

Technická univerzita v Košiciach
Fakulta elektrotechniky a informatiky
Katedra kybernetiky a umelej inteligencie

Ing. Viliam Ročkai

Konceptuálne znalostné siete

Tvorba a využitie

Písomná práca k dizertačnej skúške

Študijný program: Umelá inteligencia

Študijný odbor: Umelá inteligencia

Školiteľ: doc. Ing. Marián Mach, CSc.

Forma štúdia: externá

Košice 2008

PodĎakovanie

Touto cestou by som sa chcel poďakovať môjmu vedúcemu, pánovi doc. Ing. Mariánovi Machovi, CSc. za cenné rady a pripomienky k mojim školským aktivitám.

Predhovor

*„Väčšina ľudí si myslí, že intelekt robí veľkého vedca. Nemajú pravdu. Robí to charakter.“
(Albert Einstein)*

K výberu tejto minimovej práce som sa rozhodol na základe záujmu hlbšie preniknúť do oblasti konceptuálneho modelovania, ktorú som načal už počas inžinierskeho štúdia. Silnou motiváciou mi bola taktiež myšlienka, že môžem sám prispieť do rozvoja oblasti danej problematiky.

Obsah

Úvod	1
1 Konceptuálne znalostné siete	2
1.1 Definicionálne siete	4
1.2 Rozhodovacie siete	5
1.2.1 Existencionálne grafy	6
1.2.2 Konceptuálne grafy	6
1.2.3 SNePS	7
1.3 Implikačné siete	10
1.3.1 Bayesovské siete	10
1.4 Spustiteľné siete	11
1.4.1 Diagramy tokov údajov	12
1.4.2 Petriho siete	13
1.5 Učiace sa siete	15
1.5.1 Neurónové siete	15
1.6 Hybridné siete	16
1.6.1 UML	17
1.6.2 Sociálne siete	19
2 Vybrané konceptuálne znalostné siete	21
2.1 Existencionálne grafy	21
2.1.1 Alfa časť	21
2.1.2 Beta časť	23
2.1.3 Gama časť	25
2.2 Konceptuálne grafy	26
2.2.1 Definícia Konceptuálneho grafu	26
2.2.2 Koncept	28
2.2.3 Konceptuálna relácia	28

2.2.4	Výraz lambda	29
2.2.5	Typ konceptu	29
2.2.6	Typ relácie	30
2.2.7	Referent	31
2.2.8	Kontext	32
2.3	Ontológie	33
2.3.1	Definícia ontológie	34
2.3.2	Wordnet	35
2.3.3	CYC	37
3	Tvorba a využitie konceptuálnych znalostných sietí	39
3.1	Dolovanie bayesovských sietí z textov	39
3.2	Prístupy tvorby sociálnych sietí	41
3.2.1	Formálna fuzzy analýza	41
3.2.2	Identifikácia tématických sociálnych sietí	42
3.3	Dolovanie zdrojového kódu	42
3.4	Generovanie hierarchií konceptov pomocou FKA	45
3.5	Asociatívne učenie pojmov	46
3.5.1	Počítanie podobnosti slov	48
3.5.2	Výpočet konsezu	48
3.5.3	Vytváranie synonym	50
4	Tézy dizertačnej práce	52
	Zoznam použitej literatúry	53

Zoznam obrázkov

1 – 1	Porfýriov strom	5
1 – 2	KL-ONE	6
1 – 3	Konceptuálny graf znázorňujúci koncertujúcu kapelu	8
1 – 4	SNePS: Marcel vie zahrať skladbu X na gitare	9
1 – 5	Príklad bayesovskej siete	11
1 – 6	Diagram toku údajov	13
1 – 7	Petriho sieť zobrazujúca životný cyklus kapely	14
1 – 8	Znázornenie neurónovej siete	17
1 – 9	Sociálna sieť	20
2 – 1	EG: Marcel hrá na gitare	21
2 – 2	EG: Marcel hrá na gitare, hrá v kapele a živí sa hudbou	22
2 – 3	EG: Marcel nehrá na gitare	22
2 – 4	EG: Marcel hrá na gitare a v kapele, teda sa živí hudbou	22
2 – 5	EG: Marcel v kapele	23
2 – 6	EG: Marcel vie hrať skladbu na gitare	24
2 – 7	EG: Elektro-akustická gitara	24
2 – 8	EG: Ukážka pretínania rezov	24
2 – 9	EG: Dvaja Marcelovia	24
2 – 10	EG: Je želaným faktom, aby sa Marcel živil hudbou	26
2 – 11	EG: Zložitejší EG	26
2 – 12	EG: Zložitejší EG	27
3 – 1	Bayesovská sieť získaná z textu	40
3 – 2	Sociálna sieť zobrazujúca osobné vzťahy medzi spolužiakmi	41
3 – 3	Sociálna sieť vygenerovaná nad doménou „linux“	43
3 – 4	Reprezentácia jednoduchšej funkcie jazyka C vo forme KG	44
3 – 5	Ukážka výstupu FKA zhlukovacieho algoritmu (vľavo) a jeho úpravy do taxonómie konceptov (vpravo)	45

3–6 Učenie v kontextovom okne	47
3–7 Ukážka hierarchickej štruktúry vygenerovanej pomocou AUP	49

Zoznam použitých skratiek

AUP Asociatívne učenie pojmov.

BS Bayesovská sieť.⁶

DTU Diagram toku údajov.

EG Existencionálny graf.

FKA Fuzzy konceptuálna analýza.

KG Konceptuálny graf.

NS Neurónová sieť.

OOP Objektovo orientované programovanie.

UI Umelá inteligencia.

UML z ang. „Unified modeling language“.

SNePS z ang. „Semantic network processing system“.

ZS Znalostná sieť.

Úvod

„Múdry je ten, kto pozná potrebné veci, nie veľa vecí.“ (Aischylos)

Aj napriek tomu, že Aischylos tento výrok vyslovil už dávnejšie, ako vojak z povolania asi vedel o čom hovorí. Je iróniou osudu, že autor tejto definície múdrosti sa nakoniec preslávil písaním tragédií. Potreba vybrať si z množstva informácií, ktoré sa k človeku dostanú, teda nebude len módnym výstrelkom dnešnej doby. Dobré si vybrať hrá v živote oveľa väčšiu rolu, ako by sa mohlo zdať. Dobrý fotograf nie je ten, kto fotí iba dobré snímky, ale ten, kto vie z kvanta fotografií vybrať tie najkvalitnejšie a svojimi nepodarkami spoločnosť neobťažuje. Rovnako dobrí hudobníci alebo matematici majú za sebou kopec neúspešných pokusov. Nakoniec, je všeobecne známe, aké dôležité je dobre si vedieť vybrať partnera.

V dnešnej dobe nie je aktuálnosť extrakcie relevantných znalostí o nič aktuálnejšia ako za čias Aischyla. Zmenila sa však klíma. Znalosti sa uchovávajú a prenášajú v neprehľadných množstvách v elektronickej podobe. S postupnou automatizáciou procesu ich uchovávania, prenosu a využívania, prichádza aj potreba zautomatizovania ich „pochopenia“. Pri každej z týchto činností je dôležité vyriešiť otázku ich reprezentácie.

Táto práca sa pokúsi priblížiť súčasný stav problematiky reprezentácie znalostí v podobe znalostných sietí, ktoré pri spracovaní počítačmi v dnešnej dobe nachádzajú široké uplatnenie. V jednotlivých kapitolách popisuje základné delenie znalostných sietí, bližšie sa venuje vybraným znalostným sieťam a na záver uvádza pár príkladov ich využitia v praxi. Samotná práca môže poslúžiť ako príklad toho, že orientovať sa v dnešom svete preplnenom informáciami je ťažké, a vybrať z nich relevantné zdroje nie je jednoduché ani pre človeka samotného. Pre zvedavého čitateľa môže poslúžiť ako odrazový mostík do danej tematiky. V neposlednom rade sa po zverejnení stane táto práca len ďalším uzlom v konceptuálnej znalostnej sieti dokumentov zaoberajúcich sa problematikou reprezentácie znalostí.

1 Konceptuálne znalostné siete

Konceptuálne znalostné siete možno chápať ako grafovú reprezentáciu znalostí a je veľmi často využívanou formou reprezentácie. V závislosti od typu siete môžu byť vrcholmi grafu koncepty, poprípade celá sieť môže vďaka svojej vnútornej štruktúre vytvárať samostatný koncept na svojom výstupe.

Konceptuálne znalostné siete môžeme chápať aj ako sémantické siete. Sémantická sieť v úzkom slova zmysle je orientovaným grafom pozostávajúcim z vrcholov, ktoré reprezentujú koncepty a hrán, ktoré reprezentujú relácie medzi danými konceptami. Koncepty (resp. pojmy) sú viac alebo menej abstraktné konštrukty, pomocou ktorých sa buduje model relevantnej časti sveta. V širšom slova zmysle sa dá chápať ako sieť, ktorá obsahuje koncepty (napríklad neurónová sieť na svojom výstupe).

Rozoznáva sa niekoľko typov konceptov podľa ich charakteristiky. Sú to najmä *triedy*, *relácie*, *funkcie*, *procedúry*, *objekty*, *premenné* a *konštanty*. Tvoria zväčša komplikovanú sieťovú štruktúru – doménový model danej problémovej oblasti.

Je potrebné zdôrazniť, že nech je doménový model akokoľvek rozsiahly, nemôže si nárokovať univerzálnu platnosť. Vždy modeluje svet iba z určitého hľadiska. Ak majú byť koncepty použité zmysluplne, sú vo veľkej väčšine závislé od rámca tej-ktorej oblasti, domény. Inými slovami, pojmy a vzťahy podstatné napríklad pre poisťovaciu spoločnosť sa takmer určite nebudú dať použiť povedzme pre vzdelávaciu inštitúciu (hoci je možné, že istá časť modelu môže byť aj v tomto prípade spoločná pre obe oblasti) [15].

Medzi dôležité sémantické relácie patria napríklad:

- meronymá – A je časťou B
- holonymá – B je časťou A
- hyponymá - A je podriadený B
- hypernymá – A je nadriadený B
- synonymá - A označuje to isté ako B

- **antonymá** – A označuje presný opak toho čo B

Ako klasický príklad sémantických sietí býva často označovaný lexikón anglického jazyka – Wordnet. Sémantické siete vytvoril Richard H. Richens z Cambirdgskej univerzity v roku 1956 [40] a mali slúžiť ako medzijazyk pri výpočtovom preklade (machine translation).

Podľa [52] je sémantickou sieťou grafický zápis reprezentácie znalostí v podobe prepojených uzlov a hrán. Aj napriek tomu, že Sowa to vo svojej práci nedefinuje explicitne, znalostné štruktúry, ktoré nazýva sémantickými sieťami, majú stále nejaké priame napojenie na koncepty. Buď sú koncepty prvkami grafu, alebo sú generované na výstupe grafu. Z nášho pohľadu tvoria sémantické siete podľa Sowa konceptuálne znalostné štruktúry - konceptuálne znalostné siete.

