, spolu s numericky vyjadrenou hodnotou ich vzťahov. Tieto zoznamy sú po skončení simulácie výstupom ďalej vhodným pre analýzu, zapisujú sa vo formáte `.dot`, určenom pre ukladanie grafov. Umožňuje ukladať orientované či neorientované grafy s popismi vrcholov a hrán, a taktiež s váhovým ohodnotením hrán.

5.1.6 Interné akcie

Okrem týchto tried program zahŕňa ešte aj takzvané interné akcie, ktoré sú použiteľné priamo z kódu agenta v `AgentSpeaku`. Sú odvodené z triedy **DefaultInternalAction** platformy `Jason` a používajú jednoduché rozhranie k predávaniu hodnôt. V tomto programe sú vytvorené nasledujúce interné akcie:

- `talkSuccess(.java)`, ktorá vyhodnotí úspešnosť konverzácie medzi dvoma agentami

```
double x = AgentA.likesTopic(Topic);
2 double y = AgentB.likesTopic(Topic);
double diff = Math.abs(x - y);
4 return ((0.5 - diff) / 2) * ((a1 + a2) / 2);
```


- **pickRandomTopic(.java)**, ktorá zvolí náhodnú tému zo zoznamu tém (opäť funkcia rozloženia hustoty pravdepodobnosti, takže agent uprednostňuje tému ktorú má radšej)
- **getGossipTarget(.java)**, ktorá vyberie náhodne jedného nešťastného agenta, ktorý bude predmetom konverzácie.

5.2 AgentSpeak

Časť program napísaná v jazyku AgentSpeak predstavuje jeho jadro. Okrem krátkeho úseku kódu starajúceho sa o čas a ukončenie behu simulácie (dozorca - supervisor.asl) sa veškerá logika nachádza v súbore person.asl, ktorý opisuje správanie jednotlivca - agenta - v rámci multi-agentného systému. Inštancie tohoto agenta budú komunikovať výhradne medzi sebou, takže ak sa v kóde vyskytuje nejaká otázka smerovaná na iného agenta, reakcia je implementovaná v rámci toho istého súboru. Program v jazyku AgentSpeak pozostáva z pravidiel, cieľov a plánov.

Ako prvé sú zadefinované takzvané pravidlá - určité výrazy, ktorých vyhodnotením zistíme či sú splnené nejaké podmienky pre vykonanie niektorej varianty plánu neskôr v kóde. **inMood** zistí či je agent v nálade na rozhovor, **beNice** určí či sa agent zachová slušne, **likeHim(ID)** vyhodnotí či agent má alebo nemá rád agenta ID, a **isTopicImportant(Topic)** napovie či je téma rozhovoru pre agenta dôležitá (či je z pomedzi všetkých tém najobľúbenejšia).

Jediný cieľ ktorý dostáva agent na začiatku svojej existencie je cieľ **!live**, teda aby žil. Ten však v pláne na svoje vykonanie čaká a zadržuje celý ďalší beh programu agenta až kým mu controller nepošle vnem **created**, ktorý je mu daný až keď je všetko pripravené, aby sa zabránilo neoprávneným prístupom do pamäte a podobným neprijemnostiam. Potom pokračuje čakaním na vnemy zo svojho prostredia.

5.2.1 Stretnutie

Interakcia začína keď agent dostane vnem **see(ID)**, teda zbadá vo svojom okolí iného agenta (ID je jeho identifikátor). Vtedy si do svojej báze znalosti pridá fakt, že ho pozná, a úroveň jeho vzťahu k nemu, **like(ID, 0.5)**, čo vyjadruje jeho neutrálny postoj. V prípade že má náladu na rozhovor, a druhý agent ho ešte nestihol pozdraviť ako prvý, pozdraví sa mu tým, že mu pošle správu **hello(MyName)**, kde MyName je jeho meno, aby vedel od koho pozdrav prišiel.

Pozdravený agent nemusí mať náladu pokračovať, takže mu vráti späť správu **sorry(MyName, X)** na ktorú agenti reagujú tak, že si znížia úroveň ich vzťahu k MyName o X. V prípade že pozdravený agent chce byť zlý, odmietne prvého agenta nevyberaným spôsobom, čiže mu pošle správu **sorry** s vyššou hodnotou X. Pokiaľ konverzovať chce a zatiaľ nepovažuje vyzývateľa za priateľa, pridá sa mu cieľ **!talk**, s priateľom sa mu pridá cieľ **chat**.

