

Západočeská univerzita v Plzni  
Fakulta aplikovaných věd  
Katedra informatiky a výpočetní techniky

## **Bakalářská práce**

# **Modelování ovlivnění důvěry v sociální síti**

# **Prohlášení**

Prohlašuji, že jsem bakalářskou práci vypracoval samostatně a výhradně s použitím citovaných pramenů.

V Plzni dne 10.2.2013

.....  
Robert Adamec

## **Poděkování**

Tímto bych rád poděkoval paní Ing. Arnoštce Netrvalové, Ph.D za neskutečnou trpělivost a ochotu pomoci při vytváření této práce a také za poskytnutí podkladů a materiálů, které mi byly při realizaci práce velmi nápomocné.

# **Abstract**

## **Modeling of trust influencing in the social network**

In today's digital age, trust and its influence on humans are very important. On its basis, we buy and sell things on the internet, we share our personal information and thanks to social networks even photos from our vacations. Our decisions are highly influenced by trust and that is also the reason why to study it and try to model its behaviour.

The thesis focuses on the problem of trust and trust modeling in social network environment. The definitions of trust, social network and essential terms are introduced. Then the thesis describes the concept of trust modeling. The model is programmed in Java and it is applied on the experiments added at the end of the thesis.

# Obsah

<b>1 Úvod</b>	<b>1</b>
<b>2 Důvěra</b>	<b>2</b>
2.1 Definice důvěry . . . . .	2
2.2 Repräsentace důvěry . . . . .	3
2.2.1 Repräsentace důvěry mezi dvěma subjekty . . . . .	4
2.3 Utváření důvěry . . . . .	5
<b>3 Sociální síť</b>	<b>7</b>
3.1 Definice sociální sítě . . . . .	7
3.1.1 Uzly sociální sítě . . . . .	8
3.1.2 Utváření sociální sítě . . . . .	8
3.1.3 Relace v sociální síti . . . . .	9
3.2 Vlastnosti sociální sítě . . . . .	10
3.2.1 Vlastnosti celé sociální sítě . . . . .	10
3.2.2 Vlastnosti uzlů . . . . .	10
3.2.3 Vlastnosti relací . . . . .	11
3.3 Repräsentace sociálních sítí . . . . .	11
<b>4 Modelování důvěry v sociální síti</b>	<b>14</b>
4.1 Přístupy ke tvorbě modelů . . . . .	14
4.2 Model důvěry v sociální síti . . . . .	15
<b>5 Softwarová realizace</b>	<b>19</b>
5.1 Analýza programu . . . . .	19
5.2 Architektura programu . . . . .	19
5.3 Implementace programu . . . . .	21
5.3.1 Třída Actor . . . . .	21
5.3.2 Třída Output . . . . .	22
5.3.3 Třída ReadData . . . . .	22
5.3.4 Třída Application . . . . .	24
5.3.5 Třída ChangeValues . . . . .	24
5.3.6 Třída Localization . . . . .	24
5.3.7 Třída Log . . . . .	25
5.3.8 Třída Window . . . . .	25
5.3.9 Třída Model . . . . .	27
<b>6 Experimenty s modelem</b>	<b>29</b>
6.1 První experiment . . . . .	29
6.1.1 Zadaná data . . . . .	29
6.1.2 Výsledky experimentu . . . . .	30
6.2 Druhý experiment . . . . .	33
6.2.1 Zadaná data . . . . .	33
6.2.2 Výsledky experimentu . . . . .	35
6.3 Třetí experiment . . . . .	38
6.3.1 Zadaná data . . . . .	38
6.3.2 Výsledky experimentu . . . . .	39
<b>7 Závěr</b>	<b>42</b>

## Rejstřík obrázků

Obrázek 2.2.1 - Vyjádření důvěry spojitým intervalem .....	3
Obrázek 2.2.2 - Reprerentace důvěry fuzzy logikou .....	3
Obrázek 2.2.3 - Čtverec důvěry mezi dvěma aktéry .....	4
Obrázek 2.2.4 - Devět variant čtverce důvěry .....	4
Obrázek 3.1.1 - Příklad sociální sítě .....	8
Obrázek 3.3.1 - Matice sousednosti pro neorientovaný a orientovaný graf .....	12
Obrázek 3.3.2 - Matice incidence pro neorientovaný a orientovaný graf .....	12
Obrázek 3.3.3 - Seznam sousednosti pro neorientovaný a orientovaný graf .....	12
Obrázek 4.2.1 - Graf modelu sítě se dvěma aktéry .....	14
Obrázek 4.2.2 - Graf modelu sítě s více aktéry .....	15
Obrázek 4.2.3 - Graf modelu sítě s reputacemi .....	16
Obrázek 5.2.1 - Zjednodušený UML diagram tříd .....	19
Obrázek 5.3.1 - Struktura třídy Output .....	20
Obrázek 5.3.2 - Ukázka formátu vstupního souboru .....	21
Obrázek 5.3.3 - Struktura třídy ReadData .....	21
Obrázek 5.3.4 - Struktura třídy Actor .....	22
Obrázek 5.3.5 - Struktura třídy ChangeValues .....	23
Obrázek 5.3.6 - Struktura třídy Localization .....	24
Obrázek 5.3.7 - Struktura třídy Log .....	24
Obrázek 5.3.8 - Struktura třídy Window .....	25
Obrázek 5.3.9 - Struktura třídy Model .....	26
Obrázek 6.1.1 - Matice sousednosti sítě prvního experimentu .....	27
Obrázek 6.1.2 - Grafová reprezentace sítě prvního experimentu .....	27
Obrázek 6.1.3 - Matice důvěry sítě prvního experimentu .....	28
Obrázek 6.2.1 - Matice sousednosti sítě druhého experimentu .....	31
Obrázek 6.2.2 - Grafová reprezentace sítě druhého experimentu .....	31
Obrázek 6.2.3 - Matice důvěry sítě druhého experimentu .....	32
Obrázek 6.3.1 - Matice sousednosti sítě třetího experimentu .....	35
Obrázek 6.3.2 - Grafová reprezentace sítě třetího experimentu .....	35
Obrázek 6.3.3 - Matice důvěry sítě třetího experimentu .....	36
Obrázek P1 - Grafické rozhraní aplikace v češtině .....	41
Obrázek P2 - Grafické rozhraní aplikace v angličtině .....	41

## Rejstřík tabulek

Tabulka 1 - Velikost koeficientu $\lambda$ v závislosti na vzdálenosti .....	27
Tabulka 2 - Hodnoty váhových koeficientů pro první experiment .....	28
Tabulka 3 - Dispozice k důvěře jednotlivých aktérů pro první experiment .....	28
Tabulka 4 - Hodnoty váhových koeficientů pro druhý experiment .....	32
Tabulka 5 - Dispozice k důvěře jednotlivých aktérů pro druhý experiment .....	32
Tabulka 6 - Hodnoty váhových koeficientů pro třetí experiment .....	35
Tabulka 7 - Dispozice k důvěře jednotlivých aktérů pro třetí experiment .....	36

## Rejstřík grafů

Graf 1 - Reputace udělená aktéru B .....	29
Graf 2 - Vývoj důvěry k aktéru B .....	29
Graf 3 - Vývoj důvěry mezi všemi aktéry v iteracích .....	30
Graf 4 - Reputace udělená aktéru K .....	33
Graf 5 - Vývoj důvěry k aktéru K .....	33
Graf 6 - Vývoj důvěry k aktéru L .....	34
Graf 7 - Vývoj důvěry mezi aktéry A a I .....	34
Graf 8 - Reputace udělená aktéru G .....	37
Graf 9 - Vývoj důvěry k aktéru G .....	37
Graf 10 - Vývoj důvěry mezi aktéry B a C .....	38



# 1 Úvod

Mezilidské vztahy, pro někoho noční můra, pro jiného denní chléb. Posun v této oblasti je v posledních dekáдах obrovský, člověk už nepotřebuje osobní kontakt pro seznámení se s novými lidmi, stačí mu několik kliknutí a napsaných slov. Právě sociální sítě jsou tím prostředkem, který umožňuje, aby se seznámily osoby vzdálené přes půlku planety a mohly sdílet své zážitky a názory. Tyto internetové aplikace umožňují neustálou komunikaci mezi miliony lidí na celé planetě a jsou to právě ony, které vládou dnešním trendům v mezilidské komunikaci. Stálým a největším zástupcem je Facebook, ale oblíbenými jsou také MySpace či Twitter a stále populárnější je i nejnovější přírůstek do rodiny internetových sociálních sítí, Google+.

I v dnešní uspěchané digitalizované době se však mezilidské vztahy utvářejí a zanikají v průběhu minut i let a podstatnou proměnnou ovlivňující trvání a sílu těchto vztahů je důvěra. Jak těžko se nabývá, tak lehce je možné ji ztratit a její vliv zasahuje nejen do běžných mezilidských vztahů, ale ovlivňuje téměř všechny aspekty života, od pocitu bezpečí díky ozbrojené strážci až po uzavírání obchodních zakázek. Právě na základě důvěry si vybíráme, komu sdělíme jaké informace či zda je rozumné dané osobě půjčit peníze.

Tato práce se pokouší tyto křehké vztahy modelovat a umožnit tak zkoumání jednotlivých aspektů, které důvěru v sociální síti ovlivňují. První část se věnuje podrobnému popisu důvěry, aspektů, které ji ovlivňují a možnostem její reprezentace v informatice. Neopomeneme si popsat ani pojem sociální sítě a způsoby, jak se důvěra uvnitř takovéto sítě utváří a mění. Nakonec se zaměříme na možnosti modelování sociálních sítí.

V druhé části už se věnujeme samotné implementaci konkrétního modelu v jazyce Java, popíšeme si strukturu a jednotlivé části vytvořeného programu a zhodnotíme výsledky provedených experimentů.

Poslední část pak obsahuje přílohy a uživatelský manuál.

## 2 Důvěra

Neurčitý pojem, na který má každý člověk trochu jiný pohled a který nelze jednoznačně definovat. Přesto toto slovo dokáže bořit překážky a předem rozhodnout o osudu událostí či schůzek. Důvěra protkává náš každodenní život téměř ve všech aspektech, důvěřujeme rodině a přátelům, obchodníkům, že nás nešidí, zaměstnavateli, že nás nepropustí a v neposlední řadě také důvěřujeme politikům, že budou hájit naše zájmy na politické půdě. Důvěra formuje náš první dojem a pokud náš přítel důvěřuje nějaké osobě, jsme ochotni jí důvěřovat také.

Sociologie pak není jediným oborem, ve kterém se důvěra projevuje, její vliv a výskyt pozorujeme téměř ve všech oborech a disciplínách, příkladem budiž fyzikální či matematické teorie, kde důvěřujeme vědcům, že jejich výpočty jsou správné a tvrzení pravdivá a z nichž pak například vychází naše vlastní teorie.

Vzhledem k tomu, jak široké spektrum důvěra ovlivňuje a jak se může její význam a váha měnit v závislosti na subjektivních pocitech a zásadách každého z nás, není k podivu, že neexistuje ucelená definice co to vlastně důvěra je.

### 2.1 Definice důvěry

Jak již bylo řečeno, je důvěra těžko definovatelný pojem, který může mít zcela odlišný význam v různých odvětvích. Vyhledáním tohoto slova ve slovníku [1] dostaneme následující výsledek:

- Pevná víra ve schopnosti, poctivost a charakter osoby
- Pevná víra ve spolehlivost a funkčnost věci
- Péče, starost
- Naděje v budoucí události
- Spolehnutí se na schopnost zákazníka zaplatit pohledávky
- Čest, právo, spolehlivost

Vědeckých pokusů o přesné vymezení tohoto pojmu existuje nespočet. Pro ukázkou takové definice se podíváme na práci D. M. Rousseau [2], jejíž definice zní:

*„Důvěra je psychologickým stavem zahrnujícím záměr přijmout zranitelnost přicházející spolu s kladným očekáváním záměrů a chování druhé osoby.“*

Podobně důvěru definoval i R. J. Lewicki se svým týmem [3], kteří na důvěru pohlíželi následovně:

*„Důvěra je víra jedince ve slova, činy a rozhodnutí druhé osoby a ochota na jejich základě jednat.“*

Jak vidíme, je důvěra spjata s vírou v chování jiné osoby a nesení následků za tuto víru. Podobných definic nalezneme nepřeborné množství, méně či více odborných, avšak mezi ty nejužívanější definice patří především ta od Diega Gambetty, kterou formuloval na sympoziu o důvěře v Cambridge v roce 1988:

*„Důvěra (či analogicky nedůvěra) je konkrétní úroveň subjektivní pravděpodobnosti, se kterou agent odhadne vykonání určité akce jiného agenta či skupiny agentů ještě předtím, než takovou akci zpozoruje (nebo nezávisle na tom, je-li takového pozorování vůbec schopen) a v kontextu s tím provede svou vlastní akci“.*

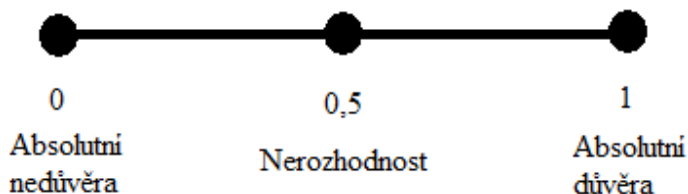
Ze všech citovaných definic vidíme, že několik společných prvků se objevuje ve všech snahách o definici důvěry. Mezi tyto prvky patří:

- Subjektivita
- Časová závislost
- Riziko
- Nejistý výsledek
- Nepředvídatelnost
- Situační závislost

## 2.2 Reprezentace důvěry

Míra důvěry (dále pouze důvěra) se dá vyjádřit více způsoby, z nichž nejčastěji je používáno vyjádření spojitým intervalem, který má pevně dané meze. Výjimečně se používá i reprezentace diskretními veličinami, avšak ty jsou nepřesné.

Meze intervalu jsou nejčastěji od 0 do 1, kde 0 znamená absolutní nedůvěru a 1 absolutní (slepu) důvěru. Většinou se uvádí i střední hodnota, nazývaná nerozhodný stav. Tento způsob reprezentace lze snadno pochopit z Obr. 2.2.1.

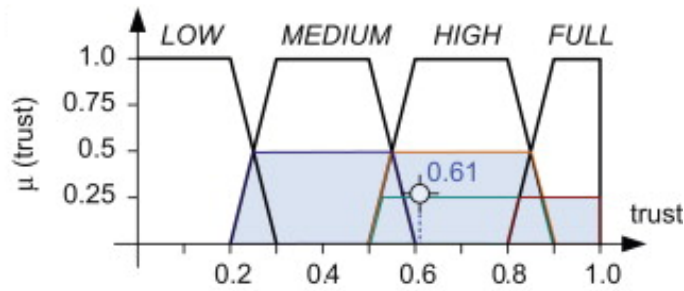


Obrázek 2.2.1 - Vyjádření důvěry spojitým intervalem

Někdy však není tento způsob dostatečně podrobný a využívá se jeho rozšíření, buď zvýšením počtu stupňů nebo zvětšením daného intervalu.

Důvěra je však pojem velmi subjektivní a tak může docházet k tomu, že hodnotu, kterou jeden subjekt považuje za nedůvěru, bude druhý subjekt považovat za nerozhodnost a proto se využívá fuzzy logiky [4].

Z Obr. 2.2.2 pak vidíme, že intervaly se nám překrývají a tím dochází k různé interpretaci hodnot.

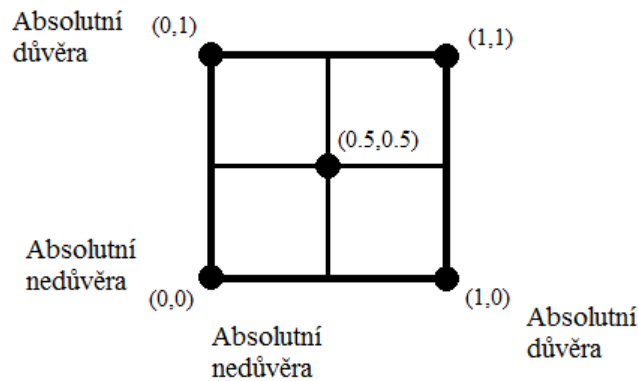


Obrázek 2.2.2 - Reprezentace důvěry fuzzy logikou, převzato z [5]

### 2.2.1 Reprezentace důvěry mezi dvěma subjekty

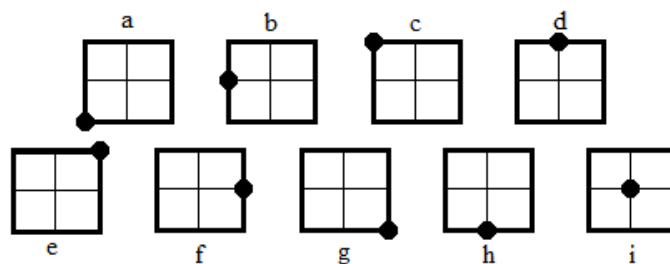
Z definice Diega Gambetty vyplývá, že důvěra je hodnota popisující vztah dvou subjektů, poskytovatele<sup>1</sup> a příjemce<sup>2</sup>. Mezi těmito subjekty se vytváří vztah dobře popsitelný dvojrozměrným souřadným systémem, tzv. čtverci důvěry [6].