Sémantické siete delí na 6 základných typov:

- **Definicionálne siete** zvyčajne zdôrazňujú reláciu podtypu taktiež známou ako is-a reláciu. Nazývajú sa sieťami generalizačnými alebo sieťami subsumpcnej hierarchie. Podporujú dedenie, a to tak, že vlastnosti definované pre nadtyp sa automaticky propagujú do všetkých jeho podtypov – prvkov, ktoré od neho dedia. Vzhľadom na to, že definície sú už zo svojej definície pravdivé, považujeme aj informáciu obsiahnutú v týchto sieťach automaticky za pravdivú.
- **Rozhodovacie siete** sú navrhnuté na prácu s výrokmi. Na rozdiel od definicionálnych sietí pokladáme informáciu v nich obsiahnutú za platnú len v prípade, že je tak explicitne označená modálnym operátorom. Niektoré asociačné siete boli navrhnuté ako modely konceptuálnych štruktúr tak, aby odzrkadľovali sémantiku prirodzeného jazyka.
- **Implikačné siete** používajú na prepájanie vrcholov ako primárnu reláciu reláciu implikácie. Bývajú používané ako vzory pre domienky, kauzalitu alebo inferencie.
- **Spustiteľné siete** v sebe zahŕňajú mechanizmus, napríklad „marker passing“ alebo pripojené procedúry, ktoré vedú vykonávať inferencie, posielat správy alebo vyhľadávať vzory a asociácie.

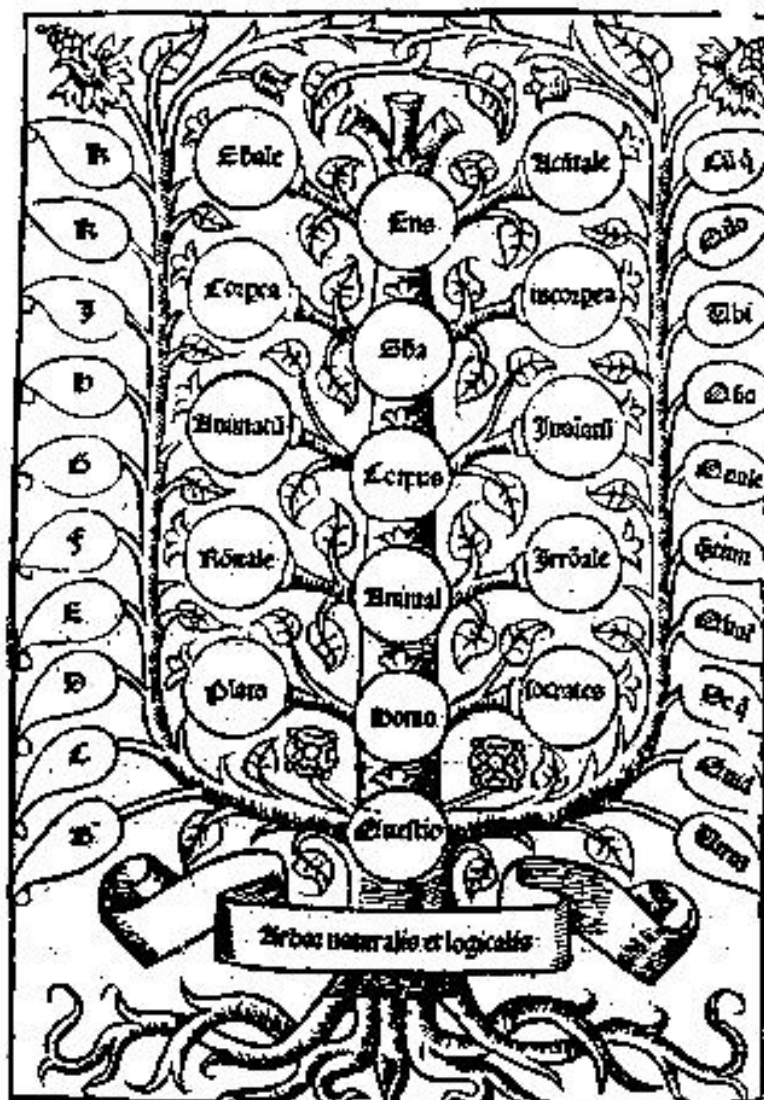
- **Učiace sa siete** stavajú, poprípade rozširujú svoju vlastnú reprezentáciu získavaním nových znalostí z príkladov. Nové znalosti môžu zmeniť pôvodnú sieť tým, že pridajú, respektíve odoberú nejaké jej uzly, poprípade menia číselné hodnoty, nazývané tiež váhy, ktoré vyjadrujú vzťahy medzi uzlami, a tak kvantitatívne označujú ich relácie.
- **Hybridné siete** kombinujú dva alebo viac vyššie spomenutých prístupov. Bežne sa nemusíme stretnúť len so spojením týchto prístupov v jednej sieti. V praxi sa často využíva aj blízke prepojenie viacerých sietí odlišných typov.

1.1 Definicionálne siete

Definicionálne siete sú štruktúry veľmi blízke programátorom objektovo orientovaných jazykov, keďže vyjadrujú jednu zo základných vlastností OOP, a to dedenie. Medzi typický príklad definicionálnych sietí sa zvyčajne predkladá Porfýriov strom zobrazovaný na obrázku 1 – 1.

Obrázok č. 1 – 1 [59] aj napriek svojmu vysokému veku (tento strom vznikol už s tretom storočím nášho letopočtu) má všetky potrebné vlastnosti, ktoré sa dnes používajú pri definovaní typov konceptov. Slúžil na zobrazenie Aristotelovej metódy na definíciu kategórií špecifikovaním rodu a kategórií. Na ilustráciu modernejšieho systému môžeme spomenúť napríklad deskriptívnu logiku, ktorá rozširuje vlastnosti Porfýriovho stromu. Deskriptívna logika je odvodená od prístupu predstavenom Woodsom v roku 1975 a implementovaným Brachmanom v roku 1979 ako systém Knowledge Language One (KL-ONE) [2]. Príklad systému je na obr. 1 – 2.

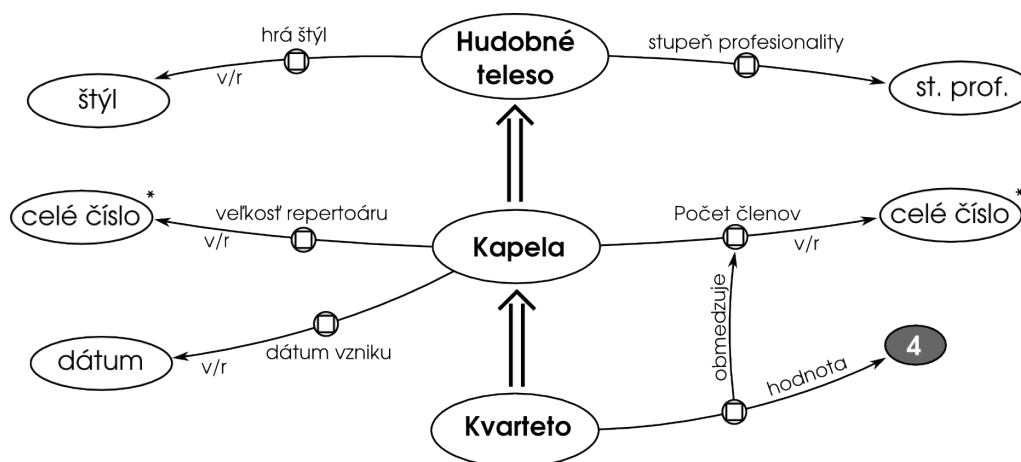
Na obr. 1 – 2 je v systéme KL-ONE definované hudobné teleso - *kvartero*. Kvartero dedí svoje vlastnosti z konceptu *kapela*, ktorý dedí opäť vlastnosti z konceptu *hudobného telesa*. Každý z konceptov má vlastné vlastnosti a ich obmedzenia ako napríklad *počet členov* alebo *dátum vzniku* a tieto vlastnosti sa propagujú do ich podtypov.



Obr. 1–1 Porfýriov strom

1.2 Rozhodovacie siete

Rozhodovacie siete sú navrhnuté na prácu s výrokmi, napríklad ich definovanie, dokazovanie a odvodzovanie. Vzhľadom na ich schopnosť dobre reflektovať znalosti z reálneho sveta bude ich vybraným typom venovaná v tejto práci osobitná kapitola. V tejto kapitole bude poskytnuté iba prvotné priblíženie



Obr. 1 – 2 KL-ONE

1.2.1 Existencionálne grafy

Koncom 19. storočia boli vytvorené 2 dôležité grafické systémy určené pre matematickú logiku. Prvým z nich bol Fregeho „Begriffsschrift“ [12]. Myšlienky charakteristické pre tento systém mali nepochybne silný vplyv na matematiku, no systém samotný sa pravdepodobne pre svoju zložitosť tlače do praxe nedostal.

Druhým grafickým systémom boli Peirceove existenciálne grafy (EG). Tie vznikli úplne na konci 19. storočia, keď v logike už držala prvenstvo symbolická logika. A tak aj napriek tomu, že sa Peirce domnieval, že existencionálne grafy sú oveľa lepším prístupom formálnej logiky ako symbolická logika, tá získala vo formálnej logike prvenstvo. Je to iróniou osudu, keďže symbolická logika [4] stavala na základoch Peircovho algebraického zápisu, ktorý bol v tom čase už skoro 10 rokov starý.