5.2.2 Rozhovor

Plán pre cieľ **talk** pozostáva z výberu témy pre rozhovor za pomoci internej akcie **pickRandomTopic** a plánu **talkAbout**, ktorému sa téma predá ako parameter. V prípade že sa o tejto téme agenti ešte nerozprávali spýta sa vykonávajúci agent druhého na jeho názor na túto tému. Potom prebehne vyhodnotenie úspešnosti tejto konverzácie pomocou internej

akcie `talkSuccess`, a obaja agenti si zapamätajú túto hodnotu pre danú tému pomocou pridania si faktu `talkedAbout(With, Topic, X)`, kde `X` je výsledok. Pridaním tohoto faktu sa spustí plán, ktorý upraví úroveň vzťahu k druhému agentovi o vypočítanú hodnotu (tá môže byť kladná aj záporná). Ak je téma pre agenta dôležitá, je úroveň vzťahu upravená o dvojnásobok tejto hodnoty. Pokiaľ sa agenti o téme už stihli v minulosti rozprávať, tiež sa upraví sa ich úroveň vzťahu, avšak len o desatinu pôvodnej hodnoty.

Plán pre cieľ `chat` je jednoduchý, pozostáva z výpočtu aktuálnej dobroty agenta a pridania cieľu `!courseOfAction`, ktorý ako parameter dostane túto hodnotu. Plán pre cieľ `courseOfAction` sa vetví podľa dobroty agenta. V prípade že je vysoká, pošle druhému agentovi kompliment (ten ho môže prijať pozitívne či negatívne), ak je v stredných hodnotách pokračujú cieľom `talk`, v prípade že je nízka si agent vyberie nejakého náhodného známeho pomocou internej akcie `getGossipTarget` a pošle druhému agentovi svoj názor na tohoto agenta správou `gossip(MyName, Target, X)`, kde `X` je úroveň vzťahu `MyName` k `Targetu`. Ten v prípade že ho ešte nepozná, utvorí si o ňom mienku značne ovplyvnenú týmito informáciami, v prípade že ho pozná bude ovplyvnený menej.

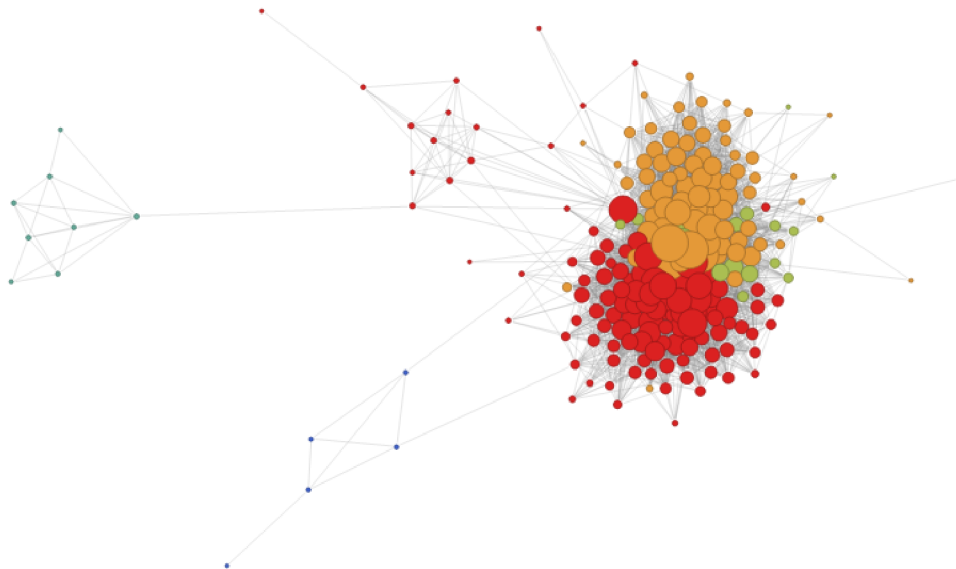
Kapitola 6

Analýza výsledného modelu

Pre porovnanie a zhodnotenie bude použitý výstup z behu simulácie s priloženou konfiguráciou. Počet agentov, ktorí v simulácii účinkovali bol zvolený tak, aby korešpondoval s druhým zdrojom dát pre porovnanie, sociálnou sieťou tvorenou tým istým množstvom mojich priateľov na Facebooku. Vzhľadom na to, že tento model sa snaží dosiahnuť prirodzenej štruktúry sociálnej siete, čím Facebook dozaista disponuje, jedná sa o vynikajúci zdroj dát pre komparáciu. Tieto dáta sú vďaka rôznym nástrojom ľahko vydolovateľné, použitá bola konkrétne Mathematica 9, v ktorej budú vykreslené aj nasledujúce grafy.