Velikost důvěry se nachází v intervalu (0, 1) a hodnoty důvěry si označíme jako  $T_L$  a  $T_P$ , kde  $T_L$  popisuje důvěru levého subjektu k pravému a  $T_P$  naopak, viz. Obr. 2.2.3.



Obrázek 2.2.3 - Čtverec důvěry mezi dvěma aktéry

Důvěru pak v tomto čtvercovém grafu nalezneme jako průsečík čar vynesných z obou subjektů. Díky čtvercové realizaci má průsečík devět možných variant, jak vidíme na Obr. 2.2.4.



Obrázek 2.2.4 - Devět variant čtverce důvěry

<sup>1</sup>přeloženo z anglického výrazu Trustor

<sup>2</sup>přeloženo z anglického výrazu Trustee

Slovní popis variant:

- a) Pár se vzájemnou absolutní nedůvěrou
- b) Pravý subjekt je nerozhodný k levému a levý subjekt cítí k pravému absolutní nedůvěru
- c) Pravý subjekt cítí k levému absolutní důvěru a levý subjekt cítí k pravému absolutní nedůvěru
- d) Pravý subjekt je nerozhodný k levému a levý subjekt cítí k pravému absolutní důvěru
- e) Pár se vzájemnou absolutní důvěrou
- f) Levý subjekt je nerozhodný k pravému a pravý subjekt cítí k levému absolutní důvěru
- g) Pravý subjekt cítí k levému absolutní nedůvěru a levý subjekt cítí k pravému absolutní důvěru
- h) Levý subjekt je nerozhodný k pravému a pravý subjekt cítí k levému absolutní nedůvěru
- i) Oba subjekty jsou v nerozhodném stavu nebo mezi nimi neexistuje žádný vztah

### **2.3 Utváření důvěry**

Jak již bylo řečeno, důvěra je pojem velmi subjektivní a stejně tak i její vznik. Každý člověk vnímá dění kolem sebe jinak a co pro jednoho znamená neštěstí, druhého nijak nevyvede z míry. Tato skutečnost se promítá i do utváření důvěry, zatímco někomu trvá dlouhou dobu, než začne důvěřovat a sebemenší zaváhání důvěřované osoby rapidně sníží danou důvěru, jinému stačí několik schůzek či významných událostí a dokáže s jistotou říct, komu může důvěřovat.

Z výše popsaného tedy vyplývá, že úbytek a přírůstek důvěry jsou přímo závislé na událostech, které se kolem důvěřované osoby dějí. Mezi základní faktory podílející se na utváření důvěry patří:

- Zkušenost
- Důvěryhodnost
- Reputace
- Kontext
- Doporučení
- Riziko
- Evoluce důvěry

Zkušenost je většinou jeden z nejpodstatnějších faktorů. Patří mezi přímé faktory, kdy jedinec na základě předchozích kontaktu se subjektem, ke kterému se důvěra utváří, hodnotí jeho chování a užitečnost rad při řešení problémů. Zároveň sem také patří možné výměny názorů či konfrontace.

Důvěryhodnost patří také mezi velmi podstatné faktory, je přímo závislá na počtu uskutečněných setkání daných subjektů. Se vzrůstajícím počtem schůzek máme více možností a událostí, na základě kterých si můžeme na daného jedince vytvořit názor a ujasnit si, jak moc mu lze důvěřovat.

Reputace reflektuje oblíbenost daného subjektu mezi určitou skupinou lidí. Nevztahuje se pouze k lidem, za reputaci můžeme považovat jakékoliv hodnocení produktů, například na srovnávacích internetových portálech. Je to faktor měnící se na základě šířených informací a tak může být jeho vlivu i zneužito.

Ke změnám důvěry dochází při určitých událostech a za určitých okolností a právě tyto vnější vlivy působící na to, k jaké a jak velké změně důvěry dojde, jsou shrnuty pojmem kontext.

V sítích větších, než jsou pouhé dva prvky, se vyskytuje faktor doporučení. Jedná se o názor třetí osoby na důvěřovaný subjekt a je posuzován na základě předchozích faktorů, jelikož právě reputace a důvěryhodnost hrají roli v tom, jakou váhu tomuto názoru jedinec přisoudí.

Společně s těmito faktory však přichází také riziko. Jedná se o míru kontroly a vlivu druhé osoby na události kolem jedince a případný kladný či záporný dopad jeho činů.

Důsledkem všech těchto faktorů tak dochází ke změnám důvěry, které se popisují jako tzv. evoluce důvěry.

Na základě výše definovaných faktorů je postavený model důvěry v sociální síti, který jich využívá v původním nebo pozměněném znění.

## 3 Sociální síť

Sociální sítě jsou fenoménem dnešní doby. Málokdo je nevyužívá a ještě méně lidí nezná význam tohoto pojmu. Hlavní zásluhu na tom samozřejmě má informatika a sociální sítě na internetu. Největším zástupcem je Facebook, sdružující milióny lidí ze všech koutů světa do jedné komunity, ve které dochází denně k obrovským transakcím dat a informací. Facebook není jediným hráčem na tomto poli, hodně používané jsou také Twitter a MySpace a stále větší oblíbenosti se těší Google+.

Nejsou to však pouze tito zástupci, sloužící pro výměnu osobních dat a komunikaci, kteří se dají v informatice spojit s pojmem sociální sítě. Za sociální síť můžeme považovat i obchodní portály Ebay či Aukro. I zde hraje komunikace mezi jednotlivými účastníky velkou roli a existuje systém hodnocení jednotlivých obchodníků, dle kterého lze odhadnout spolehlivost daného obchodníka.

Stále se držíme pouze oblasti informatiky, avšak sociální sítě zasahují i do jiných oborů. Velké společnosti například shromažďují data o zákaznících a na jejich základě pak tvoří sociální sítě jednotlivých zákazníků, které využívají pro cílený marketing.

Nesmíme zapomínat ani na predikci šíření informací či nemocí, která také hojně využívá sociálních sítí založených na osobních kontaktech jedinců.

Jak vidíme, zasahují sociální sítě do mnoha oblastí, ať už jsme si vědomi jejich existence nebo ne, a proto je důležité věnovat pozornost jejich vzniku a struktuře.

### 3.1 Definice sociální sítě

Sociální síť je pojem přeložený z anglického social network a tento překlad není jediným používaným v českém jazyce, používají se také pojmy společenská a komunitní síť a v neposlední řadě také zkráceně komunita. Výzkumem chování a vzniku těchto sítí se zabývají tzv. vědy o sítích<sup>3</sup>, které pro jejich analýzu používají kombinaci statistiky, teorie grafů a nejnovějších informačních technologií.

V oblasti sociálních sítí se místo pojmu subjekt užívá pojem aktér, ať už jsou těmito aktéry lidé, organizace či stroje, a všechny tyto pojmy tvoří sociální strukturu, která popisuje propojení těchto aktérů. Za sociální síť můžeme označit libovolnou skupinu aktérů, kteří spolu nějakým způsobem komunikují. Tyto sítě vznikají na základě společného prvku, ať už je jím práce, studium, zájmy či například rodinné vztahy.

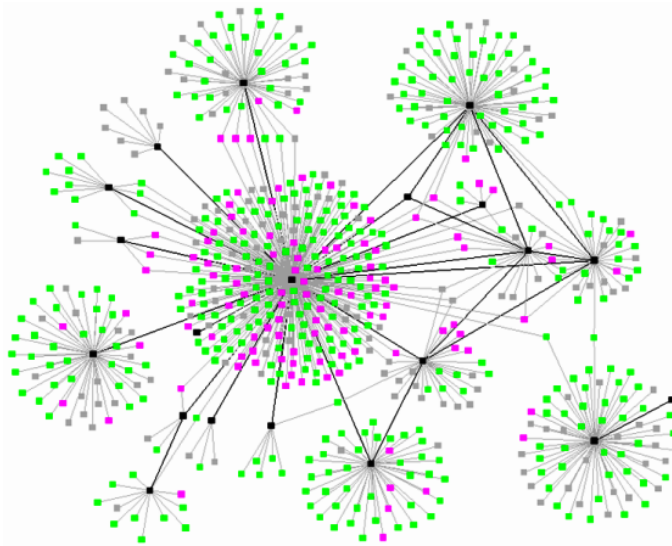
Z výše uvedeného tedy vyplývá, že sociální síť je struktura tvořená aktéry a relacemi mezi nimi. Množina těchto aktérů je konečná a definuje velikost sociální sítě. Nejjednodušší reprezentací této sítě je pak orientovaný graf bez vícenásobných vazeb mezi dvěma vrcholy.

Při grafickém znázorňování sítí se jednotlivý aktéři sítě častěji označují jako uzly.

---

<sup>3</sup>přeloženo z anglického výrazu Network science

Velmi často se používá barevného odlišení jednotlivých uzlů sítě podle důležitosti. Nejčastější volba barev je zelená či růžová barva pro důležité uzly a šedá pro ty nevýznamné. Příklad takové sítě vidíme na obrázku 3.1.1.



Obrázek 3.1.1 - Příklad sociální sítě, převzato z [7]

### 3.1.1 Uzly sociální sítě

Za uzel můžeme označit každého aktéra, který se v síti vyskytuje a je propojen s jinými uzly relacemi. Nemusí však jít vždy jen o jedince, za uzel někdy označujeme i skupinu aktérů, u kterých je jednodušší je reprezentovat jako jeden uzel. Příkladem budiž sport, kde přestože je tým složen z hráčů, vystupuje celý tým, například v tabulce, jako jeden celek.

Jak už bylo řečeno, existuje mezi jednotlivými aktéry v sociální síti nějaký společný prvek, který je spojuje. Díky tomuto prvku pak můžeme vhodné aktéry pro vytvoření nové sítě snadno vyhledat a určit tak hranice vznikající sítě.

K tomu nám slouží dva přístupy. V prvním případě hranice vznikají přirozeně, například u příbuzenských vztahů či u studentů jedné školy. Druhý případ pak bere v potaz celou populaci a hranice vytváří právě společný prvek. Příkladem takového společného prvku je například průměrná čistá mzda za měsíc.

### 3.1.2 Utváření sociální sítě

V každé společnosti lidí vznikají samovolně komunity, nebo-li sociální sítě, aniž by o to někdo přímo usiloval. Pokud však budeme chtít sociální síť vytvořit záměrně, můžeme využít tří základních metod [8].

První z nich je metoda úplně sítě<sup>4</sup>, kdy každý z uzlů dané sítě popíše své relace k ostatním uzlům sítě. Díky tomu získáme komplexní pohled na všechny relace v dané sociální

<sup>4</sup>přeloženo z anglického výrazu Full network method



síti. Zároveň je to ovšem metoda použitelná pouze pro malé sociální sítě, jelikož sbíráme data od všech uzlů sítě a metoda je tak velmi nákladná a především časově náročná.

Rozumnější přístup používá metoda sněhové koule<sup>5</sup>. Náhodně vybraný uzel je požádán, aby na základě položené otázky vyjmenoval určitý počet uzlů, se kterými je v relaci. Metoda poté pokračuje k těmto vyjmenovaným uzlům a proces se opakuje. Postupně tak v každé iteraci nacházíme nové uzly, v potaz bereme jen ty, které ještě v síti nejsou, a pokračujeme buď tak dlouho, dokud nacházíme nové uzly nebo do předem určeného počtu uzlů v síti. Metoda je ovšem přímo závislá na volbě prvotního uzlu a na počtu iterací a může dojít k tomu, že nenalezneme všechny potenciální uzly sítě.

Třetí metoda, tzv. egocentrická metoda<sup>6</sup>, se používá v případě, že nelze vyhledat všechny vyjmenované sousední uzly. Vybereme sadu uzlů, které požádáme o popis spojení s jiným setem uzlů a zároveň o vyjmenování uzlů daných dvou setů, které jsou propojeny.

### 3.1.3 Relace v sociální síti

Relace popisuje vztah mezi dvěma aktéry, kteří mají nějaký společný prvek. Síla tohoto vztahu se dá vyjádřit více způsoby, od nejjednoduššího popisu po ty podrobnější.

Nejjednodušším typem relace je binární relace. Jak už název napovídá, používá tato relace pouze dvě možné hodnoty a to 0 a 1. Můžeme díky ní tedy rozlišit pouze dva stavy a to, zda mezi aktéry existuje vztah či nikoliv.

Podrobnějším typem relace je relace s výběrem z více možností, která už nám dokáže o vztahu říci více, než pouze zda existuje či nikoliv. Dotazovaní aktéři mají na výběr z určitého počtu možností typu známí, rodina, nejlepší přítel apod. a dle typu vztahu tyto možnosti přiřazují k jednotlivým aktérům. Kódování se používá od 0 nahoru.

Dalším používaným typem relace je skupinově zaměřená relace. Ta rozděluje relace do podskupin, které kóduje -1, 0, 1. Kde -1 značí negativní vztah, 0 neutrální a 1 pozitivní. Vzhledem k podrobnějšímu popisu vztahu, než jaký poskytuje binární relace, a zároveň stále jednoduché implementaci, patří tento typ relace mezi velmi používané.

Stejně jako relace s výběrem z více možností patří relace s úplným pořadím mezi ty podrobnější. Aktéři jsou požádáni, aby seřadili jednotlivé aktéry, se kterými jsou v relaci, podle oblíbenosti od 1 do N. Každé z těchto pozic potom přiřadíme určitou váhu, která určuje sílu vazby.

Nejpodrobnějším typem relací jsou relace intervalové. Jak vyplývá z názvu, používají se pro hodnocení váhy a síly vztahu reálná čísla z pevně daného intervalu, nejčastěji od 0 do 1. To nám dává velké množství možných hodnot, kterých vztah mezi dvěma aktéry může dosahovat, a můžeme tak přesněji určit jeho sílu.

---

<sup>5</sup>přeloženo z anglického výrazu Snowball method

<sup>6</sup>přeloženo z anglického výrazu Ego-centric method

## 3.2 Vlastnosti sociální sítě

Než se dostaneme k vlastnostem samotných sítí, uzlů a relací je vhodné si definovat základní dělení sítí [8].

První možností dělení sítí je na orientovanou a neorientovanou síť. U orientované sítě jsou hrany tvořeny uspořádanými dvojicemi uzlů. Síť neorientovaná je tvořena množinami uzlů.

Druhé dělení je na spojitou, nespojitou a egocentrickou síť. Za spojitou označíme síť, kde lze projít od jednoho uzlu k jinému přes libovolný počet hran. V případě neexistence této cesty se jedná o tzv. nespojitou síť, ve které může docházet k existenci izolovaných uzlů. Pokud v síti existuje centrální uzel, ke kterému vedou cesty od všech ostatních uzlů, pak tuto síť nazveme egocentrickou. Podrobnější popis nalezneme v článku [9].

### 3.2.1 Vlastnosti celé sociální sítě

První důležitou vlastností, která je vcelku zřejmá, je velikost. Je dána počtem hran mezi jednotlivými uzly a přímo závisí na počtu uzlů v síti. Označme si počet uzlů jako  $n$ , potom existuje maximálně  $n * (n-1)$  hran. Navíc si ještě musíme uvědomit, že v neorientované síti se počet hran ještě půlí, jelikož hrana z prvního do druhého uzlu je totožná s hranou z druhého do prvního uzlu. Velikost sítě nám zároveň ovlivňuje, s jakou rychlostí a spolehlivostí se bude v dané síti přenášet zpráva mezi uzly.

Druhou často uváděnou vlastností je hustota. Ta ukazuje, jak jsou jednotlivé uzly propojeny a proto je přímo závislá na počtu hran mezi nimi. Určitě si dokážeme představit rozdíl mezi sítí o deseti aktérech, kde se budou všichni aktéři s velkou pravděpodobností vzájemně znát a sítí o sto aktérech, kde bude existovat převážné množství aktérů, kteří nebudou znát všechny ostatní aktéry.

### 3.2.2 Vlastnosti uzlů

Jednou z důležitých vlastností uváděných u uzlů je jejich stupeň. Ten vyjadřuje počet hran, které do vrcholu vstupují a zároveň může vyjadřovat i počet hran, které z něj vystupují. Pro orientované grafy totiž existuje vstupní a výstupní stupeň. Pro neorientované pak existuje pouze ten vstupní.

Tyto sousední uzly dohromady tvoří tzv. sousedství uzlu. Toto sousedství je vlastně tím, co tvoří samotný stupeň uzlu.

Dále se uvádí dosažitelnost uzlu, která vyjadřuje, zda existuje cesta mezi dvěma uzly. Pokud ano, pak je její délka rovna počtu uzlů, přes které procházíme na této cestě. Pokud cesta neexistuje, pak jsou tyto dva uzly označeny jako nedosažitelné.

Délka této cesty se označuje jako vzdálenost mezi uzly. V neorientovaných grafech pak platí, že existuje-li cesta z uzlu do uzlu jiného, existuje i cesta opačná. Pro orientované grafy už tato skutečnost platná není a vyskytnout se může i jednocestné spojení.

### 3.2.3 Vlastnosti relací

Podstatnou vlastností, která charakterizuje relace, je reciprocita, která udává vyváženost vazby. Dle [8] se aktéři v síti dělí na příjemce<sup>7</sup> a zdroje<sup>8</sup>. Kdy, jak už z názvů vyplývá, je za příjemce označen ten, kdo zprávy přijímá a za zdroj ten, kdo je vysílá. Ve většině sítí pak aktéři fungují zároveň jako příjemci i zdroje a je to právě reciprocita, která určuje vyváženost těchto činností. Aktér, který je zároveň příjemcem i zdrojem je označován jako vysílač<sup>9</sup>.