1.2.2 Konceptuálne grafy

Konceptuálne modely sa vyskytujú hlavne v oblasti návrhu softvéru na podporu mentálnych modelov užívateľa. Mayer [25] preštudoval niekoľko konceptuálnych modelov z oblastí databázových systémov či fyziky a uvádza, že konkrétny konceptuálny model podporuje konceptuálne a znižuje doslovné pamätanie si informácií, čo zlepšuje riešenie

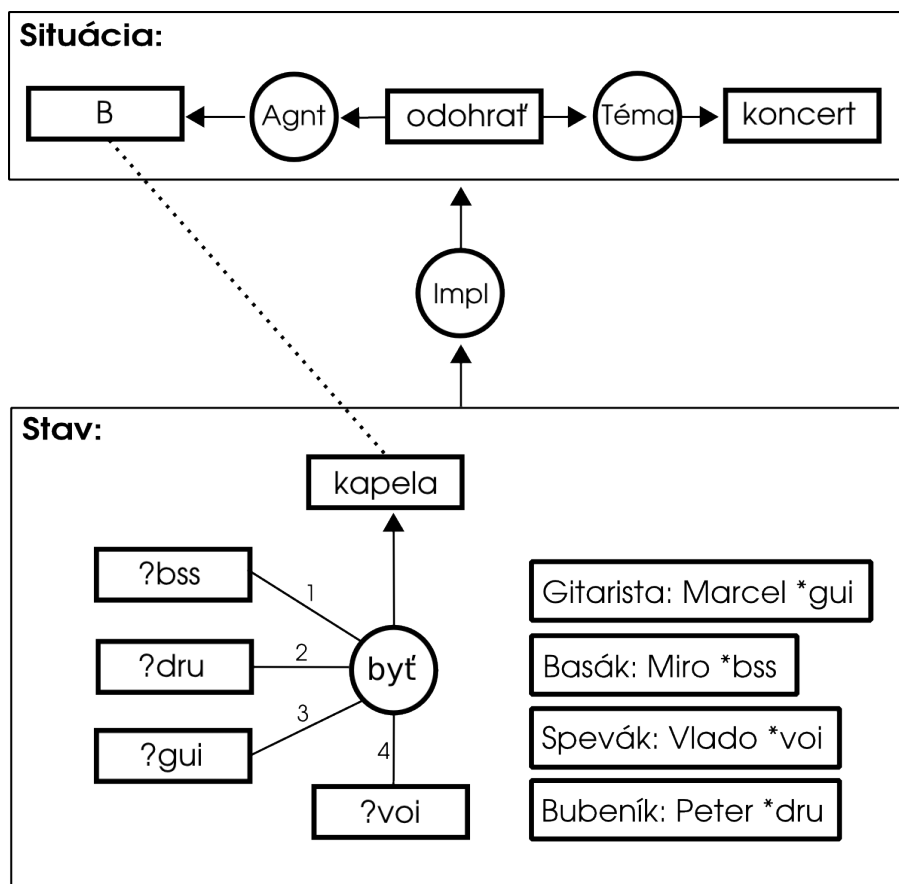
problémov. Teda znázornenie konceptuálnych vzťahov poskytuje človeku ďaleko rýchlejšie pochopenie problematiky. Existencionálne grafy slúžili neskôr ako podklad pre vznik konceptuálnych grafov, ktoré sa pokladajú za ich rozšírenie a vznikli ich spojením so sémantickými sieťami. Konceptuálne grafy sa prvýkrát spomínajú v publikácii od John F. Sowa [51], kde ich používal na reprezentáciu konceptuálnych schém databázových systémov. Vyjadrovanie pomocou konceptuálnych grafov je logicky presné, pre človeka ľahko čitateľné a vhodné na spracovanie počítačmi. Vzhľadom na ich priame prepojenie s prirodzeným jazykom by sa dali považovať za medzivrstvový jazyk na prekladanie medzi formálnymi počítačovými jazykmi a prirodzeným jazykom. Ich grafická reprezentácia ich robí veľmi dobre čitateľnými.

Príklad konceptuálneho grafu môžeme vidieť na obr. 1 – 3 a jeho vlastnosťami sa budeme bližšie zaoberať v jednotlivých definíciách jeho častí. Znázorňuje koncertujúcu kapelu o veľkosti štyroch konkrétnych členov. Vykresľuje stav, v ktorom definuje štyroch členov: gitaristu, basáka, speváka a bubeníka, a následne ich zoskupuje do konceptu kapely. Znak „*“, ktorý určuje definujúcu značku a „?“, ktorý sa na túto definíciu odkazuje budú popísané bližšie v osobitnej kapitole. Zároveň uvádza danú kapelu do situácie, kde odohráva koncert.

1.2.3 SNePS

Semantic Network Processing System (SNePS) bol navrhnutý ako systém na prezentáciu myšlienok v prirodzenom jazyku s pomocou inteligentných systémov (kognitívnych agentov). Zámerom bolo, aby sa znalostné bázy postavené na SNePS nebudovali programátorom alebo znalostným inžinierom v nejakom formálnom jazyku, lež bežným človekom, ktorý by znalosti zadával v jazyku prirodzenom (pôvodne v angličtine) či systémom čítajúcim knihy a články, ktoré boli pripravené pre bežného čitateľa. Vzhľadom na túto motiváciu boli kritériá na vývoj SNePSu nasledujúce:

- musí umožňovať vyjadriť čokoľvek vyjadriteľné v prirodzenom jazyku,

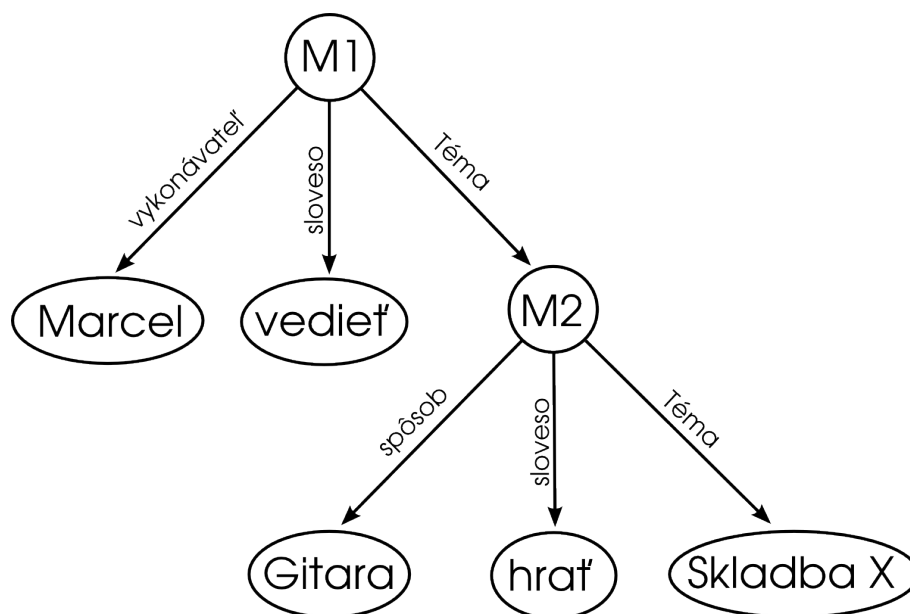


Obr. 1–3 Konceptuálny graf znázorňujúci koncertujúcu kapelu

- musí umožňovať vyjadrenie všeobecných i špecifických informácií,
- musí byť schopný vedieť tieto informácie ďalej používať pri rozhodovaní a odvodzovať informácie vzhľadom na to, čo už bolo povedané,
- nemôže očakávať, že dané informácie budú nejakým spôsobom usporiadané,
- musí sa chovať korektne aj v prípade, že daná informácia obsahuje cyklické definície, rekurzívne pravidlá alebo nekonzistentné informácie.

SNePS je rozhodovacia sémantická sieť pozostávajúca z uzlov a označených orientovaných hrán. Uzly tu predstavujú výroky. Uzly bez napojených hrán sa nazývajú základnými uzlami a používajú sa na vyjadrenie nesporných entít. Uzly, na ktoré sú napojené hrany sa nazývajú molekulárne uzly a sú štrukturálne definované štruktúrou siete tvorenou nasledo-

vaním daných hrán. Žiadna dvojica uzlov nemože mať tieto štruktúry rovnaké. Sémantika uzla (to, čo reprezentuje) je štruktúrálné určená štruktúrou, ktorá z neho vystupuje a z hľadiska tvrdenia je určená sieťovou štruktúrou, s ktorou je prepojená cez hrany, ktoré na ňu ukazujú. Štruktúrálna informácia spojená s uzlom sa meniť môže, no informácia z hľadiska tvrdenia nie. Uzly reprezentujú všetky entity, o ktorých môže namodelovaný kognitívny agent uvažovať vrátane individuí, tried, vlastností, udalostí, akcií, výrokov, pravidiel atď. Všetky uzly sú taktiež termami formalizmu SNePS. To znamená, že môžu byť v "argumetačnej" pozícii k iným uzlom. Tak umožňuje SNePS reprezentáciu výrokov o výrokoch alebo pravidiel o pravidlách bez obmedzení [48].



Obr. 1–4 SNePS: Marcel vie zahrať skladbu X na gitare

Na obr. 1–4 vidíme SNePS sieť zobrazujúcu výrok: „Marcel vie zahrať skladbu X na gitare“.

Z praktického využitia SNePS stojí za pozornosť jeho využitie ako medzijazyka pri strojovom preklade z čínštiny do angličtiny [27] alebo využitie ako nástroja na vyhodnocovanie ľudských emócií zo zápisov dialógov [39].

1.3 Implikačné siete

Implikačné siete používajú ako základnú hranu relácie implikáciu. Vzhľadom na interpretáciu hrán v daných sieťach sa môžu nazývať:

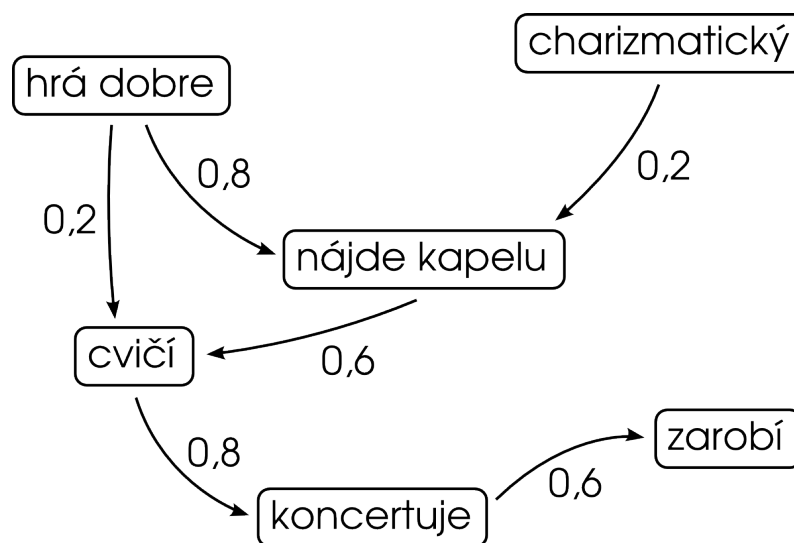
- Domienkové siete (belief networks),
- Kazuálne siete (casual networks),
- Bayesovské siete (bayesian networks),
- Truth-maintenance systémy (truth-maintenance systems [10]).

1.3.1 Bayesovské siete

Pravdepodobne najznámejším typom implikačných sietí sú bayesovské siete [34](BS). Predstavujú zväzok teórie pravdepodobnosti a teórie grafov, pričom poskytujú prirodzený nástroj na riešenie dvoch základných problémov vyskytujúcich sa v aplikovanej matematike a inžinierskych disciplínach – problémom neurčitosti a zložitosti.

Teória grafov poskytuje teoretické zázemie na intuitívne modelovanie interaktívnych súborov premenných a štruktúr údajov s využitím modularity. Teória pravdepodobnosti poskytuje spôsob ich previazania tak, aby bol celok konzistentný a modely boli prepojené s údajmi. Aktualizácia údajov v bayesovských sieťach na základe pozorovaní sa uskutočňuje aplikovaním Bayesovej teórey [18].

Uzly v BS predstavujú premenné predpokladaného záujmu (výsledok testu, teplota vzduchu alebo poruchovosť auta) a hrany predstavujú informačné alebo kauzálne závislosti medzi nimi. Závislosti sú kvantifikované podmienenými pravdepodobnosťami pre každý uzol, ktorý sa takto prepája so svojimi rodičovskými uzlami. Jedným z dôležitých aspektov BS je, že predstavujú priamu reprezentáciu sveta a nie procesov rozhodovania. Rozhodovací proces môže prebiehať propagáciou informácie, v ktoromkoľvek smere (nie len po smere hrán) [33]. Známe sú aj rozšírenia BS, napríklad dynamické BS [9] alebo hierarchické BS [21].