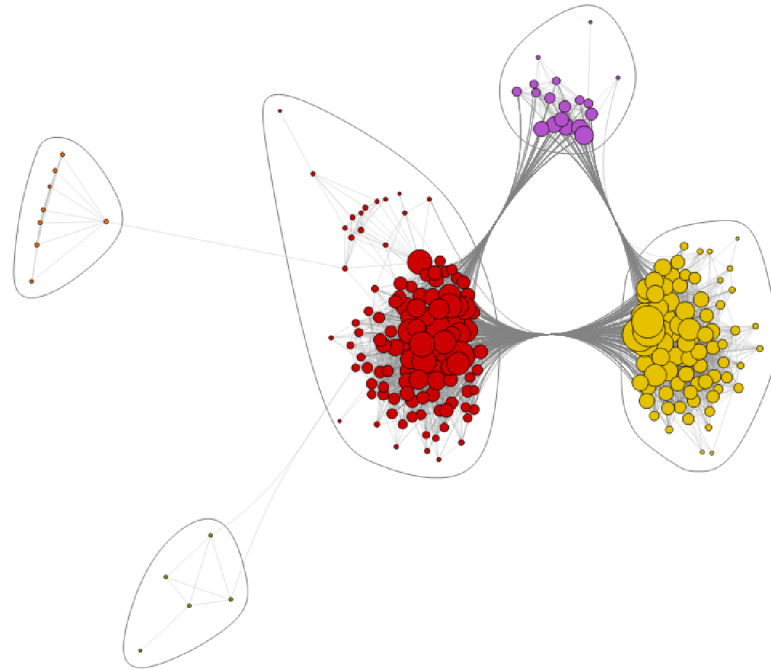
6.1 Vzorová sieť z Facebooku

Jedná sa o sieť 227 užívateľov, ktorí majú všetci spoločnú vlastnosť, že sú vo vzťahu s jednou osobou (v tomto prípade somnou). Na grafe 6.1 je dobre viditeľná geografická odčlenenosť istých malých skupín od majority.



Obrázek 6.1: Sieť mojich priateľov na Facebooku.

Na grafe 6.1 je znázornená tá istá sociálna sieť, avšak v inom rozložení a so zvýrazneným začlenením aktérov do komunit. To rozdelenie súhlasí, napríklad veľká červená (stred) komunita predstavuje moje známosti zo stredoškolských čias, pričom korektne oddelená žltá komunita (vpravo, ktoré sa v predchádzajúcom zobrazení značne prekrývali) je skupina ľudí, s ktorými sa stretávam teraz.



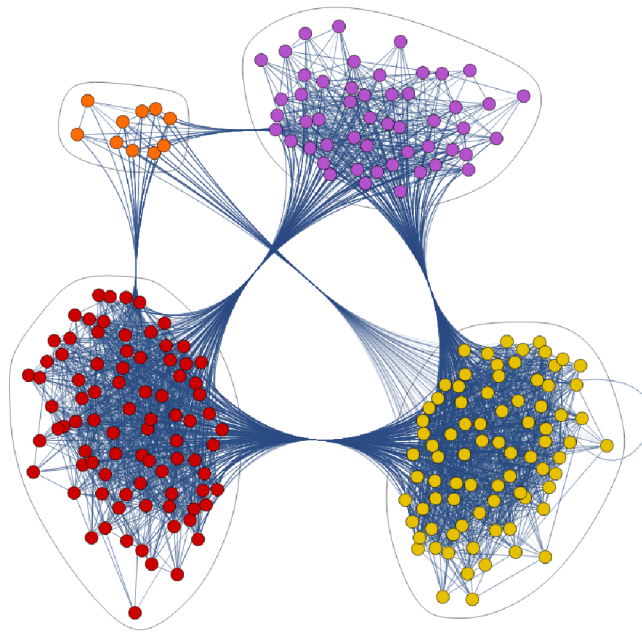
Obrázek 6.2: Sieť mojich priateľov na Facebooku so zvýraznenými komunitami.