Druhou vlastností pak je tranzitivita, která je využívána při tvorbě doporučení mezi aktéry v síti. Je to vlastnost vyskytující se, pokud aktéři nejsou přímými sousedy, to znamená, pokud v grafu například existuje cesta z A do C přes B, a říká nám, že pokud zná aktér A aktéra B a aktér B aktéra C, pak může znát i aktér A aktéra C.

## 3.3 Reprezentace sociálních sítí

Sociální sítě se velmi často reprezentují grafy. Tyto grafy mají více možných implementací, jmenovitě to jsou matice sousednosti, matice incidence a seznamy sousednosti. Implementace těmito způsoby je ve výsledku totožná, avšak každý z nich má své výhody a nevýhody.

Snadným způsobem vyobrazení sociální sítě je zakreslení daného grafu z dostupných dat bez složité implementace, avšak to lze pouze pro malé sítě, jelikož se vzrůstající velikostí sítě roste i její nepřehlednost.

Matice sousednosti je čtvercová matice, jejíž řádky a sloupce odpovídají jednotlivým uzlům sítě. Proto je rozměr této matice roven počtu uzlů v síti. Při přístupu do matice pak řádkový index značí uzel, ze kterého hrana vychází a sloupcový značí ten, do něhož hrana vede. Co se kódování týče, značí 1 existenci hrany a 0 naopak její neexistenci. Jelikož řádky i sloupce reprezentují uzly sítě, je matice sousednosti symetrickou maticí, což znamená, že záměnou sloupců za řádky získáme totožnou matici.

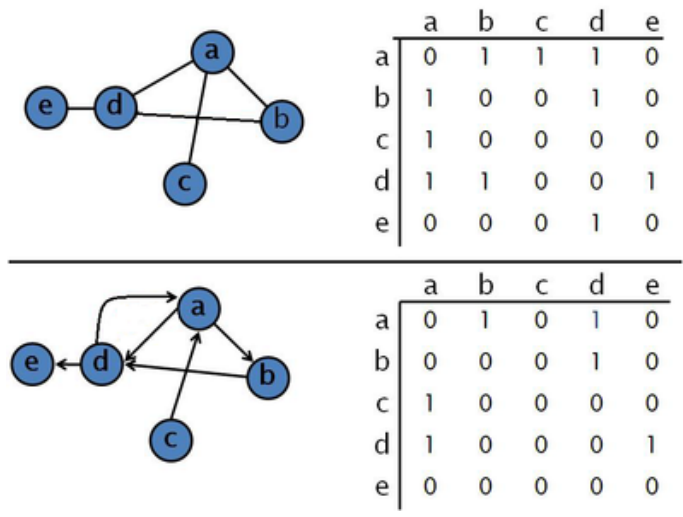
Příklad implementace grafu sítě maticí sousednosti vidíme na Obr. 3.3.1.

---

<sup>7</sup>přeloženo z anglického výrazu Sinks

<sup>8</sup>přeloženo z anglického výrazu Sources

<sup>9</sup>přeloženo z anglického výrazu Transmitter

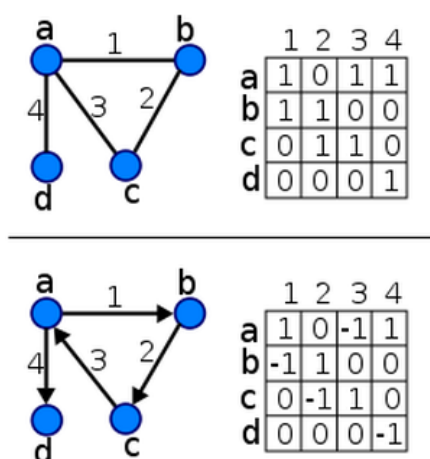


Obrázek 3.3.1 - Matice sousednosti pro neorientovaný a orientovaný graf, převzato z [10]

U matice incidence už neplatí, že by řádky a sloupce značily totéž. Řádky matice odpovídají uzlům sítě, stejně jako u matice sousednosti, avšak sloupce odpovídají hranám sítě. Matice nám tak udává, zda existuje cesta mezi daným uzlem a danou hranou. Kódování existence či analogicky neexistence této cesty je však trochu odlišné pro orientované a neorientované grafy.

Pro neorientované grafy značí 1 existenci cesty a 0 její neexistenci, stejně jako u matice sousednosti, avšak pro grafy orientované je značení jiné. U orientovaných grafů značí 1, že cesta mezi daným uzlem a hranou existuje a vede z daného uzlu do hrany. -1 značí, že cesta mezi daným uzlem a hranou existuje a vede z dané hrany do uzlu. 0 pak standardně značí, že cesta neexistuje.

Lepší obrázek o samotné implementaci si můžeme udělat z Obr. 3.3.2.

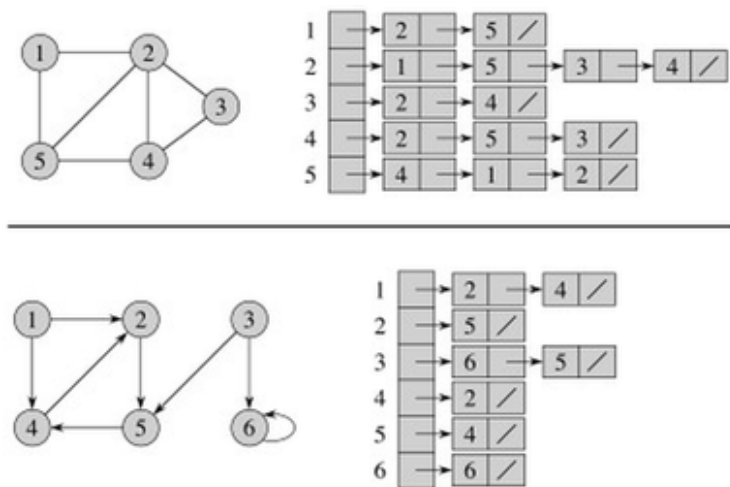


Obrázek 3.3.2 - Matice incidence pro neorientovaný a orientovaný graf, převzato z [11]

Poslední možností, jak implementovat grafy sítí, jsou seznamy sousednosti. Graf reprezentují tak, že pro každý uzel existuje spojový seznam. Pro neorientované grafy jsou v

tomto seznamu všechny sousední uzly, u orientovaných jsou tam pouze ty, do kterých vede z daného uzlu cesta.

Pro jednodušší představu se podívejme na příklad 3.3.3.



Obrázek 3.3.3 - Seznam sousednosti pro neorientovaný a orientovaný graf, převzato z [12]

## 4 Modelování důvěry v sociální síti

### 4.1 Přístupy ke tvorbě modelů

Pro modelování důvěry se využívá velké množství přístupů, každý představující jiný pohled na důvěru. Jsou to právě tyto pohledy, přesněji řečeno teorie, z nichž vychází modelování důvěry, na které se nyní podíváme blíže.

Prvním zástupcem těchto teorií je teorie her. Ta pohlíží na modelování důvěry jako na snahu získat co nejlepší strategii pro každého z aktérů, kteří se modelování účastní, a zároveň zavádí analýzu konfliktních situací mezi těmito aktéry. Za konfliktní situaci je pak považována každá situace, při níž ostatní aktéři reagují na naše rozhodnutí. Dobrým příkladem aplikace této teorie je problém známý pod názvem Vězňovo dilema, více se o tomto problému dozvíme na [13].

Teorie informace klade důraz na samotné informace přenášené v síti a na tento přenos. Zajímá se o rušení a šifrování tohoto přenosu a pokouší se ho měřit. Vybírá si tento přístup jelikož bere v potaz, že komunikace mezi aktéry je při vytváření důvěry velice důležitým prvkem.

Dalším zástupcem je teorie šíření informace a epidemií, která se inspiroje v samotné přírodě. Sleduje oblast, přes kterou se informace šíří a mapuje aktéry, kteří se v této oblasti vyskytují. Modely založené na této teorii patří mezi ty používanější, zejména ve větších sítích, jelikož jsou jednoduché a relativně málo chybové.

Pokud se na informaci budeme dívat jako na hledání vhodné strategie pro splnění zvoleného společného cíle, pak využijeme teorie grafů či teorie pravděpodobnosti. Jevy, jejichž zkoumáním se zabýváme, totiž nemusí vůbec nastat a jsou závislé na okolnostech a volbě. Zároveň je nejistý i jejich výsledek. Klasickým příkladem je hazard.

Posledním zástupcem je teorie rozhodování. Tu používáme, pokud se rozhodujeme mezi více možnými alternativami na základě kladného či záporného ohodnocení. Samotný proces rozhodování je popsán jako proces, jehož výsledkem je výběr nejlepšího možného řešení. Každé takovéto rozhodování pak produkuje nějakou konečnou akci.

Pokud si tedy shrneme teorie, z nichž vychází většina modelů důvěry, jsou to tyto:

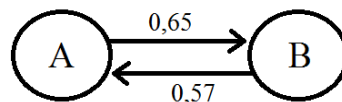
- Teorie her
- Teorie informace
- Teorie šíření informace a epidemií
- Teorie grafů
- Teorie pravděpodobnosti
- Teorie rozhodování

## 4.2 Model důvěry v sociální síti

Modelování probíhá v iteracích, během kterých se mění některé z vlastností aktérů a tím se mění i důvěra mezi sledovanými aktéry. Míru důvěry sledujeme většinou mezi dvěma aktéry, nazveme si je například  $a_i$  a  $a_j$ , a ty pak nazýváme jako poskytovatele důvěry a příjemce důvěry. Důvěra mezi těmito aktéry je hodnota spadající do intervalu  $\langle 0,1 \rangle$ . Samozřejmostí je, že čísla  $i$  a  $j$  nesmí být stejná.

Samotný model je reprezentován jako orientovaný graf s ohodnocenými hranami, jehož uzly představují jednotlivé aktéry a hrany důvěru mezi nimi. Orientace těchto hran pak ukazuje směr, ve kterém se důvěra utváří. Ohodnocením těchto hran je hodnota dané důvěry, což znamená, že důvěra mezi dvěma aktéry bude s největší pravděpodobností v každém směru jiná.

Pro lepší ilustraci se podívejme na Obr. 4.2.1.



Obrázek 4.2.1 - Graf modelu sítě se dvěma aktéry

Hodnota důvěry v iteraci, ve které se právě nacházíme se nazývá aktuální důvěra a její hodnota je dána dle vztahu (1):

$$T_{ij} = F(t_{ij}, t_{ji}, d_{ij}, r_i, g_i) \quad (1)$$

Vysvětlivky proměnných:

$t_{ij}$  značí původní míru důvěry  $i$ -tého aktéra v  $j$ -tého aktéra

$t_{ji}$  značí míru důvěry  $j$ -tého aktéra v  $i$ -tého aktéra

$d_{ij}$  značí doporučení  $j$ -tého aktéra  $i$ -tému od jiných aktérů

$r_i$  značí reputaci  $i$ -tého aktéra v dané sociální síti

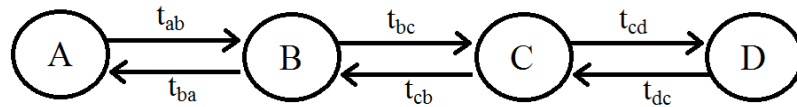
$g_i$  značí dispozici k důvěře podle okolností

Míry důvěry mezi aktéry známe z předchozích iterací, první proměnou, jejíž hodnotu musíme počítat, je tak doporučení. Co je to samotné doporučení jsme si vysvětlovali v předchozí kapitole, ujasněme si tedy ještě jeho význam názornou aplikací.

Jestliže budeme mít aktéry A, B a C, kde aktér B zná jak aktéra A tak C, aktér A zná pouze aktéra B a stejně tak aktér C zná pouze B, pak za doporučení označíme ohodnocení aktéra C aktérem B, které předá aktér B aktéru A. Velikost doporučení je pak dána průměrem míry důvěry mezi aktéry A a B a mezi aktéry B a C.

Jelikož je doporučení dané mírou důvěry mezi aktéry, je zřejmé, že velikost doporučení bude odrážet sílu vztahů mezi danými aktéry. Tento jednoduchý výpočet však platí pouze pro cesty do maximální vzdálenosti 2.

Nyní se podíváme, jak se výpočet změní pro cesty do libovolné vzdálenosti, příklad takové cesty vidíme na obrázku 4.2.2.



Obrázek 4.2.2 - Graf modelu sítě s více aktéry

Pokud se zamyslíme nad podstatou doporučení, je možná i situace, kdy aktér B předá aktéru A doporučení na aktéra D, na kterého dostal doporučení od aktéra C. V takovýchto případech už ale musíme brát v potaz vzdálenost cesty mezi aktéry A a D. Pro tuto potřebu se zavádí koeficient vzdálenosti označený  $\lambda$ . Jeho velikost pro jednotlivé vzdálenosti si můžeme prohlédnout v Tab. 1.

Tabulka 1 - Velikost koeficientu  $\lambda$  v závislosti na vzdálenosti

Délka cesty	Velikost koeficientu $\lambda$
2	1
3	1/2
4	1/3
5	1/4
>5	1/10

Jak vidíme z tabulky, pro vzdálenost větší jak 5 je už koeficient hodně malý, takže doporučení s takto velkou vzdáleností už budou mít na výsledné doporučení malý vliv. Je to z toho důvodu, že se vzrůstající vzdáleností se navyšuje i zkreslení důvěryhodnosti daného cílového aktéra.

Samotný výpočet doporučení pro cesty o  $n$  aktérech je tedy dán vztahem (2).

$$d_{ij} = \frac{\sum_0^n (\lambda \cdot \prod_{i=1}^m t_{i-1i})}{n} \quad (2)$$

Vysvětlivky proměnných:

$t_{i-1i}$  značí důvěru předchozího aktéra v nynějšího

$m$  značí pozici aktéra, který je doporučován

$n$  značí počet aktérů poskytujících doporučení

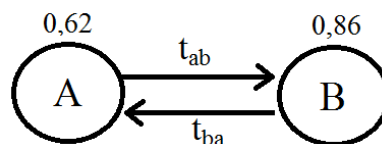
Jak tedy vidíme ze vztahu (2), je doporučení počítáno jako průměr doporučení cílového aktéra od všech aktérů na cestě mezi počátečním a cílovým aktérem. Při výpočtu však musíme zohlednit i změnu doporučení v iteracích, takže upravíme vztah (2) a vznikne nám vztah (3).

$$d_{ij} = \frac{(\sum_0^n (\lambda \cdot \prod_{i=1}^m t_{i-1i})) - d_{ij}^{l-1}}{n} \quad (3)$$



Písmeno  $l$  pak značí iteraci. Jelikož jsme si řekli, že hodnota důvěry se pohybuje v intervalu  $\langle 0,1 \rangle$ , bude se v tomto intervalu pohybovat také doporučení.

Další položkou vztahu (1) je reputace v sociální síti. I o ní jsme si už říkali, že představuje oblíbenost aktéra, ať už se jedná o osobu či věc, mezi ostatními aktéry dané sociální sítě. Kromě vzájemných důvěr mezi jednotlivými aktéry se v síti tedy vyskytuje ještě hodnota reputace, kterou má každý aktér. Příklad vidíme na Obr. 4.2.3.



Obrázek 4.2.3 - Graf modelu sítě s reputacemi

Výpočet je založen na znalosti reputací, které ostatní aktéři udělují danému aktéru a na reputacích těchto aktérů. Hodnotu tedy získáme jako průměr reputací vypočtených jako součin udílené reputace s reputací aktéra, který ji udílí. Pro lepší ilustraci se podívejme na vztah (4).

$$r_i = \frac{\sum_0^n (r_{mi} \cdot r_m)}{n} \quad (4)$$

Vysvětlivky proměnných:

$r_{mi}$  značí reputaci udílenou aktéru  $i$  aktérem  $m$

$r_m$  značí reputaci aktéra  $m$

$n$  značí počet aktérů poskytujících reputace

Stejně tak, jako důvěra a doporučení, spadají i hodnoty reputace do intervalu  $\langle 0,1 \rangle$ .

Poslední nevysvětlenou proměnnou zůstává  $g_i$ . Toto označení nese dispozice k důvěře. Jedná se o velmi proměnlivou hodnotu, jelikož je závislá na okolnostech, aktuální náladě či situaci. Její hodnota opět patří do intervalu  $\langle 0,1 \rangle$ .

Pro výpočet aktuální míry důvěry  $T_{ij}$  použijeme vztah (5), který zní:

$$T_{ij} = t_{ij} + \Delta t_{ij} \quad (5)$$

kde  $\Delta t_{ij}$  značí přírůstek důvěry a vypočteme ho dle vztahu (6):

$$\Delta t_{ij} = \sqrt{t_{ij} \cdot t_{ji}} \cdot d_{ij} \cdot w_{d_i} \cdot r_i \cdot w_{r_i} \cdot g_i \cdot w_{g_i} \quad (6)$$

Vysvětlivky proměnných:

$\sqrt{t_{ij} \cdot t_{ji}}$  značí trend vývoje důvěry mezi aktéry  $i$  a  $j$

$w_{d_i}$  značí váhový koeficient doporučení

$w_{r_i}$  značí váhový koeficient reputace

$w_{g_i}$  značí váhový koeficient dispozice k důvěře

Váhové koeficienty jsou konstantami, jejichž hodnoty se pohybují v intervalu  $\langle 0,1 \rangle$ .