Obr. 1 – 5 Príklad bayesovskej siete

Na obr. 1 – 5 vidíme jednoduchú BS, ktorá nepriamo vyjadruje potrebu predpokladov na to, aby si človek mohol zarábať hudbou. V obdĺžnikoch sú znázornené stavy, ktoré sú navzájom prepojené orientovanými hranami. Ku každej hrane je priradené číslo, ktoré vyjadruje pravdepodobnosť prechodu z jedného stavu do druhého. Napríklad ak človek ovláda dobre hru na svoj nástroj, je dosť pravdepodobné (20%), že svoj talent ďalej podporuje cvičením, ale je oveľa pravdepodobnejšie (80%), že si nájde kapelu a až táto skutočnosť ho donúti pravidelne cvičiť.

Doménové spektrum využitia BS je pomerne široké. Svoje uplatnenie si nachádzajú napríklad v medicíne ako nástroje na hľadanie interakcií medzi génmi [13]. V počítačovom priemysle sa dajú využívať aj na diagnózu porúch TCP/IP spojení [24]. Už len stručný prehľad využitia BS by si vyžadoval priestor osobitnej práce.

1.4 Spustiteľné siete

Spustiteľné sémantické siete obsahujú mechanizmus, ktorý umožňuje im samotným na sebe vykonávať zmeny. Týmto sa odlišujú od sietí, ktoré slúžia len ako statické dátové štruktúry a ich obsah je manipulovateľný skôr akciami zvonka.

Poznáme tri základné mechanizmy, ktoré sa k tomuto účelu využívajú v spustiteľných sieťach.

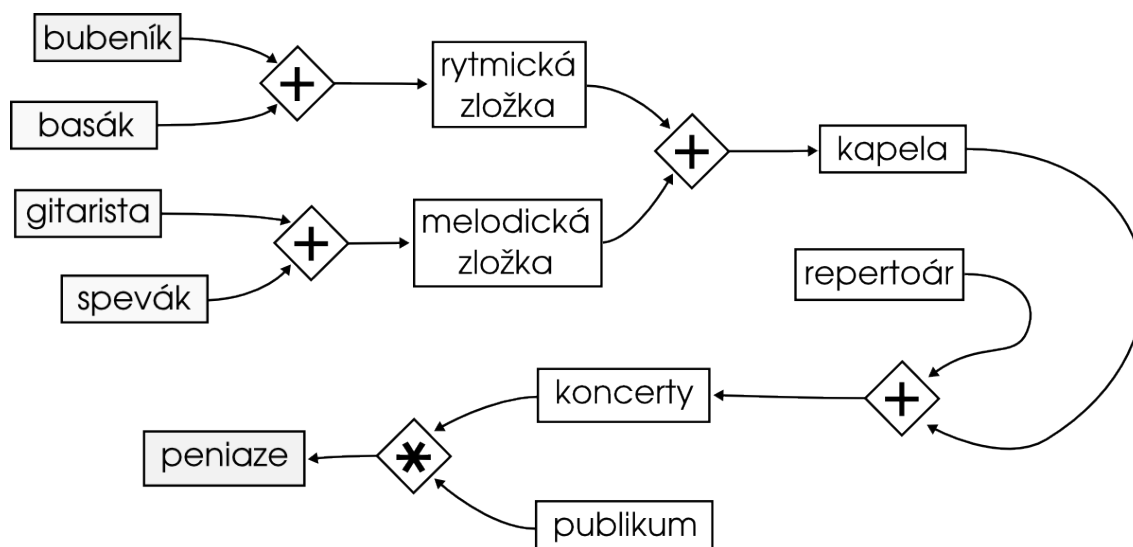
1. Posielanie správ - siete sú schopné posilať správy z jedného uzla do druhého. U niektorých sietí môže správu reprezentovať jeden bit, ktorý sa nazýva značka, token alebo trigger. Správu môže reprezentovať aj numerická váha alebo akákoľvek iná ľubovoľne veľká správa.
2. Pripojené procedúry - sú programy obsiahnuté, resp. asociované v uzloch siete. Zväčša vykonávajú nejakú akciu, prípadne výpočet nad daným uzlom, resp. jemu blízkym uzlom.
3. Transformácia grafu - kombinuje grafy, mení ich alebo ich dekomponuje na menšie celky. V typických dokazovačoch teorém sa tieto procedúry vykonávajú externým programom. Ak sú spúšťané grafom samotným, tak sa správajú ako chemické reakcie, ktoré spájajú alebo rozbíjajú molekuly.

Tieto prístupy sa môžu navzájom kombinovať. Správy poslané z jedného uzla do druhého môžu byť taktiež spracované procedúrami pripojenými na tieto uzly a podobne aj grafové transformácie môžu byť vyvolané prijatím správy do konkrétneho uzla.

1.4.1 Diagramy tokov údajov

Existuje viacero druhov diagramov toku údajov (DTU) [23]. V podstate sa jedná o grafy, ktoré vyjadrujú tok dát v informačných systémoch. Zvyčajne obsahujú dva typy uzlov. Pasívne uzly obsahujú dáta nejakého typu a aktívne uzly tieto dáta na svojom vstupe prijímajú a na výstupe poskytujú nové. Ich sémantika je podrobnejšie popísaná v [3].

Na obr. 1 – 6 vidíme reprezentáciu toku udalostí, ktorý musí nastať od založenia kapely až po začiatok jej zárobkovej činnosti. Príklad je len ilustračný a operátory boli vybrané skôr symbolicky ako najbližšie analógie. Obdĺžnikové uzly predstavujú uzly pasívne a uzly v tvare štvorca otočeného o 45 stupňov označujú aktívne uzly. Kým operátor + znázorňuje



Obr. 1–6 Diagram toku údajov

nejaké spojenie dvoch dátových typov a vytvorenie nového, operátor * znázorňuje priamu úmeru (čím väčšie publikum na koncerte, tým viac peňazí interpret dostane).

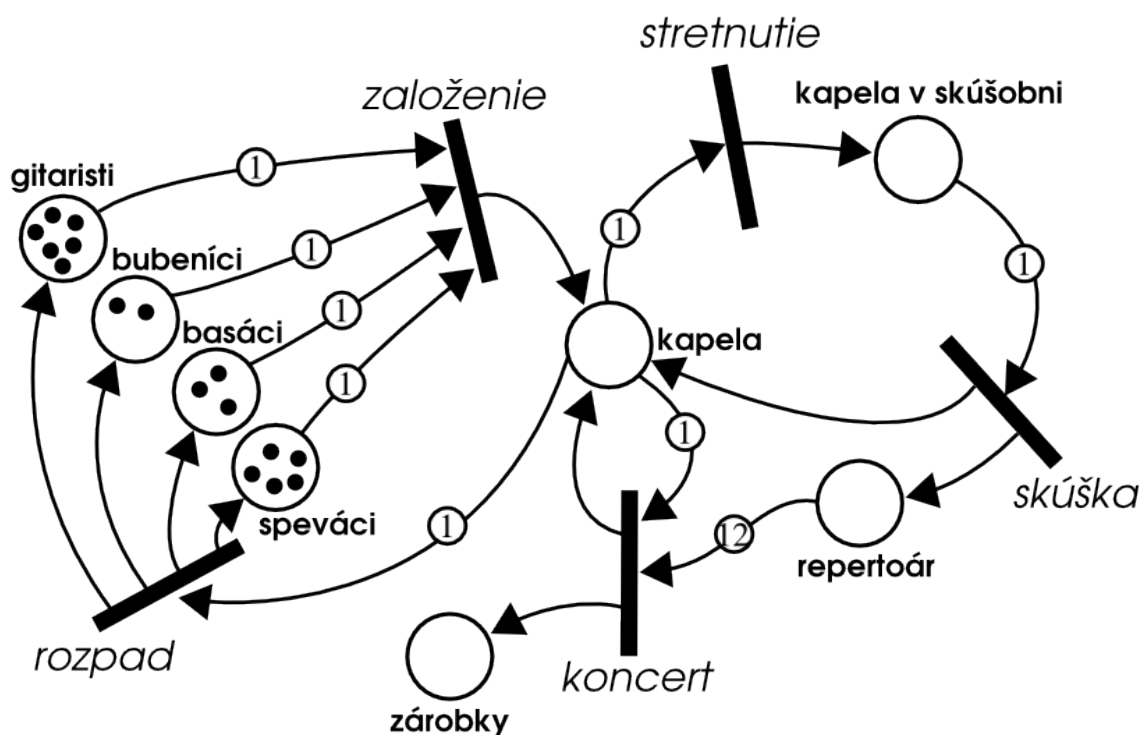
DTU sa využívajú hlavne ako modelovací nástroj tam, kde je zameranie na sekvenčné aspekty modelu potrebnéjšie ako zameranie na štruktúrny model [55].

1.4.2 Petriho siete

Petriho siete boli vytvorená pánom C. A. Petrim v roku 1962 [37], kde boli v rámci dizertačnej práce predstavené ako nový model informačného toku v systémoch. Majú uzly dvoch druhov: miesta a zmeny. Každý uzol musí spadať presne do jedného z týchto dvoch typov. Orientované hrany sú v tomto prípade reláciami toku a môžu existovať len medzi uzlami odlišného druhu. Uzly typu „miesta“ môžu obsahovať rozličný počet značiek, tzv. tokenov a relácie môžu byť rôzne ováňované. Samotné správanie sa systému potom závisí od prechodu tokenov cez relácie a deje sa vďaka uzlom typu zmena,“ [30]. Základné vlastnosti takéhoto systému by sa dali v skratke zhrnúť v troch bodoch:

1. Tranzícia **t** je aktívna, ak každý uzol, ktorý do nej vstupuje cez nejakú reláciu, obsahuje aspoň toľko tokenov, ako je veľkosť váhy danej relácie.

2. Aktívna tranzícia potom môže, ale nemusí vyvolať akciu.
3. Ak uzol vyvolá akciu, je z každého uzla, ktorý do tranzície vstupuje nejakou re-
láxiou odobraný počet tokenov definovaný jej váhou a na každý výstupný uzol je
analogicky pridaný počet tokenov definovaný reláciou, ktorá daný token prepája s
tranzíciou.



Obr. 1–7 Petriho sieť zobrazujúca životný cyklus kapely

Na obr. 1–7 vidíme Petriho sieť vyjadrujúcu kolobeh života kapely. Tokeny z množín hráčov na rôzne hudobné nástroje po založení vytvoria kapelu, skúšajú skladby a po naskúšaní ich dostatočného počtu (12) začínajú koncertovať, až kým sa kapela nerozpadne. Petriho siete možno využiť na štúdium vlastností a dynamiky biologických systémov [35], analýzu výkonnosti programovacieho jazyka Java [38] alebo plánovanie úloh [5].