6.2 Výstup modelu - všetky známosti

Množstvo agentov v simulácii sa zhoduje s množstvom priateľov na Facebooku, aby boli siete porovnateľné. Taktiež bola simulácia nastavená tak, aby sa na výstup dostali len vzťahy s hodnotou 0.5 a vyššou, čo predstavuje v spoločenskom slovníku rozpätie od bežnej známosti až po najlepšieho priateľa. Bežné známosti sú zahrnuté tiež, keďže podobné vzťahy nie sú ničím výnimočné ani na facebooku, kedy sa dvaja ľudia stretnú raz v živote a pridajú sa medzi priateľov. Na obrázku 6.2 je graf sociálnej siete vygenerovanej implementovaným modelom. Možno si všimnúť, že na rozdiel od skutočnej siete sa tu nevyskytujú skupiny vzdialené od väčšiny v strede, avšak tento fakt môžeme pripísať tomu že agenti sa pohybujú narozdiel od skutočného sveta len v jednom uzatvorenom priestore.

6.3 Výstup modelu - priatelia

Táto sieť je oproti predchádzajúcim značne riedka, keďže sa jedná len o veľmi dobré vzťahy. Napriek tomu si ju tu pre porovnanie uvedieme. Na grafickom znázornení 6.3 je dobre vidieť vlastnosť sociálnej siete - vytváranie vzťahov a komunit na základe homofílie. Koláčové grafy nahrádzajúce plné kruhy na mieste vrcholov nám dávajú vedieť, v akom pomere ten



Obrázek 6.3: Sieť vygenerovaná modelom.

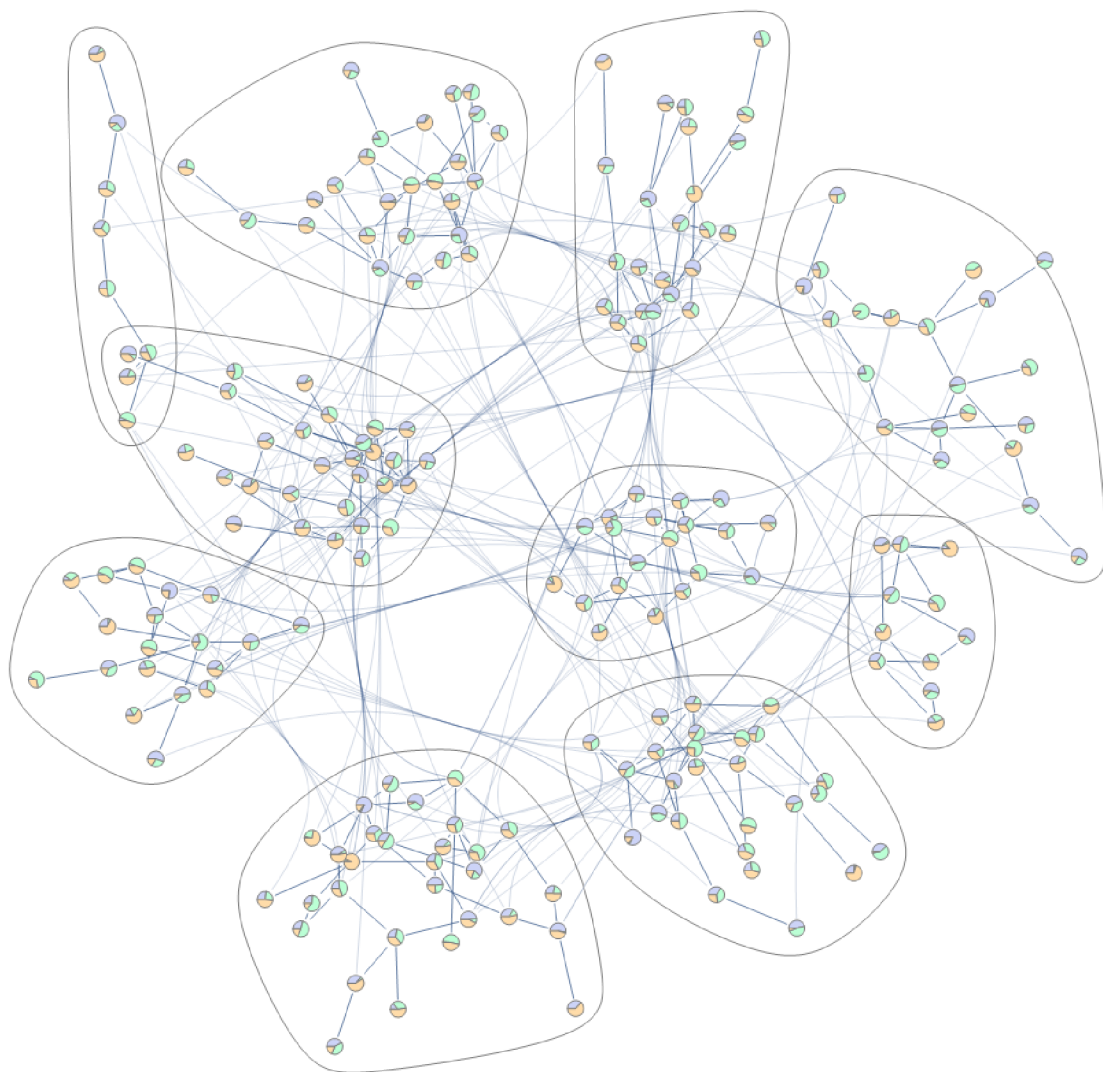
ktorý agent obľubuje určitú tému. Na základe navrhnutého modelu sa častejšie do komúnít spájajú agenti so spoločnými záujmami, čo je viditeľné ako dominancia niektorej farby v skupine.