Po dosazení do vztahu (5) získáme vztah (7):

$$T_{ij} = t_{ij} + \sqrt{t_{ij} \cdot t_{ji}} \cdot d_{ij} \cdot w_{d_i} \cdot r_i \cdot w_{r_i} \cdot g_i \cdot w_{g_i} \quad (7)$$

a pokud ještě dosadíme ze vztahů (3) a (4) získáme finální vztah (8) pro výpočet  $T_{ij}$  ve tvaru:

$$T_{ij} = t_{ij} + \sqrt{t_{ij} \cdot t_{ji}} \cdot \frac{(\sum_0^n (\lambda \cdot \prod_{i=0}^m t_{i-1i})) - d_{ij}^{l-1}}{n} \cdot w_{d_i} \cdot \frac{\sum_0^n (r_{mi} \cdot r_m)}{n} \cdot w_{r_i} \cdot g_i \cdot w_{g_i} \quad (8)$$

## 5 Softwarová realizace

Teoreticky už jsme si pojmy a způsob modelování probrali dostatečně, tak se nyní podívejme na samotnou softwarovou realizaci daného modelu. K realizaci výpočtů v modelu jsou využité v předchozí kapitole popsané vztahy. Program vypočítá v každé z iterací aktuální hodnotu důvěry mezi všemi propojenými aktéry v síti.

### 5.1 Analýza programu

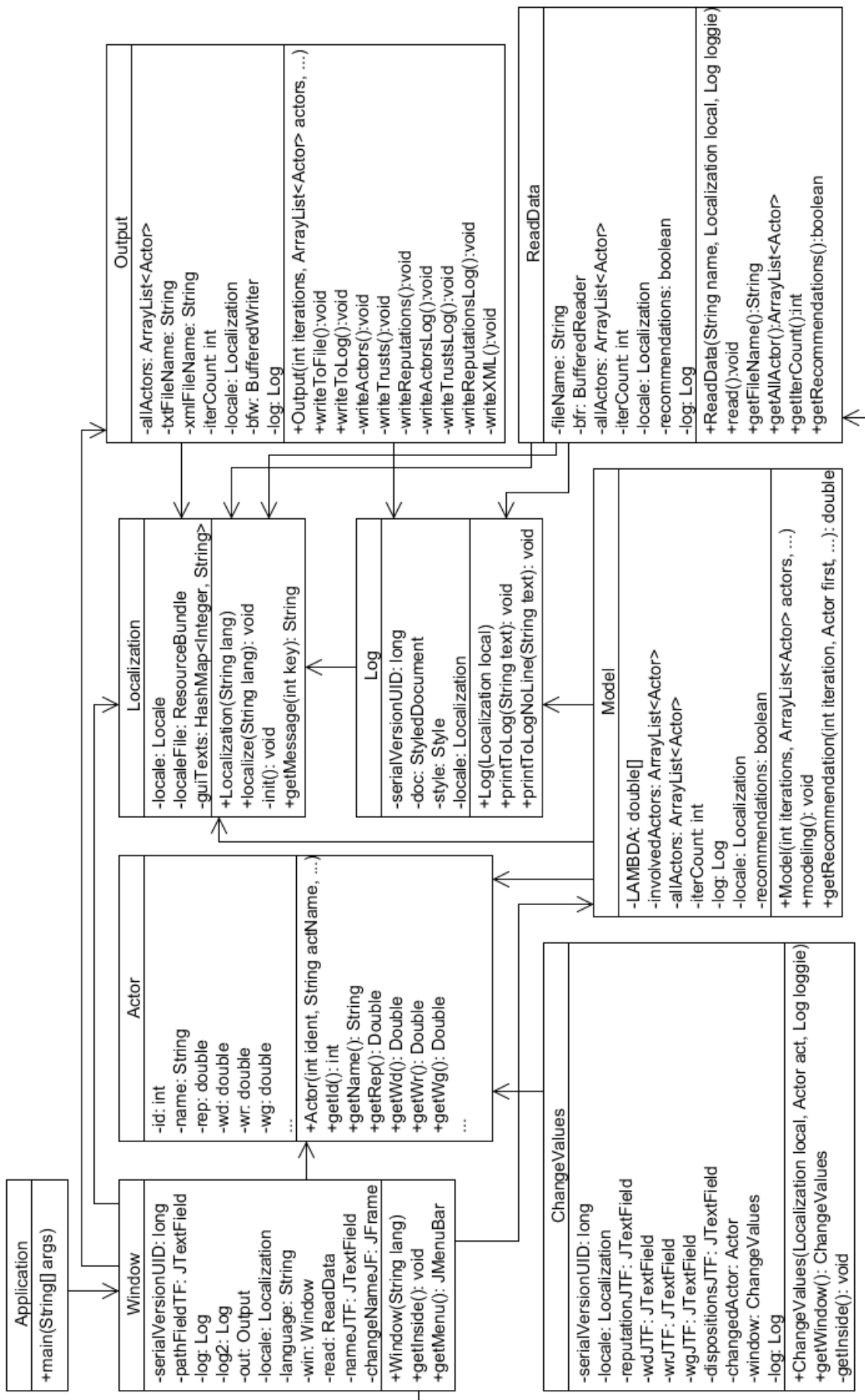
Program je napsán v programovacím jazyce Java, který byl použit i při realizaci předchozího projektu, z něhož tato bakalářská práce vychází. Program sestává z následujících částí:

- Grafické rozhraní pro jednodušší a přehlednou ovladatelnost
- Výpočet aktuální důvěry mezi aktéry v síti v každé iteraci
- Možnost změny načtených dat před spuštěním programu
- Načtení dat z textového souboru
- Zápis dat do textového a XML souboru

### 5.2 Architektura programu

Program je rozdělen do tří balíčků a to `data`, `gui` a `logic`. V balíčku `data` jsou třídy `ReadData` a `Output` starající se o čtení a zápis dat a třída `Actor` reprezentující aktéry v síti. `Gui` obsahuje třídy potřebné pro vytvoření grafického rozhraní aplikace a také třídu `Application`, prostřednictvím které se program spouští. Poslední balíček `logic` pak obsahuje jedinou třídu `Model`, ve které se provádí veškeré výpočty.

Pro lepší pochopení architektury programu a představu o spolupráci jednotlivých tříd je na Obr. 5.2.1 zobrazen zjednodušený UML diagram tříd programu.



Obrázek 5.2.1 - Zjednodušený UML diagram tříd

## 5.3 Implementace programu

### 5.3.1 Třída Actor

Třída Actor reprezentuje jednotlivé aktéry sociální sítě a uchovává o nich potřebná data. Mezi tato data patří unikátní identifikátor, název, reputace, seznam dispozic k důvěře, váhové koeficienty doporučení, dispozice a reputace, seznam reputací, které poskytl, seznam důvěry v ostatní aktéry, seznam doporučení, které poskytl, seznam reprezentující podobu sítě, seznam přímých sousedů a ukazatel do seznamu sousedů.

Seznam přímých sousedů se vytváří při každé změně rozložení sítě v metodě, ve které se dané rozložení sítě nastavuje.

Všechny tyto atributy se dají nastavovat příslušnými `set()` metodami a jelikož jsou soukromé, tak jejich hodnotu získáme přes `get()` metody.

Přesnou strukturu třídy Actor pak vidíme na obrázku 5.3.1.

Actor
<pre>-id: int -name: String -rep: double -wd: double -wr: double -wg: double -reputations: LinkedHashMap&lt;Integer, LinkedHashMap&lt;Actor, Double&gt; &gt; -recommendations: LinkedHashMap&lt;Integer, LinkedHashMap&lt;Actor, Double&gt; &gt; -trusts: LinkedHashMap&lt;Integer, LinkedHashMap&lt;Actor, Double&gt; &gt; -network: LinkedHashMap&lt;Actor, Integer&gt; -dispositions: ArrayList&lt;Double&gt; -neighbours: LinkedList&lt;Actor&gt;&lt;Double&gt; -position: int</pre>
<pre>+Actor(final int ident, final String actName, final double reputation, final double recW, final double repW, final double dispW) +getId(): int +getName(): String +getRep(): Double +getWd(): Double +getWr(): Double +getWg(): Double +getRecommendations(): LinkedHashMap&lt;Integer, LinkedHashMap&lt;Actor, Double&gt; &gt; +getTrusts(): LinkedHashMap&lt;Integer, LinkedHashMap&lt;Actor, Double&gt; &gt; +getReputations(): LinkedHashMap&lt;Integer, LinkedHashMap&lt;Actor, Double&gt; &gt; +getDispositions(): ArrayList&lt;Double&gt; +getNetwork(): LinkedHashMap&lt;Actor, Integer&gt; +getNeighbours(): LinkedList&lt;Actor&gt; +getPosition(): int +setId(final int iden): void +setName(final String actName): void +setRep(final double reputation): void +setWd(final double recW): void +setWr(final double repW): void +setWg(final double dispW): void +setRecommendations(final int iteration, final LinkedHashMap&lt;Actor, Double&gt; recs): void +setTrusts(final int iteration, final LinkedHashMap&lt;Actor, Double&gt; trust): void +setReputations(final int iteration, final LinkedHashMap&lt;Actor, Double&gt; reps): void +setDispositions(final ArrayList&lt;Double&gt; disps): void +setNetwork(final LinkedHashMap&lt;Actor, Integer&gt; net): void +setPosition(int pos): void +incrPosition(): void</pre>

Obrázek 5.3.1 - Struktura třídy Actor

### 5.3.2 Třída Output

Hlavní činností této třídy je zápis výsledků do výstupních souborů ve formátech .txt a .xml a to v té lokalizaci, na kterou je aplikace aktuálně nastavena. K tomuto účelu poskytuje třída veřejnou metodu `writeToFile()`, která volá soukromé metody `writeActors()`, `writeTrusts()`, `writeReputations()` a `writeXML()`.

Metody `writeActors()`, `writeTrusts()` a `writeReputations()` postupně zapíše do výstupního textového souboru údaje o jednotlivých aktérech, podobu sítě a výsledky výpočtů. Pro samotný zápis je používán `BufferedWriter`.

Metoda `writeXML()` zapisuje stejná data jako metody uvedené výše, avšak do XML souboru a tudíž musí dodržovat potřebnou syntax dle pravidel tvoření XML souborů. Názvy výstupních souborů jsou vytvořeny z názvu vstupního souboru a přípony „\_results“.

Třída `Output` zároveň slouží i pro zápis přečtených dat do logu, který se zobrazí po kliknutí na tlačítko pro zobrazení načtených dat. Pro tyto účely používá veřejnou metodu `writeToLog()`, která volá soukromé metody `writeActorsLog()`, `writeTrustsLog()` a `writeReputationsLog()`. Výpis se provádí do instance třídy `Log`.

Přesnou strukturu třídy `Output` vidíme na obrázku 5.3.2.

Output
-allActors: ArrayList<Actor>
-txtFileName: String
-xmlFileName: String
-iterCount: int
-locale: Localization
-bfw: BufferedWriter
-log: Log
+Output(final int iterations, final ArrayList<Actor> actors, final String file, final Localization local)
+Output(final int iterations, final ArrayList<Actor> actors, final Localization local)
+writeToFile():void
+writeToLog():void
-writeActorts():void
-writeTrusts():void
-writeReputations():void
-writeActorsLog():void
-writeTrustsLog():void
-writeReputationsLog():void
-writeXML():void

Obrázek 5.3.2 - Struktura třídy Output

### 5.3.3 Třída ReadData

Obsluhuje načítání dat ze vstupního souboru, jehož název se předává při vytváření instance této třídy. Podstatné atributy třídy jsou atributy popisující název vstupního sou-

boru, seznam aktérů v síti, počet iterací a ukazatel, zda ve výpočtu používáme doporučení. Všechny tyto atributy jsou soukromé a pro získání jejich hodnoty používáme `get()` metody.

Samotné načítání obsluhuje veřejná metoda `read()`, která postupně přečte celý soubor a uloží hodnoty do příslušných proměnných. Pro načítání je používán `BufferedReader`. Pokud není soubor v požadovaném formátu, vypíše program chybovou hlášku do logu.

Ukázku požadovaného formátu vidíme na Obr. 5.3.3, pro účely zmenšení obrázku je použito zarovnání do sloupců, ve skutečnosti je psané vše pod sebe.

```

Actors:           Trusts:           1
A                - 0.450 0.300 0.200 0.160 - 0.8 0.7 0.4 0.5
0.5              0.350 - 0.760 0.130 0.200 0.6 - 0.6 0.6 0.6
0.1 0.4 0.2     0.650 0.520 - 0.480 0.300 0.5 0.6 - 0.5 0.4
                 0.090 0.210 0.420 - 0.180 0.3 0.7 0.8 - 0.5
B                0.830 0.720 0.200 0.310 - 0.4 0.7 0.9 0.7 -
0.3                                                      2
0.2 0.3 0.3     Network:           - 0.8 0.8 0.6 0.7
                 - 1 1 0 0           1.0 - 0.8 0.3 0.7
C                1 - 1 1 1           0.8 1.0 - 0.8 1.0
0.6              1 1 - 1 0           0.3 0.4 0.4 - 0.3
0.3 0.5 0.2     0 0 1 - 0           0.5 0.9 0.7 0.5 -
                 1 1 0 1 -           ...
D
0.4              Recommendations: Dispositions:
0.1 0.3 0.4     Yes                0.7 0.6 0.4 0.4 0.5 0.6 0.7
                                     0.6 0.6 0.7 0.8 0.9 0.9 0.9
E                Reputations:       0.7 0.5 0.7 0.1 0.2 0.0 0.2
0.7              0                   0.7 0.4 0.5 0.6 0.5 0.6 0.6
0.2 0.2 0.2     - 0.8 0.6 0.5 0.4   0.4 0.1 0.7 0.2 0.8 0.3 0.8
                 0.7 - 0.6 0.5 0.5
Iterations:      0.6 0.7 - 0.4 0.6
7                0.3 0.7 0.9 - 0.6
                 0.5 0.6 1.0 0.6 -

```

Obrázek 5.3.3 - Ukázka formátu vstupního souboru

Přesnou strukturu třídy `ReadData` pak vidíme na obrázku 5.3.4.

ReadData
-fileName: String
-bfr: BufferedReader
-allActors: ArrayList<Actor>
-iterCount: int
-locale: Localization
-recommendations: boolean
-log: Log
+ReadData(final String name, final Localization local, final Log loggie)
+read():void
+getFileName():String
+getAllActors():ArrayList<Actor>
+getIterCount():int
+getRecommendations():boolean

Obrázek 5.3.4 - Struktura třídy `ReadData`

### 5.3.4 Třída Application

Jedná se o spouštěcí třídu obsahující metodu `main()`. Jejím hlavním a jediným účelem je spouštění programu, defaultně lokalizovaným na češtinu. Toho dosahuje vytvořením nové instance třídy `Window` s parametrem `Česky`.

Jelikož tato třída obsahuje pouze metodu `main()` a nemá žádné atributy, nemá smysl ukazovat si její strukturu, která je na první pohled zřejmá.

### 5.3.5 Třída ChangeValues

Tato třída umožňuje změnu načtených dat po kliknutí na příslušné tlačítko v uživatelském rozhraní. Dědí od třídy `JFrame` a zobrazuje okno s textovými poli, která vyplňujeme v případě, že danou hodnotu chceme změnit, jinak je necháme prázdná. Změny potvrdíme příslušným tlačítkem. Jediným podstatným atributem je údaj o aktéru, pro kterého se změny provádí.

Pro obsluhu události stisku tlačítka pro potvrzení zadaných změn je implementována vnitřní třída `SubmitAction` dědicí od třídy `AbstractAction`. Ta obsahuje konstruktor a překrytou metodu `actionPerformed()`, ve které dochází k obsluze události stisku tlačítka.

Dále pak třída ještě obsahuje metodu `getWindow()`, která vrací instanci této třídy a soukromou metodu `getInside()`, ve které jsou vytvořeny a nastaveny veškeré prvky uživatelského rozhraní.

Přesnou strukturu třídy `ChangeValues` pak vidíme na obrázku 5.3.5.

ChangeValues
-serialVersionUID: long
-locale: Localization
-reputationJTF: JTextField
-wdJTF: JTextField
-wrJTF: JTextField
-wgJTF: JTextField
-dispositionsJTF: JTextField
-changedActor: Actor
-window: ChangeValues
-log: Log
+ChangeValues(final Localization local, final Actor act, final Log loggie)
+getWindow(): ChangeValues
-getInside(): void

Obrázek 5.3.5 - Struktura třídy `ChangeValues`

### 5.3.6 Třída Localization

Je to právě tato třída, která se stará o lokalizaci veškerých textů do zvoleného jazyka. Umožňuje českou a anglickou lokalizaci grafického rozhraní a výstupních souborů. Dů-



ležité atributy jsou `locale` a `localeFile`, které se nastavují při vytváření instance této třídy dle předaného parametru. Dle lokalizace nastavené v `locale` se v metodě `localize()` do `localeFile` nastaví název příslušného souboru s lokalizovanými texty.

Pro uložení těchto textů je implementována mapa `guiTexts`, do které se texty nahrají metodou `init()`. Když pak chceme při vytváření grafického rozhraní použít nějaký z lokalizovaných textů, přistupujeme k nim přes metodu `getMessage()` pomocí indexů.

Přesnou strukturu třídy `Localization` lze vidět na obrázku 5.3.6.