1.5 Učiace sa siete

Učiace sa systémy, či už prirodzený alebo umelý, reaguje na nové informácie zmenou svojej vnútornej reprezentácie v takom zmysle, aby sa vedel lepšie adaptovať svojmu okoliu [36]. Systémy, ktoré používajú sieťovú reprezentáciu, ju môžu meniť nasledujúcimi tromi spôsobmi:

1. **Rote memory** - najjednoduchším spôsobom je pretransformovať novú informáciu opäť do podoby siete a túto sieť bez ďalších úprav potom k existujúcej sieťovej reprezentácii pridať.
2. **Zmena váh** - niektoré siete majú k svojim hranám asociované váhy. Napríklad v implikačných sieťach môžu byť hranám určené pravdepodobnosti. Zmenami týchto váh sa potom môže systém lepšie adaptovať k svojmu prostrediu.
3. **Reštrukturalizácia** - najkomplexnejší prístup učenia tvorí silný zásah do samotnej štruktúry siete. Vzhľadom na veľký počet druhov možných zmien je štúdium a klasifikácia reštrukturalizačných metód najzložitejšia, no potencionálne pri nájdení vhodnej metódy, najslubnejšia.

1.5.1 Neurónové siete

Medzi základné siete tohto typu patria neurónové siete. Neurónovú sieť (NN) je možné definovať ako paralelný distribuovaný procesor, ktorý má schopnosť uchovávať poznatky získané skúsenosťou a mať ich k dispozícii na použitie. Podobá sa mozgu v dvoch ohľadoch:

- sieť získava poznatky procesom učenia,
- na uchovanie poznatkov sa používajú medzi-neurónové spojenia známe ako synaptické váhy.

Neurónovú sieť môžeme definovať aj čisto z matematického hľadiska pomocou teórie grafov [20]: Neurónová sieť je štruktúra pre paralelné a distribuované spracovanie informácií v tvare orientovaného grafu s nasledujúcimi pravidlami:

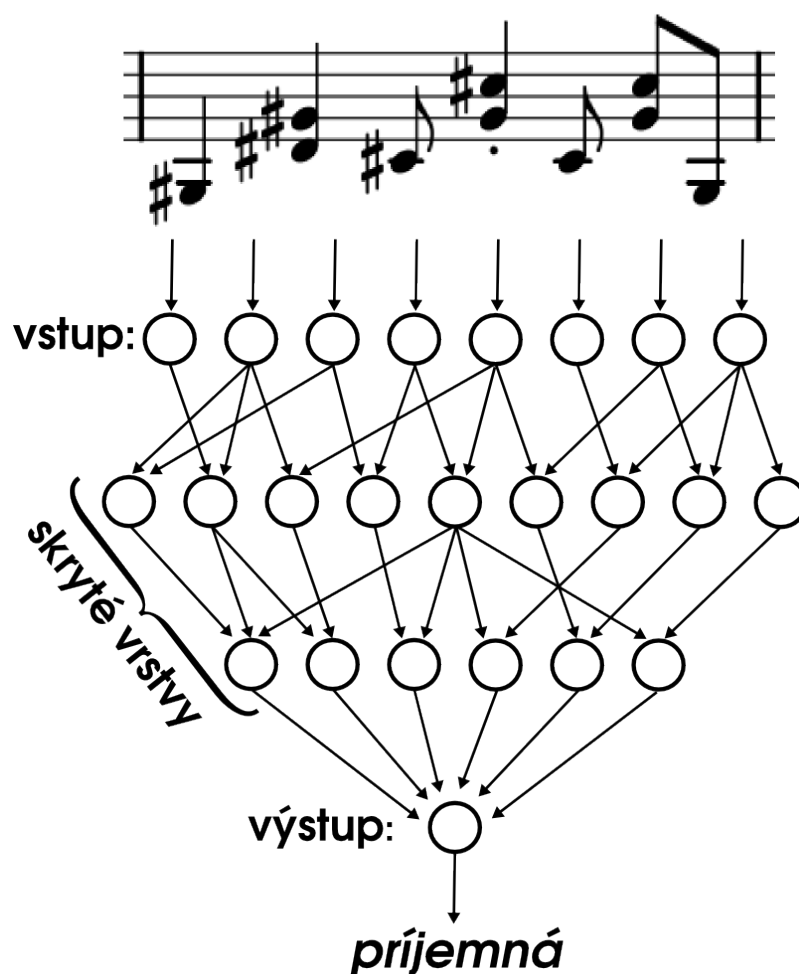
- uzly grafov sa nazývajú výpočtové prvky,
- hrany grafu sa nazývajú spojenia,
- každý výpočtový prvok môže obsahovať ľubovoľný počet vstupných, spojení
- každý výpočtový prvok môže mať ľubovoľný počet výstupných spojení - v skutočnosti sa jedná o ten istý výstup, ktorý sa rozvetvuje,
- výpočtové prvky môžu mať lokálnu pamäť,
- každý výpočtový prvok obsahuje prenosovú (aktivačnú) funkciu, ktorá môže využiť lokálnu pamäť, vstupné signály a ktorá vytvorí výstupný signál výpočtového prvku.

Na obr. 1 – 8 je schématicky znázornená neurónová sieť, ktorá má vyhodnocovať kvalitu hudobnej melódie. Ako vstup slúži samotný úsek melódie a výstup je označenie kvality melódie. Prepokladáme pritom, že systém bol predtým učný na pozitívnych a negatívnych príkladoch, ktoré melódie má pokladať za príjemné a ktoré nie.

Neurónové siete sa používajú hlavne na aproximáciu funkcií, klasifikáciu a spracovanie dát (ako zhlukovanie, filtrovanie, kompresia). Často je možné stretnúť sa s ich využitím pri rozpoznávaní vzorov [42] a rozpoznávaní reči [6]. S prihliadnutím na hudobnú doménu príkladov v tejto práci je dobré spomenúť, že neurónové siete sa môžu využívať aj na kompozíciu hudby [29].

1.6 Hybridné siete

Spojením viacerých z vyššie uvedených princípov pre dané kategórie sémantických sietí dostávame sieť hybridnú. Asi najpoužívanejšou hybridnou znalostnou sieťou je ontológia, spájajúca v sebe definicionálne siete s rozhodovacími sieťami, ktorej je venovaná v tejto



Obr. 1 – 8 Znáznornenie neurónovej siete

práci špeciálna sekcia v kapitole o vybraných znalostných sieťach.

1.6.1 UML

„Unified Modelling Language“ (UML) bola špecifikovaná pánmi Grady Boochom, Ivar Jacobsonom a Jim Rumbaughom. UML bol vytvorený so zámerom pre záznam, konštrukciu, vizualizáciu a dokumentáciu artefaktov systémov s prevažne softvérovou charakteristikou. Definícia UML obsahuje základné časti:

- Definícia notácie UML (syntax)

- Metamodel UML (sémantika)
- Jazyk OCL (Object Constraint Language) pre popis ďalších vlastností modelu
- Špecifikácia konverzie do výmenných formátov (CORBA IDL, XML DTD)

Dokumentácia v UML sa nemusí skladať iba z (viac či menej formálnych) textov. Zo základných lexikálnych elementov môžeme vytvárať dvojrozmerné diagramy rozmiestnením množiny elementov a ich prípadným prepojením spojkami. Teoreticky možno použiť ľubovoľné spojky, z hľadiska unifikácie je však výhodné, pokiaľ zvolíme určitú sadu elementov, ktorá je vhodná pre vyjadrenie nejakého zmysluplného pohľadu na modelovaný systém. UML preto reprezentuje 8 typov diagramov, tzn. 8 rôznych pohľadov na model systému:

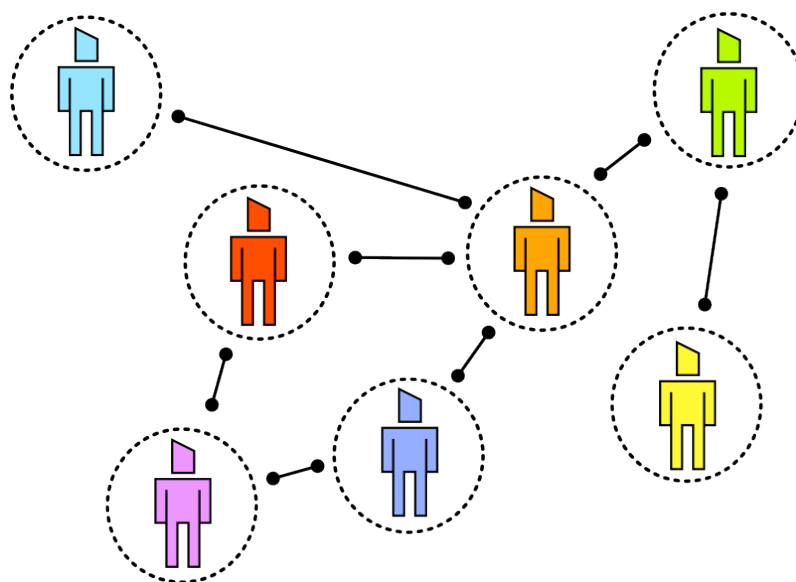
- **Diagramy tried a objektov** popisujú statickú štruktúru systému, znázorňujú datový model systému od konceptuálnej úrovne až po implementáciu.
- **Modely jednania** dokumentujú možné prípady použitia systému - udalosti, na ktoré musí systém reagovať.
- **Scenáre činností** popisujú scenár priebehu určitej činnosti v systéme.
- **Diagramy spolupráce** zachytávajú komunikáciu spolupracujúcich objektov.
- **Stavové diagramy** popisujú dynamické správanie objektu alebo systému, možné stavy a prechody medzi nimi.
- **Diagramy aktivít** popisujú priebeh aktivít a činností.
- **Diagramy komponentov** popisujú rozdelenie výsledného systému na funkčné celky a definujú náplň jednotlivých komponentov.
- **Diagramy nasadenia** popisujú umiestnenie funkčných celkov (komponentov) na výpočtové uzly informačného systému.

Podľa [52] obsahuje UML aj napriek tomu, že sa nepokladá za sémantickú sieť, vo svojich častiach prvky sietí, ktoré boli popísané vyššie. Základom UML je definicionálna sieť na

definovanie typov objektov. Tá v sebe zahŕňa základné vlastnosti Porfýriovho stromu ako typ-podtyp, typ-inštancia atď. UML zahŕňa 2 typy spustiteľných sietí, ktoré by sa dali považovať za špeciálne prípady Petriho sietí, a to stavové diagramy a diagramy aktivít. Ostatné siete v UML by sa mohli považovať za prípady relačných grafov špecializovaných na reprezentáciu metaúrovňovej informácie. Ako príklad takéhoto grafu by mohol poslúžiť diagram znázorňujúci relácie medzi entitami. Jazyk OCL v UML je verziou logiky prvého rádu s notáciou, ktorá sa približuje vlastnostiam objektovo orientovaného programovania.