6.4 Štatistické porovnanie

	Facebook	Sieť známostí	Sieť priateľov
Priemerný stupeň vrcholu	43.921	41.401	4.327
Hustota	0.193	0.182	0.020
Reciprocita	1.0	0.693	0.468
Globálny clustrovací koeficient	0.491	0.230	0.027

Tabulka 6.1: Štatistické porovnanie sociálnych sietí.

Z údajov uvedených v tabuľke 6.1 je zrejmé, že zatiaľ čo riedka podmnožina priateľov v porovnaní neobstojí, širšia vzorka vygenerovanej siete sa čo sa týka hustoty a stupňa vzťahov pohybuje v rovnakej oblasti ako skutočná sociálna sieť. Reciprocitu vzťahov nieje možné objektívne porovnať, pretože vzťahy vytvorené na Facebooku sú automaticky neorientované a vzájomné. Jediný výrazný rozdiel nastáva pri globálnom clustrovacom koeficiente, kde Facebook dosahuje dvojnásobných hodnôt. Tento fakt možno pripísať tomu, že Facebook je narozdiel od tohoto modelu otvorená sociálna sieť obrovského rozsahu, z ktorej skúmame len určitú nepatrnú podmnožinu.



Obrázek 6.4: Sieť priateľov vygenerovaná modelom.

Kapitola 7

Záver

Cieľom práce bolo navrhnuť a vytvoriť funkčný model sociálnej siete pomocou multi-agentných systémov. Je implementovaný v jazykoch Java a AgentSpeak na platforme Jason. Výstupom simulácie tohoto modelu je dátová štruktúra reprezentujúca sociálnu sieť, akýsi graf, kde vrcholy predstavujú osoby a hrany predstavujú vzťahy medzi nimi. Štatistickým porovnaním tejto umelej štruktúry a štruktúry skutočnej sociálnej siete bola potvrdená schopnosť modelu úspešne takúto skutočnú sociálnu sieť napodobniť. Skôr ako výsledné dáta je však dôležitejší postup akým tieto dáta vznikli. Ten je možné v budúcnosti jednoducho modifikovať a tak simulovať rôzne javy vyskytujúce sa v skutočných sociálnych sieťach.

Cieľ práce sa podarilo naplniť. Vznikol základ pre simulovanie ľudských interakcií, ktorý ponecháva množstvo priestoru na ďalšie rozšírenia a využitia. Z mnohých uvediem ako príklad rozšírenie modelu o nové interakcie, pridanie pokročilejšieho grafického užívateľského rozhrania, alebo použitie modelu na skúmanie rôznych javov bez nutnosti angažovať skupinu skutočných ľudí.