Localization
-locale: Locale
-localeFile: ResourceBundle
-guiTexts: HashMap<Integer, String>
+Localization(final String lang)
+localize(final String lang): void
-init(): void
+getMessage(final int key): String

Obrázek 5.3.6 - Struktura třídy `Localization`

### 5.3.7 Třída `Log`

Třída `Log` umožňuje vypisování chybových a systémových hlášek. Dědí od třídy `JTextPane`. Důležitým atributem této třídy je `style`, který určuje styl a velikost písma zobrazovaného v poli logu, dále pak atribut `doc`, jenž zastupuje textové pole, do něhož zapisujeme.

Pro zápis do tohoto pole můžeme použít dvě veřejné metody. Metoda `printToLogNoLine()` vypíše do logu zadaný text v podobě dle nastaveného stylu. Metoda `printToLog()` udělá to samé, avšak na konec tohoto textu ještě přidá ukončovací znak řádky „\n“ a tím text odřádkuje.

Přesnou strukturu třídy `Log` vidíme na obrázku 5.3.7.

Log
-serialVersionUID: long
-doc: StyledDocument
-style: Style
-locale: Localization
+Log(final Localization local)
+printToLog(final String text): void
+printToLogNoLine(final String text): void

Obrázek 5.3.7 - Struktura třídy `Log`

### 5.3.8 Třída `Window`

Jak už název napovídá, tato třída slouží pro kompletní vytvoření a obsluhu událostí základního uživatelského rozhraní aplikace. Dědí ze třídy `JFrame`. Při vytváření instance

této třídy se okno uživatelského rozhraní inicializuje na zvolenou lokalizaci předanou parametrem konstruktora.

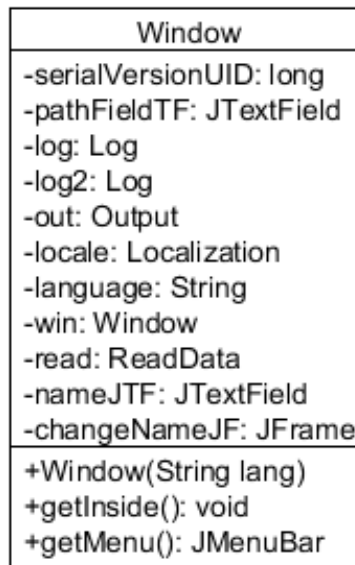
Okno uživatelského rozhraní je rozděleno na lištu menu a vnitřní pole. Toto pole je ještě rozděleno na další čtyři části. V menu se nachází položky Program, Nastavení a Nápověda. V těchto menu nalezneme tlačítka pro načtení dat a jejich změnu, spuštění modelování, nastavení jazyka, nápovědu a další. Každé z těchto tlačítek má svou klávesovou zkratku:

- Procházet - CTRL + S
- Načíst data - CTRL + D
- Načtená data - CTRL + I
- Změnit data - CTRL + N
- Modelovat - CTRL + M
- Konec - ALT + F4
- Jazyk - CTRL + L
- Vyčistit log - CTRL + G
- Nápověda - CTRL + H
- O Programu - CTRL + P

Vnitřní pole ke svému rozdělení používá BorderLayout. V horní části nalezneme pole pro ruční zadání cesty ke vstupnímu souboru a tlačítko, které vyvolá průzkumník, v němž můžeme daný vstupní soubor nalézt v adresářové struktuře. Střední část je rozdělena na levou a pravou část, v levé části se nachází log, do něhož se vypisují systémová a chybová hlášení a v pravé části nalezneme tlačítka pro ovládání programu. Ve spodní části se nachází jediné tlačítko sloužící pro vymazání obsahu logu.

Pro každé z těchto tlačítek je realizována vnitřní třída dědicí od třídy `ActionListener`. V této třídě nalezneme konstruktora, ve které se nastavuje název dané položky a příslušná klávesová zkratka a poté metodu `actionPerformed()`, ve které se provádí obsluha události stisku tlačítka.

Přesnou strukturu třídy `Window` pak vidíme na obrázku 5.3.8.



Obrázek 5.3.8 - Struktura třídy Window

### 5.3.9 Třída Model

Jedná se o nejdůležitější třídu aplikace, ve které je implementovaný v přechozí kapitole popsaný model důvěry. K samotnému výpočtu dochází v metodě `modeling()`. Po inicializaci proměnných se dle stavu ukazatele `recommendations` a toho, zda jsou aktéři přímými sousedy rozhodne, zda se bude počítat s doporučeními či ne. Pokud ne, proběhne výpočet dle v předchozí kapitole uvedených vztahů, avšak bez doporučení a jeho váhy. Pro výpočet reputace je implementovaná metoda `getReputation()`.

Pokud budeme počítat s doporučeními, musíme si nejdříve najít cesty mezi aktéry v síti. K tomuto účelu je implementovaná metoda `getShortestPath()`, která nalezne nejkratší cesty mezi aktéry. Ke své práci používá dvě metody, `getInvolvedActors()` a `addToInvolved()`.

Metoda `getInvolvedActors()` vyhledá všechny cesty mezi aktéry v síti, kromě těch, které jsou hlouběji, než ty již nalezené, jelikož ty nemohou být kratší, a uloží je do seznamu, který vrací. Z těchto cest potom metoda `getShortestPath()` vybírá ty nejkratší.

Po nalezení nejkratších cest už můžeme spočítat doporučení, k tomu slouží metoda `getRecommendation()`.

Přesnou strukturu třídy `Model` vidíme na obrázku 5.3.9.

Model
-LAMBDA: double[] -involvedActors: ArrayList<Actor> -allActors: ArrayList<Actor> -iterCount: int -log: Log -locale: Localization -recommendations: boolean
+Model(final int iterations, final ArrayList<Actor> actors, final boolean rec, final Localization local, final Log loggie) +modeling(): void +getRecommendation(final int iteration, final Actor first, final Actor second): double +getReputation(final Actor first, final int iteration): double -getShortestPath(final Actor first, final Actor second, final int iteration): void -getInvolvedActors(final Actor first, final Actor second, final int iteration): ArrayList<ArrayList<Actor> > -addToInvolved(LinkedList<Actor> actors, Actor second): ArrayList<Actor>

Obrázek 5.3.9 - Struktura třídy Model

## 6 Experimenty s modelem

V této části si popíšeme provedené experimenty na implementovaném modelu a rozebereme si jejich výsledky. Pro lepší představu o získaných datech bude každý experiment doplněn o grafy. Experimenty byly provedeny celkem 3.

### 6.1 První experiment

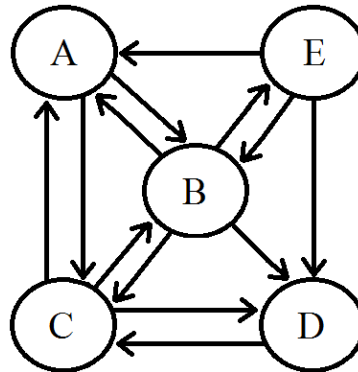
#### 6.1.1 Zadaná data

První experiment byl proveden nad sítí o pěti aktérech, jmenovitě A, B, C, D a E. Na Obr. 6.1.1 vidíme matici sousednosti této sítě. Jak jsme si již popsali v kapitole 3, v matici sousednosti jsou řádky i sloupce tvořeny aktéry sítě.

$$\begin{pmatrix} - & 1 & 1 & 0 & 0 \\ 1 & - & 1 & 1 & 1 \\ 1 & 1 & - & 1 & 0 \\ 0 & 0 & 1 & - & 0 \\ 1 & 1 & 0 & 1 & - \end{pmatrix}$$

Obrázek 6.1.1 - Matice sousednosti sítě prvního experimentu

Pro lepší představu o síti si ukážeme i její grafovou reprezentaci na Obr. 6.1.2.



Obrázek 6.1.2 - Grafová reprezentace sítě prvního experimentu

Z Obr. 6.1.2 je zřejmé, že aktér B je centrálním aktérem sítě, jelikož v síti existuje vazba mezi ním a všemi ostatními aktéry sítě. Přesto však není prvkem nutným, jelikož síť by existovala i bez jeho přítomnosti. Naopak za nejméně sociální aktéry můžeme označit aktéry E a D, kde aktér D má dokonce jen jednu vazbu, která vychází od něj, což ho činí velice izolovaným.

Pro každého z aktérů jsou předem známé hodnoty váhových koeficientů doporučení, reputace i dispozice k důvěře. Hodnoty těchto koeficientů pro tento experiment vidíme v Tabulce 2.

Tabulka 2 - Hodnoty váhových koeficientů pro první experiment

Název aktéra	A	B	C	D	E
Váha doporučení	0,1	0,2	0,3	0,1	0,2
Váha reputace	0,4	0,3	0,5	0,3	0,2
Váha dispozice	0,2	0,3	0,2	0,4	0,2

Z obr 6.1.2 lze poznat, že se všichni aktéři v síti vzájemně neznají. Pro účely modelování je však nutné, aby existoval vztah mezi všemi aktéry v síti a proto se přistupuje k dopočtení důvěry mezi těmi aktéry, kteří se neznají a tak vznikne vztah i mezi těmito aktéry. Tuto důvěru dopočteme jako součin reputace daného aktéra s váhovým koeficientem reputace u aktéra, který danému aktéru důvěřuje.

Výslednou matici důvěry si můžeme prohlédnout na Obr. 6.1.3. Je zřejmé, že nyní již existuje vztah mezi všemi aktéry v síti a je tak splněna podmínka pro modelování důvěry.

$$\begin{pmatrix} -0.450 & 0.300 & 0.200 & 0.160 \\ 0.350 & -0.760 & 0.130 & 0.200 \\ 0.650 & 0.520 & -0.480 & 0.300 \\ 0.090 & 0.210 & 0.420 & -0.180 \\ 0.830 & 0.720 & 0.200 & 0.310 \end{pmatrix}$$

Obrázek 6.1.3 - Matice důvěry sítě prvního experimentu

Jak jsme si ukázali v teoretické části, je výpočet aktuální důvěry závislý na aktuální dispozici k důvěře daného aktéra. Hodnoty těchto dispozic aktérů pro jednotlivé iterace vidíme v Tabulce 3.

Tabulka 3 - Dispozice k důvěře jednotlivých aktérů pro první experiment

Číslo iterace	0	1	2	3	4	5	6
A	0,7	0,6	0,4	0,4	0,5	0,6	0,7
B	0,6	0,6	0,7	0,8	0,9	0,9	0,9
C	0,7	0,5	0,7	0,8	0,1	0,0	0,2
D	0,7	0,4	0,5	0,6	0,5	0,6	0,6
E	0,4	0,1	0,7	0,2	0,8	0,3	0,8

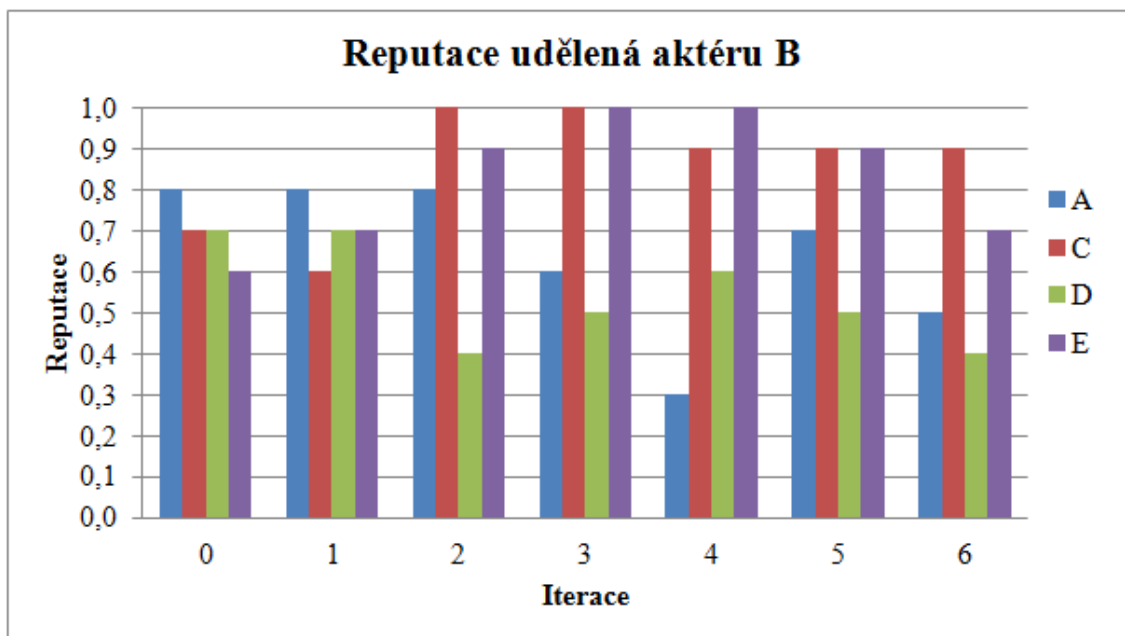
Z Tabulky 3 vidíme, že aktéři C a E jsou velmi nestabilní v dispozicích k důvěře. Za poměrně stabilní můžeme naopak označit aktéry B a D.

## 6.1.2 Výsledky experimentu

Na souhrnné výsledky změn důvěry v jednotlivých iteracích se podíváme na konci této kapitoly, nyní se však zaměříme na několik podstatných detailů, které ovlivňovaly právě

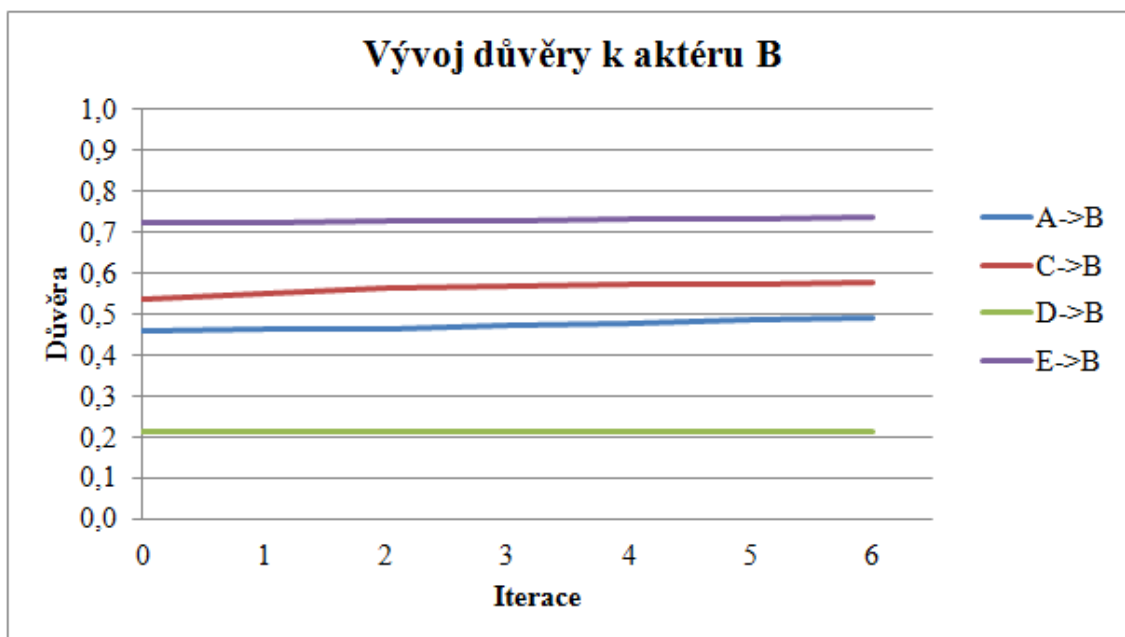
tyto změny důvěry.

Ze zadaných dat vyplývá, že nejoblíbenějším aktérem sítě se zdá být aktér B, který získává v každé iteraci od většiny ostatních aktérů vysoké hodnoty reputace a díky tomu jeho důvěryhodnost roste nejvíce ze všech aktérů. Tyto udělované hodnoty reputace vidíme v Grafu 1.



Graf 1 - Reputace udělená aktéru B

Pro představu o jednotlivých vztazích mezi ostatními aktéry a aktérem B se podívejme na Graf 2.