1.6.2 Sociálne siete

Za špeciálny prípad znalostných sietí môžeme považovať aj sociálne siete. V sociálnej sieti reprezentujú uzly grafu jednotlivé osoby a hrany reprezentujú nejaké vzťahy medzi danými osobami. Príklad takejto siete je znázornený na obr. 1 – 9. Môže sa jednať o príbuzenské, priateľské, obchodné alebo iné vzťahy. Takéto siete majú veľmi široké využitie, od ktorého aj nepriamo závisí typ sémantickej siete. Od zobrazenia vzťahov v kolektíve [22] (definicionálne), cez epidemiologické výskumy [26] až po prieskum tzv. „difúzie inovácie“ [1], kde sa snažíme zistiť, ako rýchlo sa dostane nová myšlienka medzi ostatných účastníkov siete (spustiteľné).



Obr. 1–9 Sociálna sieť

2 Vybrané konceptuálne znalostné siete

V nasledujúcich podkapitolách budú bližšie popísané vybrané druhy znalostných (konceptuálnych sietí): existencionálnych grafov, konceptuálnych grafov a ontológií. Výber týchto konkrétnych typov bol založený na poznatku, že ako konceptuálne grafy [50] (vychádzajúce z existencionálnych grafov), tak ontológie obsahujú vlastnosti potrebné na vyjadrenie znalostí z reálneho sveta.

2.1 Existencionálne grafy

Systém existencionálnych grafov (EG) je elegantný systém logiky, ktorý zahŕňa výrokovú logiku, logiku prvého rádu a tiež určité aspekty logiky vyššieho rádu. Je rozdelený na tri časti: Alfa, Beta a Gama. Tieto tri časti na sebe vzájomne stavajú. Beta je postavená na Alfe a Gama na Bete. Keďže na základoch EG boli postavené aj *konceptuálne grafy*, bližší popis ich vlastností je prínosný.

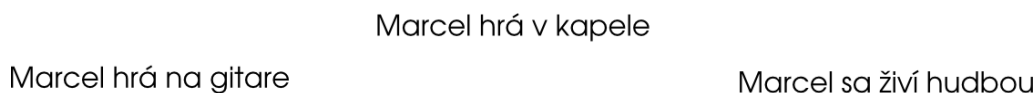
2.1.1 Alfa časť

Alfa EG pozostávajú len z dvoch syntaktických entít: (atomárnych) výrokov a rezov, ktoré sú reprezentované uzatvorenými nepretínajúcimi sa krivkami. Atomárne výroky môžu byť vnímané ako predikátové mená s aritou rovnou nule. Peirce ich nazval medádami. Medády sú napísané na ploche, ktorá sa nazýva listom tvrdenia (assertion). Listom tvrdenia môže byť napríklad list papiera, tabuľa alebo hocijaká iná plocha. Na obr. 2 – 1 sa nachádza EG s významom „Marcel hrá na gitare“. Môžeme taktiež vedľa seba napísať viacero výrokov. To zodpovedá konjunkcii daných výrokov.

Marcel hrá na gitare

Obr. 2 – 1 EG: Marcel hrá na gitare

EG na obr. 2 – 2 by potom znamenal „Marcel hrá na gitare, Marcel hrá v kapele a Marcel sa živí hudbou“.



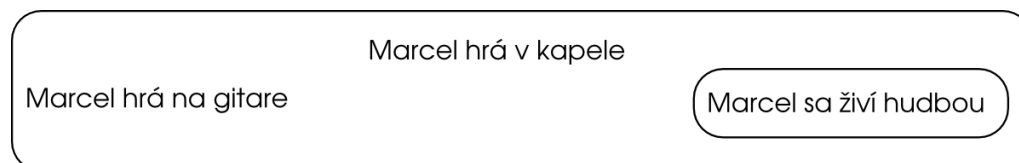
Obr. 2–2 EG: Marcel hrá na gitare, hrá v kapele a živí sa hudbou

Ohraničením grafu, tzv. rezom dostávame jeho negáciu.

Marcel hrá na gitare

Obr. 2–3 EG: Marcel nehrá na gitare

Graf na obr. 2–3 teda znamená „nie je pravda, že Marcel hrá na gitare“, prípadne prirodzenejšie „Marcel nehrá na gitare“. Oblasť vo vnútri rezu sa nazýva jeho uzavretím. Rezy sa nesmú pretínať, dotýkať ani prekryvať, no môžu byť vnorené. Graf na obr. 2–4 znázorňuje vnorené rezy.



Obr. 2–4 EG: Marcel hrá na gitare a v kapele, teda sa živí hudbou

EG sa čítajú smerom dovnútra. Predchádzajúci graf by sa dal prečítať takto: „nie je pravda, že Marcel hrá na gitare, že Marcel hrá v kapele a že nie je pravda, že sa Marcel živí hudbou“ alebo „Ak Marcel hrá na gitare a v kapele, tak sa živí hudbou“. Akákoľvek oblasť listu tvrdení môže byť ohraničená párne alebo nepárne v závislosti od počtu rezov, ktoré sa na nej nachádzajú. O prvkoch v týchto oblastiach potom hovoríme, že sú ohraničené priamo alebo nepriamo. Jednotka dvoch rezov, kde je jeden z nich vnorený do druhého sa nazýva „zvitkom“ a zodpovedá implikácii. Ak v prvej oblasti nemá žiadne predikáty, nazýva sa dvojitém rezom a zodpovedá dvojitej negácii. Peirce popísal základný kalkulus nad EG takto:

- **Vymazanie** – každý párne ohraničený podgraf môže byť vymazaný.
- **Vkladanie** – akýkoľvek graf môže byť vpísaný do akejkol'vek nepárne ohraničenej oblasti.
- **Iterácia** – Ak sa podgraf **S** nachádza na liste prehlásení alebo v reze, tak jeho kópia môže byť zapísaná na tú istú alebo vnorenú oblasť, ktorá nepatrí **S**.
- **Deiterácia** – Každý podgraf, ktorého výskyt môže byť výsledkom iterácie, môže byť vymazaný.
- **Dvojitý rez** – dvojitý rez môže byť vložený alebo vymazaný okolo akejkol'vek oblasti.

2.1.2 Beta časť

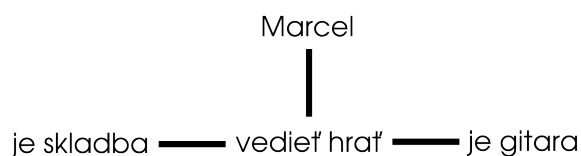
Rozdiel medzi EG typu alfa a beta zodpovedá rozdielu medzi predikátovou logikou a predikátovou logikou prvého rádu. V predikátovej logike prvého rádu je zavedený nový symbol - linka identity. Tá vyjadruje rovnako existenciu objektov, ako aj identitu medzi objektami a graficky je reprezentovaná jednoducho ako čiara. Okrem toho v beta EG už nepoužívame iba medády, teda predikáty nulovej arity, ale môžeme používať aj predikáty s rozdielnymi aritami.

Marcel — v — kapela

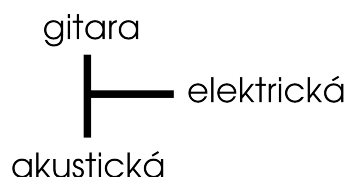
Obr. 2 – 5 EG: Marcel v kapele

Napríklad graf na obr. 2 – 5 obsahuje dve čiary identity, teda definuje dva (nie nevyhnutne rozdielne) objekty: Marcela a kapelu. Obe čiary sú prepojené s diadickým predikátom „v“. Graf by sa dal prečítať ako „existuje taký Marcel a kapela, že Marcel je jej súčasťou“ alebo jednoduchšie „Marcel je členom kapely“. :

Analogicky, graf na obr. 2 – 6 znamená „Marcel vie hrať skladbu na gitare“. Linky identity môžu byť spájané do sietí. Tie sa nazývajú ligatúry.



Obr. 2–6 EG: Marcel vie hrať skladbu na gitare



Obr. 2–7 EG: Elektro-akustická gitara

Napríklad v grafe na obr. 2–7 sa nachádza ligatúra troch liniek identity, ktoré sa stretávajú v takzvanom „rozvetvujúcom bode“ a jeho význam je „existuje objekt, ktorý je gitara, akustický a elektrický“ alebo skráteno „existuje elektro-akustická gitara“. Ligatúry môžu pretínať rezy.



Obr. 2–8 EG: Ukážka pretínania rezov

Tri grafy na obr. 2–8 značia „*existuje Marcel*“, „*nie je pravda, že existuje Marcel*“ a do tretice „*existuje niečo, čo nie je Marcel*“. Čiara, ktorá prechádza rezom, popisuje neidentitu objektov, ktoré prepája.



Obr. 2–9 EG: Dvaja Marcelovia

Graf na obr. 2–9 znamená, že existuje nejaký objekt (osoba) **o1**, ktorý je (volá sa) Marcel a rovnako existuje aj nejaký iný objekt **o2**, ktorý je Marcel, no tieto dva objekty nie sú identické. Teda vyjadruje existenciu dvoch rozdielnych Marcelov. Pomocou beta EG teda môžeme vyjadriť existenciálny kvantifikátor, predikáty väčších arít, konjunkciu a

negáciu, čo zodpovedá predikátovej logike prvého rádu. Pravidlá pre prácu s beta grafmi sú rozšírením pravidiel pre alfa grafy:

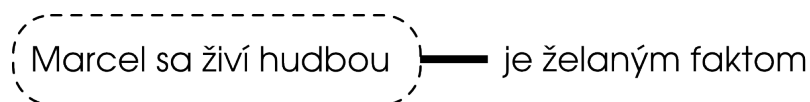
- **Mazanie** – Každý párne uzavretý podgraf a každá párne ohraničená časť linky identity môžu byť zmazané.
- **Vloženie** – akýkoľvek graf môže byť zapísaný do nepárne uzavretej oblasti a dve časti liniek identít, ktoré sú nepárne uzavreté v rámci jednej oblasti, môžu byť spojené.
- **Iterácia** – ak podgraf **S** je na liste prehlásení alebo v reze, tak kópia tohto grafu môže byť zapísaná do rovnakej alebo vnorenej oblasti, ktorá nepatrí do **S**. V rámci tejto operácie je povolené prepojiť akúkoľvek linku identity podgrafu **S**, ktorá nie je zapísaná v oblasti akéhokoľvek rezu podgrafu **S**, s jej kópiou.
- **Deiterácia** – akýkoľvek podgraf, ktorého výskyt môže byť výsledkom iterácie, môže byť zmazaný.
- **Dvojitý rez** - akýkoľvek dvojitý rez môže byť vytvorený alebo zmazaný okolo akejkol'vek oblasti. Táto transformácia je povolená aj v prípade, že máme ligatúry, ktoré začínajú mimo vonkajšieho rezu a prechádzajú oblasťou vonkajšieho rezu dovnútra vnútorného rezu.