Literatura

- [1] JADE White Paper [online].
<http://jade.tilab.com/papers/2003/WhitePaperJADEEXP.pdf>, 2003 [cit. 2013-5-5].
- [2] Global Clustering Coefficient [online]. <http://reference.wolfram.com/mathematica/ref/GlobalClusteringCoefficient.html>, 2012 [cit. 2013-4-27].
- [3] AgentSpeak [online]. <http://en.wikipedia.org/wiki/AgentSpeak>, 2013-2-23 [cit. 2013-4-18].
- [4] Social network [online]. http://en.wikipedia.org/wiki/Social_network, 2013-4-18 [cit. 2013-4-27].
- [5] Belief-desire-intention model [online].
http://en.wikipedia.org/wiki/Belief-desire-intention_model, 2013-4-20 [cit. 2013-4-18].
- [6] Social network analysis.
http://en.wikipedia.org/wiki/Social_network_analysis, 2013-4-29 [cit. 2013-4-27].
- [7] FIPA [online]. <http://en.wikipedia.org/wiki/FIPA>, 2013-4-5 [cit. 2013-5-5].
- [8] Bergenti, F.; Franchi, E.; Poggi, A.: Agent-based Social Networks for Enterprise Collaboration. In *20th IEEE International Workshops on Enabling Technologies: Infrastructures for Collaborative Enterprises, WETICE 2011, Paris, France, 27-29 June 2011, Proceedings*, IEEE Computer Society, 2011, s. 25–28, doi:<http://dx.doi.org/10.1109/WETICE.2011.12>.
- [9] Bonabeau, E.: Agent-based modeling: Methods and techniques for simulating human systems. *Proceedings of the National Academy of Sciences of the United States of America*, ročník 99, č. Suppl 3, 2002: s. 7280–7287, doi:[10.1073/pnas.082080899](https://doi.org/10.1073/pnas.082080899).
URL <http://www.pnas.org/content/99/suppl.3/7280.short>
- [10] Bordini, R. H.; Hübner, J. F.; Wooldridge, M.: *Programming multi-agent systems in AgentSpeak using Jason*. Wiley, c2007.
- [11] Franchi, E.: A Multi-Agent Implementation of Social Networks. *Proceedings of the 11th WOA 2010 Workshop*, , č. 621, 2010.
URL <http://ceur-ws.org/Vol-621/paper09.pdf>
- [12] Franchi, E.; Poggi, A.: Multi-Agent Systems and Social Networks. In *Handbook of research on business social networking*, Business Science Reference, c2012, s. 84–97.

- [13] Hamill, L.; Gilbert, N.: Social Circles: A Simple Structure for Agent-Based Social Network Models. *J. Artificial Societies and Social Simulation*, ročník 12, č. 2, 2009.
- [14] Machulka, T.: *Metody analýzy sociální sítě pro dolování znalostí*. Bakalářská práce, FIT VUT v Brně, Brno, 2011.
- [15] O'Hare, G.; Jennings, N.: *Foundations of Distributed Artificial Intelligence*. A Wiley-Interscience publication, Wiley, 1996, ISBN 9780471006756.
- [16] Toušek, L.: Problematika vytváření relačních dat. *AntropoWEBZIN*, ročník 2-3/2009, 2009: s. 35–41, ISSN 1801-8807.
URL
http://antropologie.zcu.cz/media/webzin/WEBZIN_2-3_2009/Tousek__L..pdf
- [17] Wasserman, S.; Faust, K.: *Social Network Analysis: Methods and Applications*. Structural Analysis in the Social Sciences, Cambridge University Press, 1994, ISBN 9780521387071.
- [18] Wooldridge, M. J.: *An Introduction to Multiagent Systems*. Chichester, UK: Wiley, druhé vydání, 2009, ISBN 978-0-470-51946-2.
- [19] Zbořil, F.: *Plánování a komunikace v multiagentních systémech*. Disertační práce, FIT VUT v Brně, Brno, 2004.
- [20] Švec, T.: *Vyhodnocování důvěry nebo reputace na základě analýzy sociální sítě*. Bakalářská práce, FIT VUT v Brně, Brno, 2012.

Příloha A

Obsah CD

- Zdrojové kódy implementovaného programu
- Návod na konfiguráciu a spustenie programu
- Dáta vytvorené modelom použité v práci pre vytvorenie obrázkov
- Súbor s programom pre Wolfram Mathematica 9 pre vytvorenie zobrazenia s koláčovými grafmi
- Archív so spustiteľným programom Jason