Graf 2 - Vývoj důvěry k aktéru B

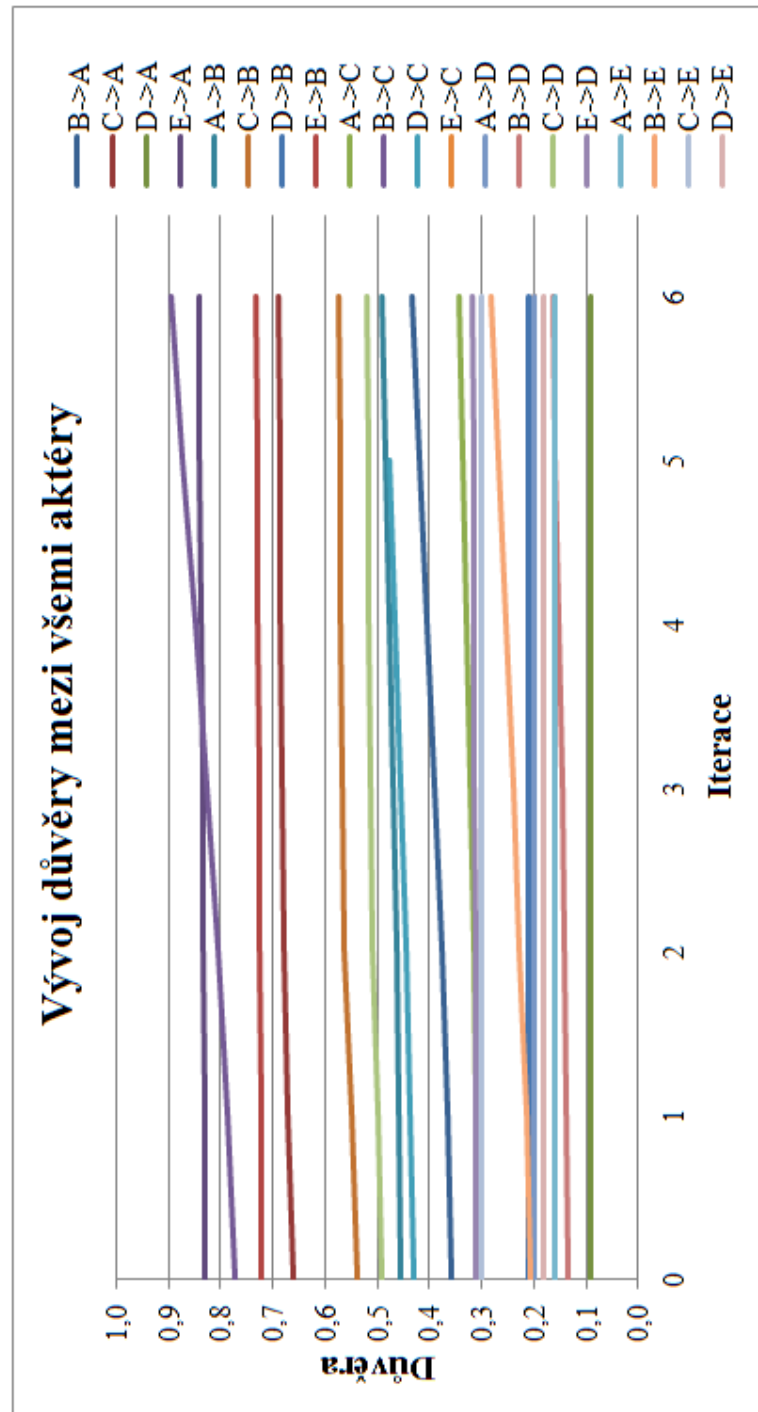
Z Grafu 2 je zřejmé, že důvěra mezi aktéry D a B se změnila téměř nezatelně, to je způsobeno tím, že aktér D dává nízkou váhu reputaci a extrémně nízkou váhu doporučení.

O něco větší nárůst pozorujeme u aktérů E a A, u kterých už jsou hodnoty vah vyrovnanější a způsobeno je to také vyšším trendem důvěry, jelikož tito aktéři se s aktérem B lépe znali už na začátku.

K největšímu nárůstu dochází u aktéra C, který má vysoké váhy jak doporučení, tak reputace.

Zbylé pohyby hodnoty aktuální důvěry a rozdíly v nárůstech této důvěry už jsou převážně způsobeny kolísáním hodnoty aktuální dispozice k důvěře.

Vývoj důvěry v iteracích mezi všemi aktéry vidíme v Grafu 3.



Graf 3 - Vývoj důvěry mezi všemi aktéry v iteracích



## 6.2 Druhý experiment

### 6.2.1 Zadaná data

Pro druhý experiment jsem si dovolil převzít část reálných dat z diplomové práce [14]. Jakým způsobem a v jakém časovém období byla tato reálná data získána popsal autor následovně:

*Data byla získávána na sociální síti Facebook od října roku 2011 do dubna roku 2012 ve vytvořené skupině s několika účastníky. Prvních sedm týdnů byla skupina čtyřčlenná, dalších sedm týdnů sedmičlenná a posledních sedm týdnů dvanáctičlenná.*

*Každý účastník se vstupem do této skupiny odevzdal uvítací dotazník, ve kterém odpověděl na otázky, díky nimž se určily hodnoty váhy doporučení, reputace a dispozice k důvěře. Dále každý odpověděl na základní otázky ohledně dalších účastníků, zdali se znají nebo ne a jak moc a jak druhého účastníka hodnotí.*

*Každý týden, který byl zvolen jako iterace, potom účastníci odevzdali dotazník s hodnocením ostatních a své momentální dispozice k důvěře. Díky tomu byly získávány hodnoty reputace a nálady subjektu ve všech iteracích.*

*Toto se opakovalo po sedm iterací, kdy někteří z účastníků přivedli do skupiny svého známého a napsali na něj krátké doporučení pro ostatní. Na základě doporučení si mohli ostatní utvořit názor na nově příchozí. Všichni nováčci stejně jako jejich předchůdci vyplnili vstupní dotazník a vše se opakovalo.*

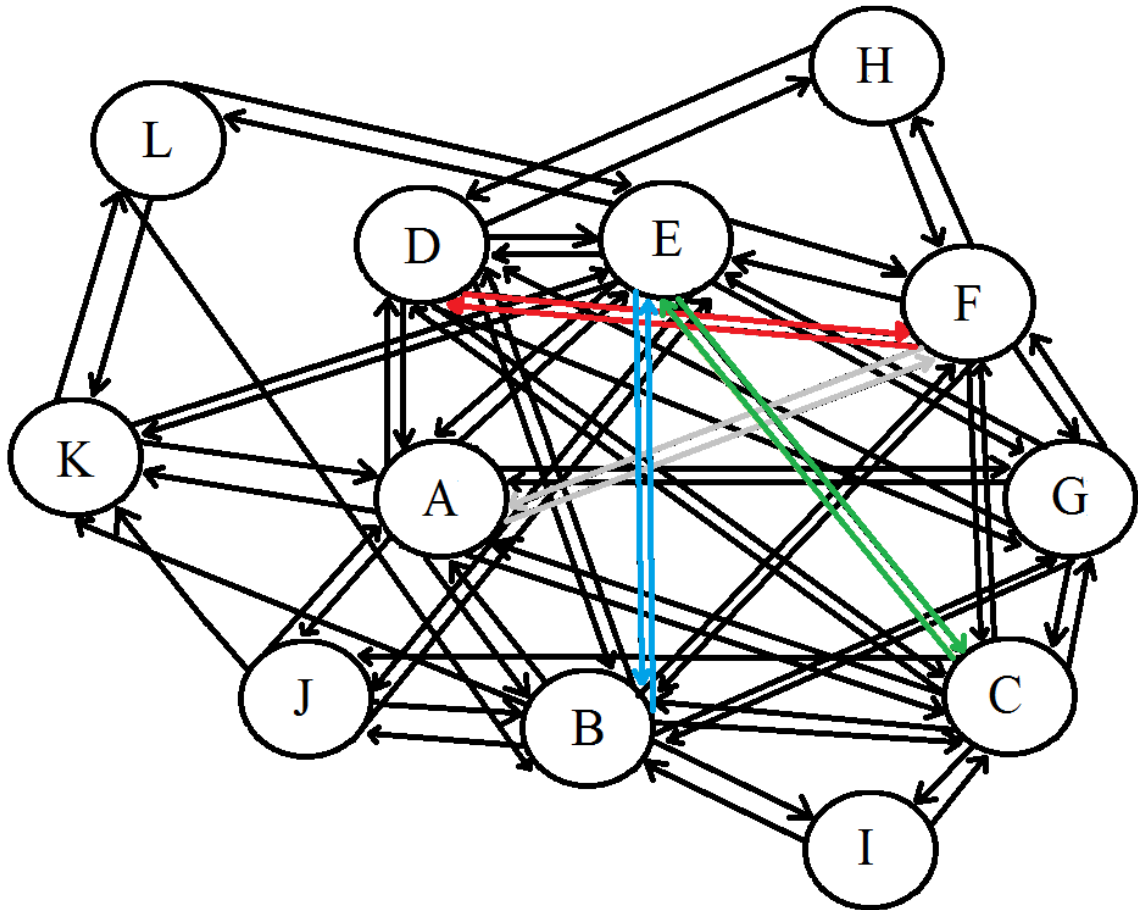
*Stejně tak probíhala i poslední třetí část experimentu s konečnými dvanácti aktéry.*

Převzata byla data ze třetí části experimentu, ve které se skupina skládala z dvanácti aktérů, A - I. Matici sousednosti této sítě si můžeme prohlédnout na Obr. 6.2.1.

-	1	1	1	1	1	1	0	0	1	1	1
1	-	1	1	1	1	1	0	1	1	1	0
1	1	-	1	1	1	1	0	1	1	0	0
1	1	1	-	1	1	1	1	0	0	0	0
1	1	1	1	-	1	1	0	0	1	1	1
1	1	1	1	1	-	1	1	0	0	0	0
1	1	1	1	1	1	-	0	0	0	0	0
0	0	0	1	0	1	0	-	0	0	0	0
0	1	1	0	0	0	0	0	-	0	0	0
1	0	0	0	1	0	0	0	0	-	1	1
1	0	0	0	1	0	0	0	0	0	-	1
1	1	0	0	1	0	0	0	0	1	1	-

Obrázek 6.2.1 - Maticе sousednosti sítě druhého experimentu

Opět si ukážeme i grafovou reprezentaci dané sítě, viz Obr. 6.2.2.



Obrázek 6.2.2 - Grafová reprezentace sítě druhého experimentu

Obrázek 6.2.2 poukazuje na to, že pro větší sítě by již nebylo příliš vhodné používat grafovou reprezentaci, jelikož graf takovéto sítě by byl velmi nepřehledný. I přes nepřehlednost tohoto grafu si můžeme povšimnout, že nejvíce vazeb mají aktéři A, B, D a E a tak je můžeme označit jako centrální aktéry této sítě. Nejméně interaktivními aktéry jsou pak aktéři H a I, kteří mají oba vztahy pouze se dvěma aktéry z celé sítě.

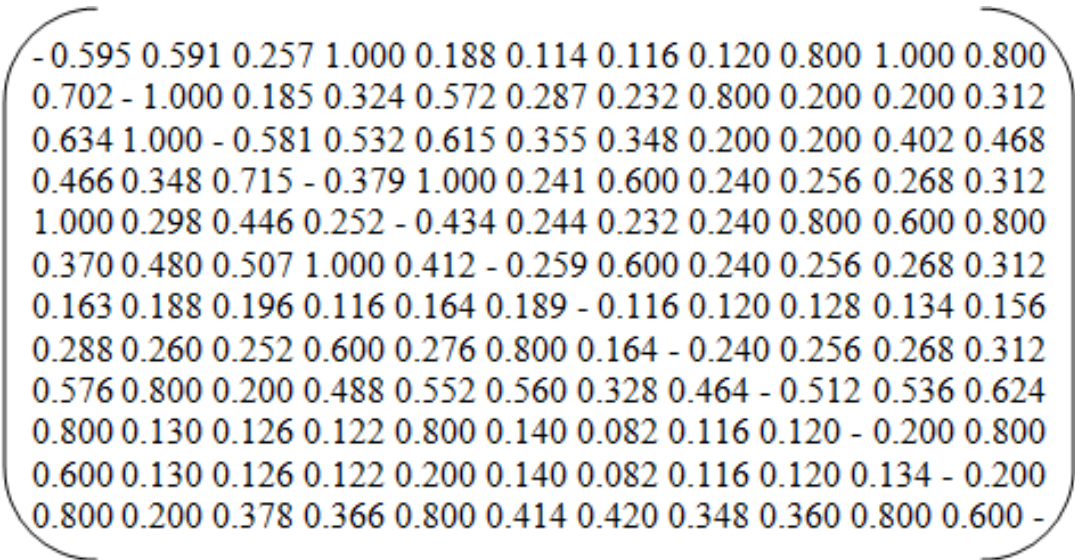
Hodnoty váhových koeficientů pro druhý experiment vidíme v Tabulce 4.

Tabulka 4 - Hodnoty váhových koeficientů pro druhý experiment

Název aktéra	A	B	C	D	E	F	G	H	I	J	K	L
Váha doporučení	0,6	0,4	0,4	0,4	0,4	0,4	0,8	0,2	0,8	0,8	0,0	0,4
Váha reputace	0,2	0,4	0,6	0,4	0,4	0,4	0,2	0,4	0,8	0,2	0,2	0,6
Váha dispozice	0,5	0,7	0,6	0,6	0,7	0,6	0,4	0,5	0,8	0,3	0,6	0,4

Ani v tomto experimentu neexistují vztahy mezi všemi aktéry v síti a proto je nutné opět zbylé vztahy vytvořit postupem popsaným v předchozím experimentu.

Následně vytvořenou matici důvěry pro tento experiment vidíme na Obr. 6.2.3.



Obrázek 6.2.3 - Matice důvěry sítě druhého experimentu

Pro zahájení výpočtu už nám chybí pouze aktuální dispozice k důvěře v iteracích jednotlivých aktérů, ty nalezneme v Tabulce 5.

Tabulka 5 - Dispozice k důvěře jednotlivých aktérů pro druhý experiment

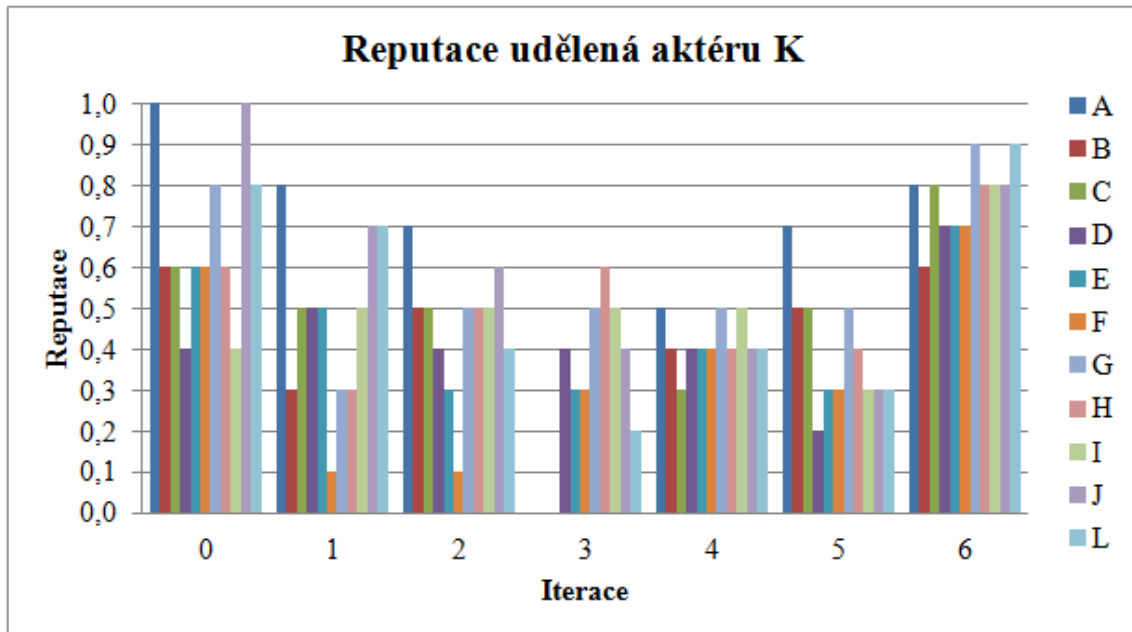
Číslo iterace	0	1	2	3	4	5	6
<b>A</b>	0.6	0.6	0.5	0.8	0.8	0.7	0.6
<b>B</b>	0.6	0.6	0.7	0.7	0.6	0.5	0.7
<b>C</b>	0.8	0.8	0.7	0.6	0.2	0.2	0.8
<b>D</b>	0.6	0.5	0.5	0.5	0.7	0.7	0.7
<b>E</b>	0.5	0.4	0.5	0.6	0.7	0.8	0.8
<b>F</b>	0.6	0.7	0.3	0.3	0.2	0.4	0.5
<b>G</b>	0.7	0.8	0.8	0.9	0.9	0.9	0.9
<b>H</b>	0.6	0.7	0.7	0.7	0.7	0.7	0.7
<b>I</b>	0.2	0.4	0.6	0.8	1.0	1.0	0.2
<b>J</b>	0.6	0.8	0.3	0.7	0.4	0.7	0.2
<b>K</b>	0.4	0.4	0.5	0.5	0.4	0.3	0.3
<b>L</b>	0.2	0.5	0.5	0.4	0.6	0.8	0.8

Pokud se pozorněji podíváme na tabulku 5 zjistíme, že aktéři F, I a L jsou v dispozicích k důvěře velmi nestálí a vyskytují se u nich velké propady dispozice. Ostatní aktéři se zdají být stabilnější a jejich dispozice se tak výrazně nemění.