2.1.3 Gama časť

Gama časť EG nebola nikdy dokončená a je ťažké nadobudnúť istotu pri chápaní Peircových zámerov. Veľmi hrubo povedané: Spája v sebe logiku vyššieho rádu a modálnu logiku. Snáď najznámejšou jednotkou gama časti je takzvaný „lomený rez“. Takým rezom sú označované tvrdenia, o ktorých pravdivosti nemáme dostatočné informácie, a preto môžu byť aj pravdivé, aj nepravdivé.

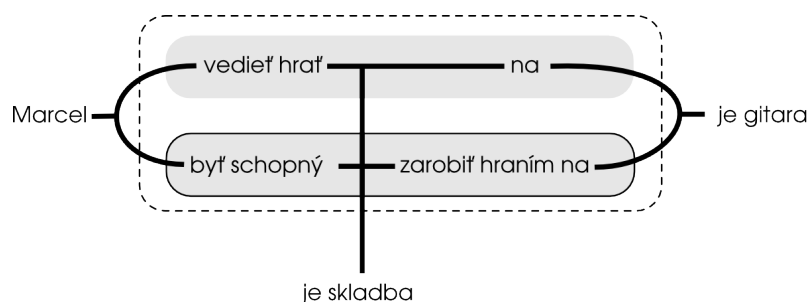
Ďalším dôležitým aspektom gama grafov je možnosť vyjadrenia meta-úrovňových výrokov. Zjednodušene povedané, kým beta grafy hovorili o objektoch, gama grafy môžu byť

považované za objekty samotné. Objekty sú zakreslené podobne ako rezy, ale krivka je zakresľovaná svetlejším odtieňom (v tejto práci prerušovanou čiarou).



Obr. 2 – 10 EG: Je želaným faktom, aby sa Marcel živil hudbou

Graf na obr. 2 – 10 potom znamená: „*Marcel sa živí hudbou*“ je želaným faktom. Peirce pracoval s týmto postupom ďalej a rozličnými farbami vyjadroval rozdielne kontexty. Napríklad červenou (v tejto práci je použitá šedá farba) vyjadroval skutočnosť.



Obr. 2 – 11 EG: Zložitejší EG

V poslednom príklade na obr. 2 – 11 máme dva červené (šedé) objekty. Prvý je červený (šedý), čiže vyjadruje, že jeho vnútro je možné. Druhý objekt je síce tiež vyplnený červenou (šedou) farbou, no je ohraničený rezom, tým pádom vyjadruje, že jeho obsah je nemožný. Daný graf teda v skratke vyjadruje, že Marcel vie zahrat' nejakú skladbu na gitare, ale nie je schopný si týmto spôsobom zarobiť.

2.2 Konceptuálne grafy

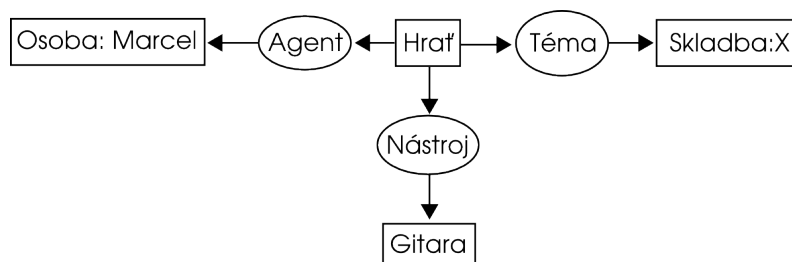
2.2.1 Definícia Konceptuálneho grafu

KG g je bipartitný graf s dvoma typmi vrcholov – konceptami a konceptuálnymi reláciami. Každá hrana grafu g musí spájať konceptuálnu reláciu r v g s konceptom c v g . Hovoríme

potom, že hrana patrí do relácie **r**. Hovoríme, že je pripojená na koncept **c**. Konceptuálny graf **g** môže mať koncepty, ktoré nie sú pripojené na žiadnu konceptuálnu reláciu v grafe **g**, ale každá hrana patriaca nejakej konceptuálnej relácii v grafe **g**, musí byť pripojená najmenej na jeden koncept v danom grafe.

Rozlišujeme 3 druhy KG podľa osobitných mien:

- **Prázdny KG** je konceptuálny graf bez konceptov, konceptuálnych relácií a hrán.
- **Singletonový KG** je konceptuálny graf, ktorý pozostáva z jediného konceptu, ale neobsahuje žiadne konceptuálne relácie ani hrany.
- **Hviezdicový KG** je konceptuálny graf, ktorý pozostáva z jedinej konceptuálnej relácie a konceptov, ktoré sú pripojené na jej hrany.



Obr. 2 – 12 EG: Zložitejší EG

Na obr. 2 – 12 je ilustrovaný jednoduchý KG pre vetu „Marcel hrá skladbu X na gitare“. Pojem bipartitný nám určuje, že každá hrana grafu spája koncept s konceptuálnou reláciou a nikdy nespája dva koncepty alebo konceptuálne relácie.

Konceptuálny graf **g** s *n* konceptuálnymi reláciami môže byť zostrojený z *n* hviezdicových grafov pôvodne vytvorených pre každú konceptuálnu reláciu grafu **g**. Keďže náš príklad obsahuje tri konceptuálne relácie, môže byť zostavený z týchto troch grafov:

$[Osoba : Marcel] \leftarrow (Agent) \leftarrow [Hrat]$

$[Hrat] \rightarrow (Tema) \rightarrow [Skladba : X]$

$[Hrat] \rightarrow (Nastroj) \rightarrow [Gitara].$

2.2.2 Koncept

Každý koncept má konceptuálny typ \mathbf{t} a referenta \mathbf{r} .

Táto abstraktná definícia nehovorí nič o tom, ako má byť typ a referent reprezentovaný. Koncept je teda definovaný svojím typom a referentom. Ilustrujeme to na príkladoch, kde koncept zapíšeme do hranatých zátvoriek ako dvojicu typ:referent oddelenú dvojbodkou. Napríklad koncept [Osoba:] je reprezentovaný len svojím typom „Osoba“ a obsahuje aj prázdne pole pre referent, čo vyjadruje existencionálny kvantifikátor. Naproti tomu koncept [Osoba:Macel] reprezentuje koncept typu „Osoba“ a určuje aj konkrétnu osobu „Marcela“, ktorý je jeho referentom. Referent môže byť aj úplne vynechaný. V tom prípade do hranatých zátvoriek píšeme iba typ konceptu ([Osoba]) a chápeme ako koncept prípad popísaný vyššie, keď referentom je prázdne pole.

2.2.3 Konceptuálna relácia

Každá konceptuálna relácia \mathbf{r} má relačný typ \mathbf{t} a nezáporné celé číslo n , ktoré sa nazýva valencia.

- Valencia vyjadruje počet hrán, ktoré patria konceptuálnej relácii \mathbf{r} . Konceptuálnu reláciu s určitou valenciou veľkosti n potom nazývame n -adickou konceptuálnou reláciou.
- Pre každú n -adickú reláciu \mathbf{r} potom existuje rad konceptuálnych typov t_1, \dots, t_n , a nazýva sa signatúrou \mathbf{s} relácie \mathbf{r} .
- Všetky konceptuálne relácie rovnakého relačného typu potom majú rovnakú valenciu n a rovnakú signatúru \mathbf{s} .
- Výraz monadický znamená 1-adický, dyadický znamená 2-adický a triadický znamená 3-adický.

Klasickým príkladom triadickej relácie by bola relácia „medzi“. Dve z jej hrán sú napojené na dva objekty, ktoré sú po stranách objektu, na ktorý ukazuje. Príklad 5-adickej relácie je relácia „byť“ (v zmysle tvoriť kapelu) na obr. 1 – 3.

2.2.4 Výraz lambda

Pre každé nezáporné celé číslo n je n -adický výraz lambda e konceptuálnym grafom, ktorý sa nazýva telom e , a kde n konceptov bolo určených ako formálne parametre e .

- Formálne parametre e sú očíslované od 1 do n .
- Je daná postupnosť (t_1, \dots, t_n) nazvaná signatúrou výrazu e , kde t_i je konceptuálny typ i -teho formálneho parametra výrazu e . Vzhľadom na to, že 0-adický lambda výraz nemá žiaden formálny parameter, je jeho signatúrou prázdna postupnosť.

Táto abstraktná definícia nešpecifikuje určovanie formálnych parametrov. Tradičná notácia spočíva v označení parametra gréckym písmenom lambda λ . Ak $n > 1$, potom môžu byť parametre označené $\lambda_1, \lambda_2, \dots, \lambda_n$. Ako príklad nám môže poslúžiť konceptuálny graf pre vety Marcel hrá skladbu X. Ten môže byť transformovaný na nasledujúci lambda výraz, kde meno „Marcel“ vymeníme za symbol λ_1 a meno skladby „X“ vymeníme za λ_2 .
 $[Osoba : \lambda_1] \leftarrow (Agent) \leftarrow [Hrat] \rightarrow (Tema) \rightarrow [Skladba : \lambda_2]$

Tento lambda výraz môže byť napríklad použitý na definovanie konceptuálnej relácie, ktorá priradzuje osobe skladbu, ktorú hrá.

2.2.5 Typ konceptu

Hierarchia typov je čiastočne zoradená množina \mathbf{T} , ktorej prvky sa nazývajú typové značky. Každá typová značka \mathbf{T} je špecifikovaná ako primitívum alebo definovaná.

- pre akýkoľvek koncept c , typom c je buď typová značka \top , alebo monadický lambda výraz .

- Hierarchia typov \top obsahuje dve primitívne typové značky: *Entity* nazývanú univerzálnym typom a *Absurdity* nazývanú absurdným typom. Symbol \top predstavuje synonymom pre *Entity* a symbol \perp je synonymom pre *Absurdity*.
- Pre každú definovanú typovú značku existuje monadický lambda výraz nazývaný definícia.
- Definovaná typová značka a jej definícia sú zameniteľné. Všade, kde sa môže vyskytnúť jedna z nich, môže byť nahradená druhou.
- Čiastočné zoradenie nad \mathbf{T} je určené reláciou podtypu reprezentovanou značkou \leq pre podtyp, $<$ pre čistý podtyp, \geq pre supertyp a $>$ pre čistý supertyp. Ak \mathbf{t} je typová značka $\mathbf{Entity} \geq \mathbf{t}$ a $\mathbf{t} \geq \mathbf{Absurdity}$... vo všeobecnosti $\mathbf{Entity} > \mathbf{Absurdity}$.
- Čiastočné zoradenie typových značiek musí byť konzistentné s pravidlami inferencie.

Hierarchia typov musí začínať nejakou množinou primitívnych typových značiek, ktorá v sebe zahŕňa prinajmenšom *Entitu* a *Absurditu*. Definičný mechanizmus potom udáva nové typové značky, ktorých miesto v hierarchii je určené ich vlastnou definíciou. Ako príklad uvádzame rovnicu definujúcu typovú značku *JazzovýHráč* pomocou lambda výrazu pre hráča hrajúceho jazz:

$$\mathbf{JazzovyHrac} = [\mathbf{Hrac} : \lambda] \rightarrow (\mathbf{styl}) \rightarrow [\mathbf{Zaner} : \mathbf{Jazz}].$$

Znak " λ " hovorí, že koncept $[\mathbf{Hrac}]$ je formálnym parametrom a sekvencia (\mathbf{Hrac}) je signatúrou lambda výrazu. Typová značka formálneho parametru je vždy supertypom novo definovaného typu: $\mathbf{Hrac} \geq \mathbf{JazzovyHrac}$.