### 6.2.2 Výsledky experimentu

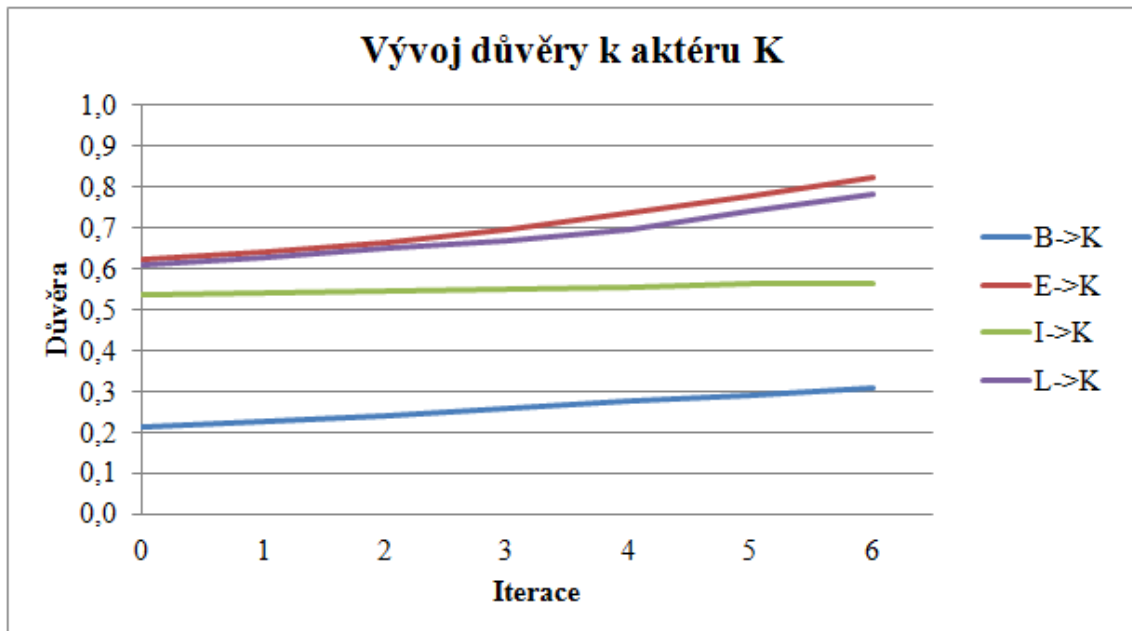
První zajímavostí, na níž se podíváme blíže, je vývoj reputací udílených ostatními aktéry aktéru K. Jak můžeme vidět na Grafu 4, v počátečních iteracích tyto reputace postupně klesají, v některých případech se dotknou až nulové hranice, avšak v posledních dvou

iteracích dojde k velkému vzrůstu, což naznačuje, že aktér K byl zpočátku izolován od zbytku sítě a do interakce se dostal až v posledních iteracích.



Graf 4 - Reputace udělená aktéru K

Jelikož reputace ovlivňuje důvěru v daného aktéra, je jisté, že i míra důvěry v aktéra K měla tendenci růst jen málo a v posledních iteracích znatelně stoupla. Její vývoj můžeme sledovat na grafu 5. Vzhledem k velikosti sítě jsou v Grafu 5 uvedeny pouze vztahy s aktéry, u nichž jsou tyto změny nejznatelnější.

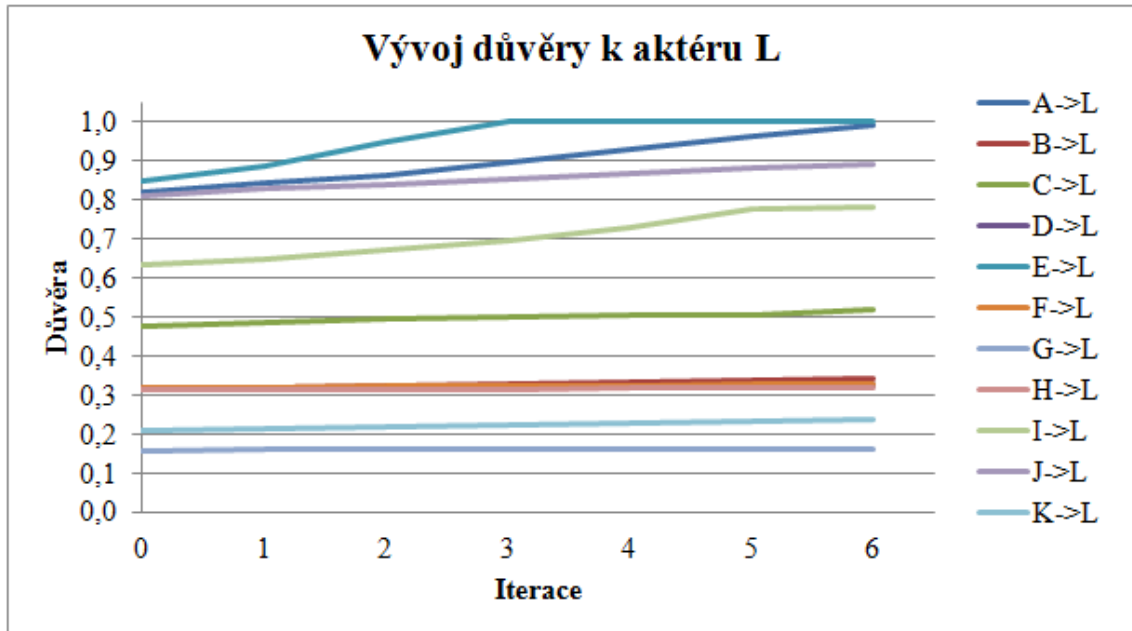


Graf 5 - Vývoj důvěry k aktéru K

Další zajímavostí je vývoj důvěry k aktéru L. Větší množství aktérů zaznamenalo v průběhu iterací značný nárůst důvěry k tomuto aktéru, několik vazeb se dokonce dostalo nebo

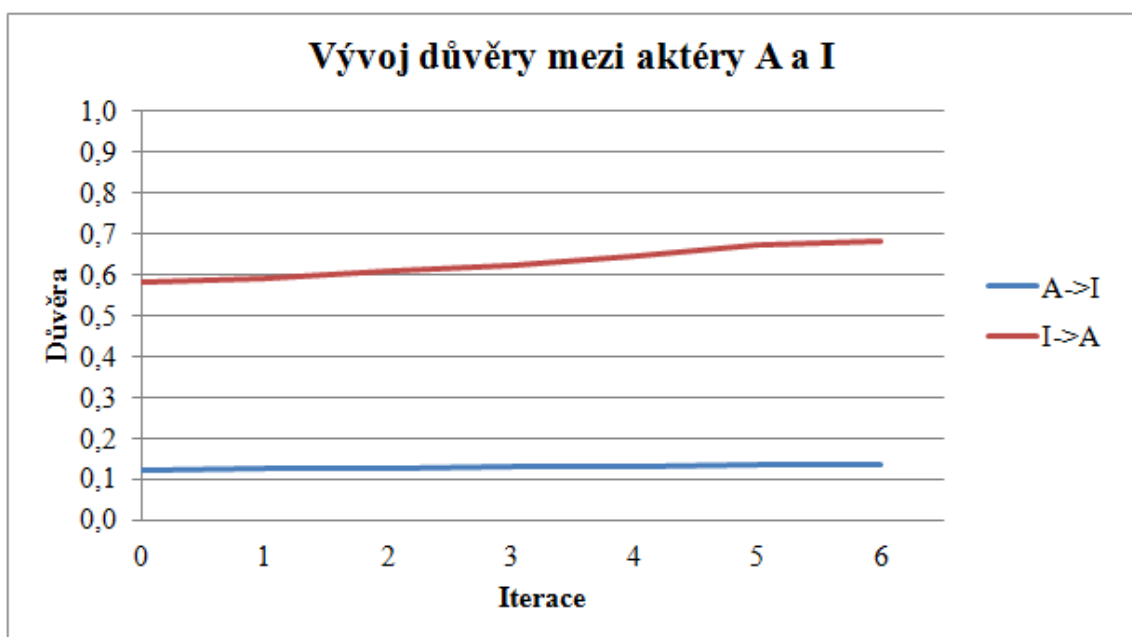
velmi blížilo hodnotě 1, tedy úrovni absolutní důvěry. Způsobeno je to tím, že aktér L je mezi ostatními aktéry velmi oblíben a získával v každé iteraci vysoké hodnoty reputace od ostatních aktérů a zároveň i jeho reputace v síti byla vysoká.

Vývoj důvěry ostatních aktérů v aktéra L je zobrazen v Grafu 6.



Graf 6 - Vývoj důvěry k aktéru L

Jako poslední si předvedeme názornou ukázkou jedné z častých vlastností důvěry, jíž je asymetričnost. Tento pojem znamená, že první aktér důvěřuje druhému, avšak opačně to neplatí. Hezkým příkladem této vlastnosti je vztah mezi aktéry A a I. Asymetričnost zde vzniká proto, že aktér I přikládá velkou váhu reputaci, jejíž hodnota u aktéra A je vysoká. Z pohledu aktéra A je pak situace opačná. Situace je ilustrovaná Grafem 7.



Graf 7 - Vývoj důvěry mezi aktéry A a I

Vývoj důvěry mezi všemi aktéry v jednotlivých iteracích by se graficky špatně zobrazoval, výsledný graf by byl velmi nepřehledný. Na výsledky tohoto experimentu lze nahlédnout v souboru na příloženém CD.

## 6.3 Třetí experiment

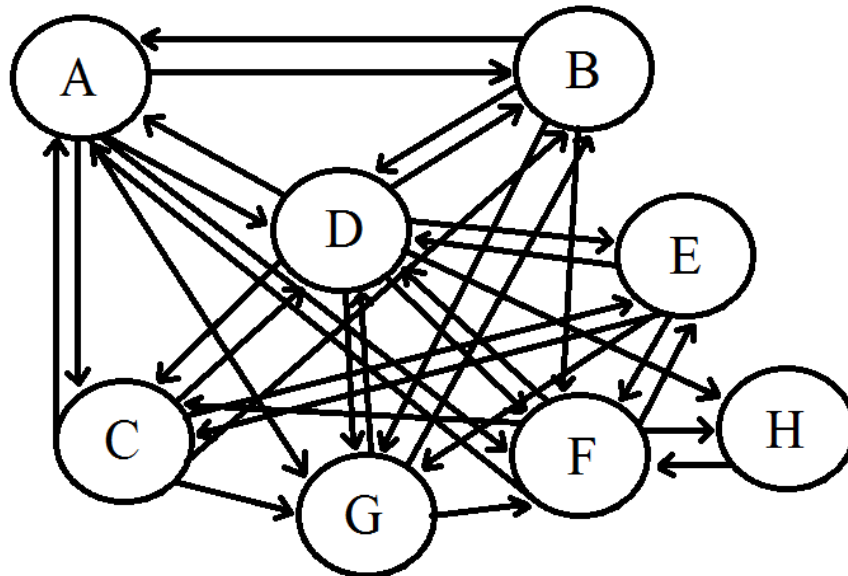
### 6.3.1 Zadaná data

Poslední experiment byl realizován nad menší sítí o osmi aktérech, A - H. Matici sousednosti této sítě vidíme na Obr. 6.3.1.

$$\begin{pmatrix} - & 1 & 1 & 1 & 0 & 1 & 1 & 0 \\ 1 & - & 0 & 1 & 0 & 1 & 1 & 0 \\ 1 & 1 & - & 1 & 1 & 0 & 1 & 0 \\ 1 & 1 & 1 & - & 1 & 1 & 1 & 1 \\ 0 & 0 & 1 & 1 & - & 1 & 1 & 0 \\ 1 & 0 & 1 & 1 & 1 & - & 0 & 1 \\ 0 & 1 & 0 & 1 & 0 & 1 & - & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & - \end{pmatrix}$$

Obrázek 6.3.1 - Matice sousednosti sítě třetího experimentu

Grafová reprezentace dané sítě je zobrazena na Obr. 6.3.2.



Obrázek 6.3.2 - Grafová reprezentace sítě třetího experimentu

Z obrázku 6.3.2 lze vidět, že už pro tento počet aktérů v síti začíná být grafová reprezentace nepřehledná, avšak podstatné vazby jsou z ní stále vidět.

Povšimnout si tak můžeme, že centrálním aktérem této sítě je aktér D, jenž má vazbu se všemi ostatními aktéry. Aktér H je naopak velmi izolovaný od ostatních aktérů a sám má vazbu pouze jednu vycházející od něj a to s aktérem F.

V Tabulce 6 jsou zaneseny hodnoty váhových koeficientů pro třetí experiment.

Tabulka 6 - Hodnoty váhových koeficientů pro třetí experiment

Název aktéra	A	B	C	D	E	F	G	H
Váha doporučení	0,6	0,6	0,6	0,9	0,8	0,6	0,8	0,7
Váha reputace	0,6	0,5	0,7	0,6	0,7	0,8	0,9	0,7
Váha dispozice	0,7	0,7	0,6	0,5	0,6	0,5	0,5	0,6

Na Obr. 6.3.3 si pak můžeme prohlédnout matici důvěry této sítě.

-0.726	0.631	0.126	0.360	0.298	0.532	0.420	
0.650	-0.350	0.102	0.250	0.572	0.950	0.350	
0.820	0.614	-0.486	0.333	0.420	0.355	0.280	
0.195	0.400	0.654	-0.480	0.598	0.600	0.730	
0.560	0.490	0.446	0.473	-0.412	0.377	0.350	
0.557	0.560	0.500	0.931	1.000	-0.420	0.510	
0.450	0.777	0.810	0.269	0.540	0.467	-0.630	
0.420	0.490	0.420	0.560	0.420	0.863	0.630	-

Obrázek 6.3.3 - Matice důvěry sítě třetího experimentu

Poslední zbývající proměnnou, o níž jsme se ještě nezmínili, je dispozice k důvěře. Hodnoty této proměnné pro jednotlivé aktéry v iteracích pro tento experiment se nachází v Tabulce 7.

Tabulka 7 - Dispozice k důvěře jednotlivých aktérů pro třetí experiment

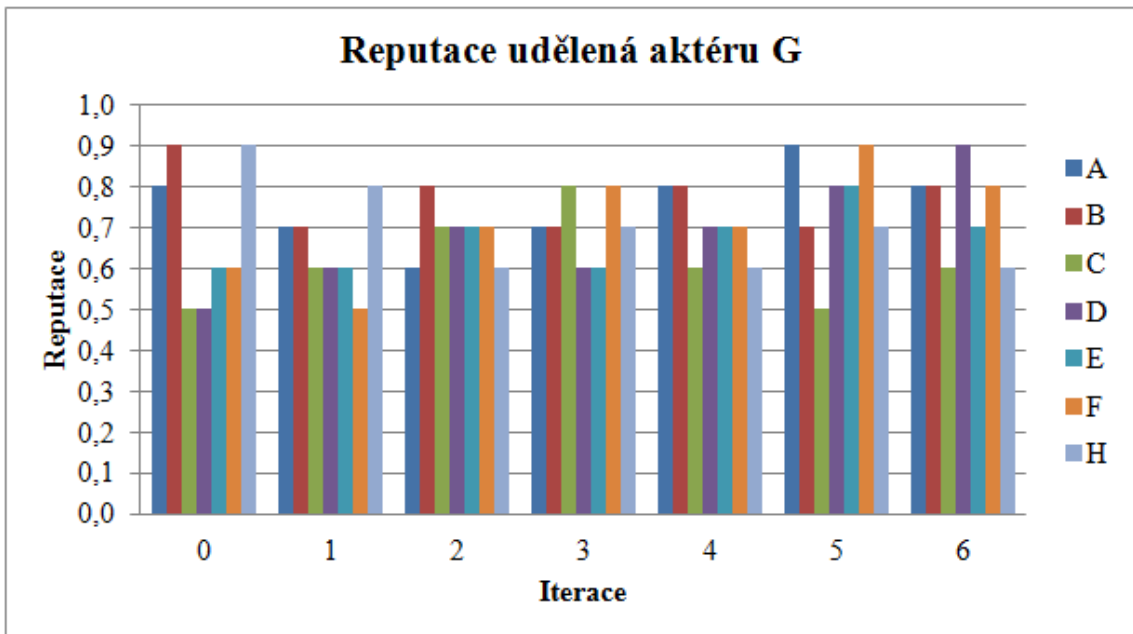
Číslo iterace	0	1	2	3	4	5	6
<b>A</b>	0.7	0.6	0.7	0.8	0.9	0.8	0.6
<b>B</b>	0.6	0.5	0.8	0.7	0.6	0.7	0.8
<b>C</b>	0.7	0.8	0.6	0.5	0.7	0.5	0.7
<b>D</b>	0.8	0.9	0.7	0.6	0.8	0.6	0.7
<b>E</b>	0.7	0.6	0.5	0.7	0.6	0.4	0.6
<b>F</b>	0.6	0.7	0.9	0.8	0.7	0.9	0.8
<b>G</b>	0.7	0.8	0.6	0.8	0.9	0.8	0.9
<b>H</b>	0.8	0.8	0.7	0.7	0.6	0.5	0.7

Z Tabulky 7 vidíme, že pro tento experiment mají všichni aktéři vysoké dispozice a žádný z nich nezaznamenává velké propady v hodnotách, což naznačuje, že důvěra mezi aktéry by mohla rovnoměrně a rychle stoupat.

### 6.3.2 Výsledky experimentu

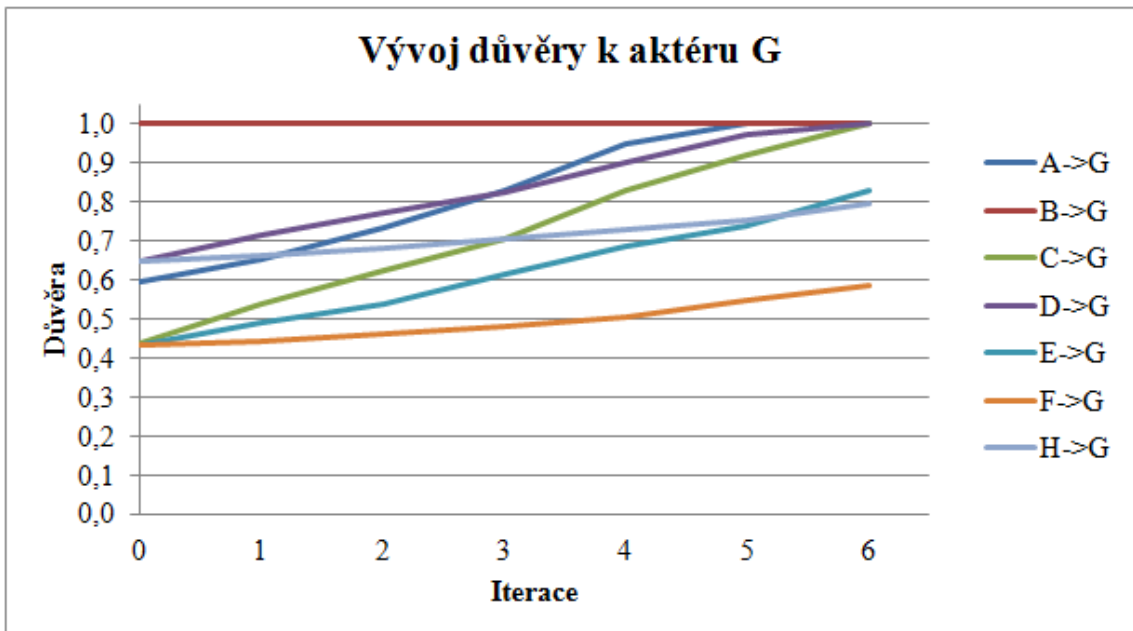
Největší nárůst důvěry lze pozorovat u aktéra G. Míra důvěry ostatních aktérů v tohoto aktéra se v průběhu iterací u většiny z nich dostala až na hodnotu 1, tedy absolutní důvěru.

Způsobeno je to vysokou reputací aktéra G v síti, obecně vysokými hodnotami váhových koeficientů v tomto experimentu a také reputací, kterou aktéru G udíleli v průběhu iterací ostatní aktéři. Tyto udílené reputace jsou zobrazené v Grafu 8.



Graf 8- Reputace udělená aktéru G

Jak již bylo řečeno, na základě těchto okolností, byla změna důvěry v aktéra G nejznatelnější změnou v síti. Jak přesně se důvěra ostatních aktérů v aktéra G vyvíjela v jednotlivých iteracích lze vidět v Grafu 9.

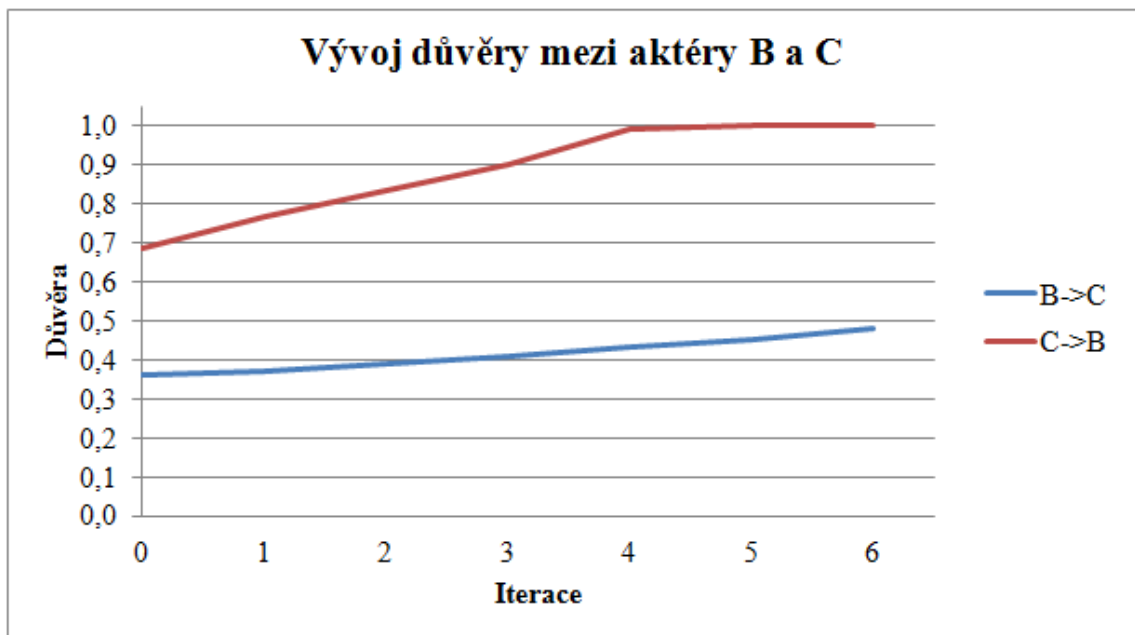


Graf 9 - Vývoj důvěry k aktéru G

Z grafu 9 vidíme, že důvěra aktéra B v aktéra G byla již na začátku iterací rovna 1 a zůstala tak po celou dobu.



I v tomto experimentu můžeme nalézt příklad asymetričnosti důvěry. Je jím vazba mezi aktéry B a C, kterou zobrazuje Graf 10.



Graf 10 - Vývoj důvěry mezi aktéry B a C

Tato asymetrie je způsobena tím, že na rozdíl od aktéra C má aktér B nízké hodnoty váhových koeficientů a taktéž dispozice k důvěře.

Ani v tomto experimentu nelze přehledně graficky zobrazit vývoj důvěry mezi všemi aktéry. Díky vysokým hodnotám váhových koeficientů docházelo k velkým nárůstům důvěry ve většině vazeb a větší množství vazeb se přiblížilo či dokonce dosáhlo hodnoty 1.

## 7 Závěr

Cílem této práce bylo předvést jeden z možných způsobů, jak nahlížet na důvěru a její modelování v informačních technologiích. Současně se snažila vysvětlit pojem sociálních sítí a ukázat spojnici mezi těmito dvěma pojmy. Zároveň se tato práce pokusila nastínit chování jednoho z modelů důvěry a jeho vlastnosti v experimentech. Přestože jsou pojmy důvěra a sociální síť obecně známe, jejich přesné definice už zasahují do odbornějších sfér a proto byly v teoretické části podrobně rozebrány.

Do teoretické části spadají kapitoly 2, 3 a 4, v nichž byly podrobně popsány základní pojmy a vysvětleny veškeré další poznatky podstatné pro realizaci modelování důvěry v sociální síti. Zároveň byl představen použitý model důvěry a také veškerá problematika, která tento model provází.

Kapitola 5 se zabývá popisem implementovaného softwarového řešení problému modelování důvěry v sociální síti v jazyce Java. Během realizace tohoto řešení jsem dostal možnost si procvičit objektové programování a práci se třídami, stejně jako návrh kvalitního grafického rozhraní. Zároveň jsem si osvojil i dovednost vytvářet lokalizované aplikace. Vzhledem ke specifčnosti zadání je však uplatnění programu omezeno pouze na podobná zadání.

V poslední kapitole jsou prezentovány provedené experimenty a diskuze jejich výsledků. Tato trojice experimentů byla zvolena tak, aby otestovala chování modelu pro malé sociální sítě a zároveň naznačila jeho chování pro krajní případy. V diskuzi je ukazováno na zajímavé situace, které nastaly během realizace těchto experimentů, a je popisován možný důvod jejich vzniku.

K samotné práci je přiloženo CD na němž lze nalézt zdrojové soubory softwarové realizace a také její spustitelnou verzi spolu s použitými vstupními daty všech experimentů. Dále zde nalezneme i výsledky těchto experimentů, vygenerovanou dokumentaci ke zdrojovému kódu a tento text v elektronické podobě.

## **Přehled zkratk**

UML	Unified Modelling Language je grafický jazyk
XML	Extensible Markup Language je značkovací jazyk

## Literatura, elektronické odkazy a zdroje

- [1] American Heritage Dictionary  
URL: <http://ahdictionary.com/word/search.html?q=trust>  
[citováno 15. února 2013]
- [2] Rousseau, D. M., Sitkin, S. B., Burt, R. S., and Camerer, C. (1998). "Not so Different After All: A Cross-Discipline View of Trust," in Academy of Management Review, 23, 393-404.
- [3] Lewicki, R. J., McAllister, D. J., & Bies, R. J. (1998). Trust and distrust: New relationships and realities. Academy of Management Review, 23, 438-458.
- [4] Almenarez F., Marín A., Díaz D., Sanchez J.: Developing a Model for Trust Management in Pervasive Devices. In proceedings of PerSec 2006 (Third IEEE International Workshop on Pervasive Computing and Communication Security Held in conjunction with IEEE PerCom 2006), Pisa, Italy, 2006
- [5] Science Direct  
URL: <http://www.sciencedirect.com/>  
[citováno 18. února 2013]
- [6] Urbánek Š.: Modelling and Simulation of Trust Evolution in Complex Systems, Master Thesis, STU – Faculty of Informatics and Information Technologies, Bratislava, 2004.
- [7] Strategic Public Relations  
URL: <http://www.prblog.typepad.com/>  
[citováno 22. února 2013]
- [8] Hanneman R., Riddle M: Introduction to Social Networks, Department of Sociology, UCLA, Riverside, 2005
- [9] Social network analysis pro začátečníky  
URL: <http://www.lupa.cz/clanky/social-network-analysis-pro-zacatecniky/>  
[citováno 2. dubna 2013]
- [10] Source Code Mania  
URL: <http://www.sourcecodemania.com/>  
[citováno 12. března 2013]
- [11] Wikipedia, the free encyclopedia  
URL: <http://www.wikipedia.org/>  
[citováno 12. března 2013]
- [12] University of Hawaii System  
URL: <http://www.hawaii.edu/>  
[citováno 14. března 2013]
- [13] Stanford Encyclopedia of Philosophy  
URL: <http://plato.stanford.edu/entries/prisoner-dilemma/>  
[citováno 21. března 2013]
- [14] Václavěk V.: Modelování důvěry v sociální síti, Diplomová práce, Katedra informatiky, Západočeská univerzita v Plzni, Plzeň, 2012.

# Přílohy

## Uživatelský manuál

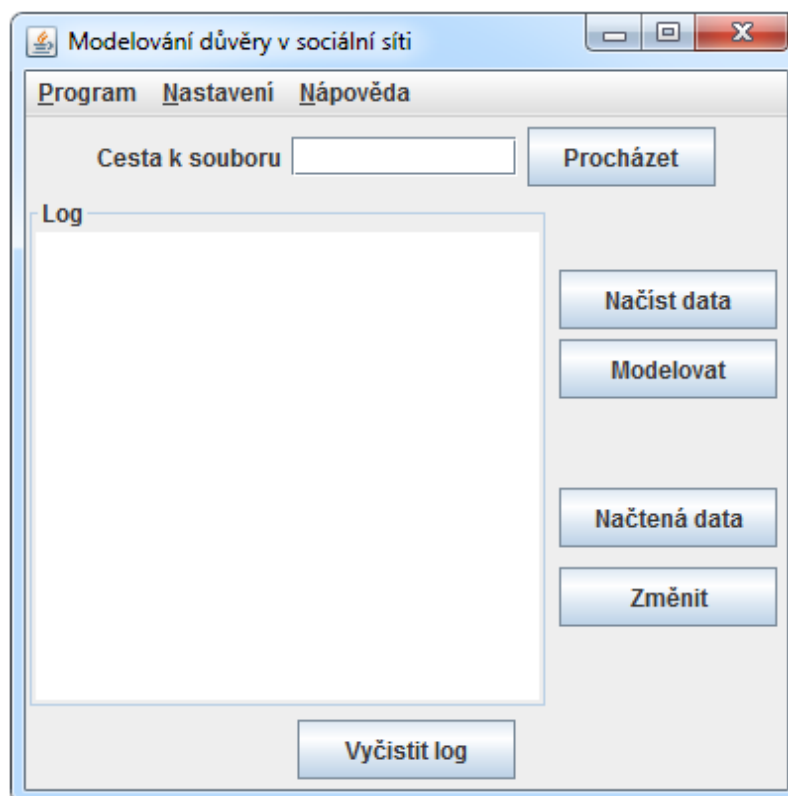
Aplikace je vytvořena v programovacím jazyce Java. Pro její spuštění je tedy nutné mít nainstalovanou aktuální verzi Javy, samotný program byl tvořen ve verzi 1.6 Update 22. Javu je možné nainstalovat po stažení z oficiálních stránek tvůrců tohoto jazyka jimiž jsou stránky <http://java.com/en/download/index.jsp>.

Adresářová struktura vypadá takto:

- bin - spustitelná verze aplikace
- src - zdrojové soubory aplikace
- text - tento text ve formátu PDF
- experiments - vstupní data pro experimenty a jejich výsledky
- doc - vygenerovaná JavaDoc dokumentace

Samotnou aplikaci ke spuštění tedy nalezneme v adresáři bin, pro její spuštění stačí dvojklik na příslušný .jar soubor.

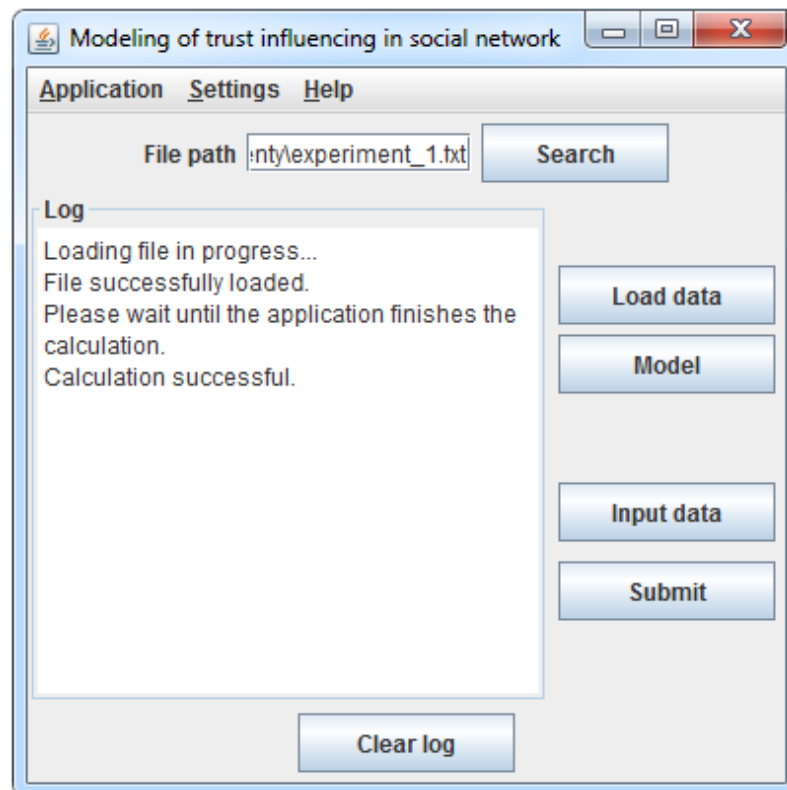
Po spuštění se objeví grafické rozhraní aplikace v české lokalizaci, které můžeme vidět na Obr. P1.



Obrázek P1 - Grafické rozhraní aplikace v češtině

Tuto lokalizaci můžeme změnit na anglickou lokalizaci, jak bude popsáno níže. Tento krok je nutné udělat ihned po spuštění aplikace, jelikož při změně lokalizace dochází k restartu aplikace. Pokud bychom změnili lokalizaci v průběhu práce s aplikací, byla by veškerá nahraná data ztracena.

Pro ilustraci je na Obr. P2 vyobrazeno uživatelské rozhraní v anglické lokalizaci v průběhu experimentu.



Obrázek P2 - Grafické rozhraní aplikace v angličtině

Popis ovládacích prvků:

- Cesta k souboru - pole, do něhož můžeme ručně napsat cestu k souboru se vstupními daty a nebo se do něj cesta k tomuto souboru automaticky doplní po jeho vyhledání v souborovém prohlížeči dostupném přes tlačítko Procházet
- Procházet - po stisku tohoto tlačítka se objeví okno se souborovým prohlížečem, ve kterém můžeme vyhledat soubor se vstupními daty
- Načíst data - načte data ze souboru k němuž vede cesta uvedená v poli Cesta k souboru
- Modelovat - provede výpočet na základě navrhnutého modelu důvěry a výsledky uloží do výstupního textového a XML souboru
- Načtená data - zobrazí okno s výpisem dat načtených ze vstupního souboru
- Změnit data - umožňuje změnu načtených dat, po stisku tohoto tlačítka se objeví okno požadující jméno měněného aktéra. Pokud zadaný aktér existuje, zobrazí okno s poli pro změnu dat daného aktéra, pokud neexistuje, vypíše chybovou hlášku

- Vyčistit log - vymaže veškerá data v logu

Menu aplikace obsahuje tři podmenu, Program, Upravit a Nápověda. V podmenu Program nalezneme položky se stejnou funkcionalitou, jakou mají tlačítka rozhraní, navíc zde nalezneme i položku End, po jejímž stisku se aplikace ukončí.

Podmenu Upravit obsahuje položku Jazyk, která nám umožňuje změnit lokalizaci aplikace z české na anglickou a zpět. Druhou položkou tohoto podmenu je položka Vyčistit log, jejíž funkčnost jsme si již popsali.

Poslední podmenu Nápověda pak obsahuje dvě položky. Položka Nápověda zobrazí stručnou nápovědu k ovládání a funkcionalitě programu, pod položkou 0 programu se pak skrývají informace o verzi programu a jeho autorovi.