2.2.6 Typ relácie

Hierarchia relácií je čiastočne zoradená množina \mathbf{R} , ktorej prvky sa nazývajú relačné značky. Každá relačná značka je špecifikovaná buď ako primitívna, alebo definovaná.

- Pre každú relačnú značku v \mathbf{R} existuje nezáporné celé číslo, ktoré sa nazýva jej valenciou.
- Pre každú n -adickú konceptuálnu reláciu \mathbf{r} je typom \mathbf{r} relačná značka z \mathbf{R} valencie n alebo n -adický lambda výraz.
- Pre každú definovanú relačnú značku valencie n existuje práve jeden n -adický lambda výraz, ktorý nazývame jej definíciou.
- Definovaná relačná značka a jej definícia sú zameniteľné. Kdekoľvek sa môže vyskytnúť jedna, môžeme ju zameniť druhou.
- Čiastočné usporiadanie nad \mathbf{R} je určené podtypovou reláciou so symbolmi \leq pre podtyp, $<$ pre čistý podtyp, \geq pre supertyp, a $>$ pre čistý supertyp.
- Čiastočné zoradenie relačných značiek musí byť konzistentné s pravidlami inferencie.
- Ak \mathbf{r} je n -adická relačná značka, \mathbf{s} je m -adická relačná značka a n sa nerovná m , potom nič z nasledujúceho neplatí: $r < s$, $r > s$, $r = s$.

Ako príklad možno uviesť definíciu vyššie spomínanej relácie pre diadický lambda výraz, ktorá priradí osobe skladbu, ktorú vie daná osoba zahrať: $[Relacia : HratSkladbu] \rightarrow (Def) \rightarrow [LambdaVyzraz :$

$[Osoba : \lambda_1] \leftarrow (Agent) \leftarrow [Hrat] \rightarrow (Tema) \rightarrow [Skladba : \lambda_2]]$

2.2.7 Referent

Referent konceptu je špecifikovaný kvantifikátorom, designátorom a deskriptorom.

- **Kvantifikátor** môže byť dvojakého druhu: existencionálny alebo definovaný.
- **Designátor** môže byť buď:
 1. *literál* - číslo, reťazec alebo literál,

2. *lokátor* - ktorý môže byť individuálnou značkou, indexom alebo menom,

3. *neurčený*.

- **Deskriptor** je konceptuálny graf, najčastejšie prázdny, ktorý má za cieľ popisovať referenta.

Kvantifikátor, designátor a deskriptor určujú prepojenia medzi formalizmom KG a entitami, ktoré označuje. Implementácia týchto prepojení nie je určená, keďže môžu označovať veci, ľudí, akcie alebo udalosti z vonkajšieho sveta. Existenčný kvantifikátor deklaruje, že existuje najmenej jedna inštancia daného typu. Definovaný kvantifikátor môže už bližšie špecifikovať množstvo. Designátor špecifikuje referenta tým, že buď ukazuje jeho formu (literál), alebo na neho ukazuje (lokátor). Tri druhy lokátorov sa líšia podľa toho, ako je referent určený. Jedinečný koncept vybraný zo zoznamu znalostnej bázy špecifikuje individuálna značka. Index je symbol, ktorý určuje referenta podľa naimplementovaného vyhľadávania a meno je symbol, ktorý určuje referenta podľa nejakých konvencií, ktoré sú nezávislé na aktuálnej znalostnej báze.

Príklad deskriptora vidieť na obr. 1–3. Typom konceptu je „Stav“ a deskriptorom je konceptuálny graf, ktorý popisuje existenciu konkrétnej kapely.

2.2.8 Kontext

Kontext **C** je koncept, ktorého designátor je nie čistý konceptuálny graf **g**.

- Graf **g** sa nazýva bezprostredne vnorený v **C** a akýkoľvek koncept **c** z **g** je tiež bezprostredne vnorený v **C**.
- Koncept **c** sa nazýva vnorený v **C** ak **c** je bezprostredne vnorený v **C** alebo **c** je bezprostredne vnorený v nejakom kontexte **D**, ktorý je vnorený v **C**.
- Dva koncepty **c** a **d** sú medzivnorené ak **c = d** alebo existuje kontext **C**, v ktorom sú **c** a **d** bezprostredne vnorené.

- Ak koncept x je medzivnorený s kontextom \mathbf{C} , potom hovoríme, že akýkoľvek koncept vnorený v \mathbf{C} je hlbšie vnorený ako x .
- Koncept \mathbf{d} sa nenazýva v rámci konceptu \mathbf{c} , ak \mathbf{d} je spoluvnorený s \mathbf{c} alebo \mathbf{d} je hlbšie vnorený ako \mathbf{c} .

Kontext je koncept s vnoreným konceptuálnym grafom, ktorý popisuje referenta. Na obr. 1 – 3 poslúži ako príklad kontextu koncept typu „Situácia“.

2.3 Ontológie

Z oblasti znalostného modelovania je ontológia snáď najrozšírenejším modelom. Čiasťtočne je to spôsobené súčasným trendom a pojem ontológia sa často používa aj ako zámena za pojmy klasifikácia alebo taxonómia.

Pojem „*ontológia*“ v poslednom čase získava v oblasti znalostného inžinierstva na popularite. Jeho počiatky siahajú až do starogréckej filozofie, kde reprezentovala filozofickú disciplínu zaoberajúcu sa „*bytím*“. Systematické vymedzenie ontológie v podobe vedy, v súvislosti s najvšeobecnejšími určeniami bytia, významom bytia a pojmami bytia, sa nachádza v diele Ch. Wolffa „*Ontológia*“ z roku 1730. Jej prvopočiatky siahajú už do obdobia Aristotela, ktorý položil základnú otázku tejto filozofickej disciplíny a to „*Čo je bytie?*“. Pojem „*ontológia*“ v rámci umelej inteligencie, resp. znalostného inžinierstva aj napriek tomu stále ostáva vágny a vedecké kruhy sa plne nezhodujú na určení jeho presného významu.

Medzi najznámejšie ontológie patrí Wordnet a Cyc, tie budú bližšie popísané v nasledujúcich kapitolách. V súčasnosti sa ontologické modely používajú v širokom doménovom spektre popísanom napríklad v [14]. Od projektu GENSIM z oblasti medicíny, cez projekt PLINIUS z oblasti mechaniky, až po projekt CYC, ktorý sa snaží popísať bežné znalosti z reálneho sveta.

2.3.1 Definícia ontológie

Existuje množstvo definícií, ktoré sa rôzne prelínajú. Uvedme aspoň zopár relevantných podľa [17]:

1. Ontológia ako filozofická disciplína.
2. Ontológia ako neformálny konceptuálny systém.
3. Ontológia ako formálny sémantický význam (account).
4. Ontológia ako špecifikácia konceptualizácie.
5. Ontológia ako reprezentácia konceptuálneho systému teórie logiky:
 - (a) charakterizovaná špecifickými formálnymi vlastnosťami
 - (b) charakterizovaná len svojím špecifickým významom
6. Ontológia ako slovník využívaný teóriou logiky.
7. Ontológia ako (meta-úrovňová) špecifikácia teórie logiky.

V mierne pozmenenej podobe sa vo vyššie uvedenom zozname nachádza aj definícia od T. R. Grubera [16] prijímaná s menšími pripomienkami v UI komunite :

Ontológia je explicitnou špecifikáciou konceptualizácie.

N. Guarino predkladá [17] aj ďalšie definície, stavia sa k nim kriticky a odhaľuje ich nedostatky z viacerých hľadísk. Za uspokojivú nakoniec pokladá túto definíciu:

Ontológia je explicitná, čiastočná špecifikácia konceptualizácie, ktorá sa dá vyjadriť ako meta-úrovňový náhľad na množinu možných doménových teórií za účelom modulárneho návrhu, prestavby a opätovného použitia systémových častí súvisiacich so znalosťami [47].

Konceptualizácia hovorí, že sa jedná o taký abstraktný model výseku reálneho sveta, ktorý identifikuje relevantné koncepty daného výseku. Explicitná znamená, že je jednoznačne definovaný typ konceptu i podmienky jeho použitia [49].

Podľa [49] je možné ontológie ďalej deliť do štyroch základných skupín podľa zdrojov konceptualizácie:

- Generické ontológie alebo ontológie vyššieho rádu usilujúce o zachytenie bežných zákonitostí.
- Doménové ontológie sú najčastejším typom a ich predmetom je stále nejaká určitá špecifická oblasť.
- Úlohové ontológie označujú generické modely znalostných úloh a metód ich riešení.
- Aplikačné ontológie sú najšpecifickejšie a v ich prípade sa jedná o konglomerát modelov adaptovaných na konkrétnu aplikáciu spravidla zahrňujúc doménovú aj úlohovú časť.

Podľa [54] nachádzajú svoje uplatnenie v troch hlavných oblastiach:

- Komunikácia
- Inter-operabilita
- Systémové inžinierstvo: špecifikácia, spoľahlivosť a možnosť opakovaného použitia

Veľmi jednoduchú a pochopiteľnú definíciu poskytuje aj Web Ontology Working Group. Tá definuje ontológiu takto: *Ontológia je strojovo zrozumiteľná množina definícií, ktorá vytvára taxonómiu tried, podtried a relácie, ktoré medzi nimi existujú.*

Na definovanie ontológií sa používajú ontologické jazyky. Jedná sa často (CycL, KIF) o deklaratívne jazyky založené na predikátovej logike prvého rádu. KG sa dajú tiež chápať ako spôsob zápisu ontológií [58].

V nasledujúcich podkapitolách priblížime dve často uvádzané ontológie: CYC a Wordnet.

2.3.2 Wordnet

WordNet je sémantický lexikón anglického jazyka. Zoskupuje anglické slová do množín synonym, nazývaných synsety a poskytuje krátke definície a záznamy rôznych sémantic-

kých relácií medzi týmito synsetmi. Má to dvojaký účel: poskytnúť slovník a tezaurus, ktorý by bol viac intuitívne použiteľný a poskytnúť podporu automatickej analýzy textov a aplikácií pre UI. Databáza a softwarové nástroje sú vydané pod BSD style licenciou a môžu byť voľne stiahnuté a používané. Databázu je možné prezerat' si aj on-line.

WordNet vznikol v roku 1985 a je udržiavaný na pôde Princetonskej univerzity v Laboratóriu kognitívnych vied pod vedením profesora psychológie George A. Millera. V roku 2003 obsahovala databáza okolo 140 000 slov organizovaných do viac ako 110 000 synsetov s celkovým počtom 195 000 slovných párov.

WordNet rozlišuje medzi podstatnými menami, slovesami, prídavnými menami a príslovkami na predpoklade, že aj v ľudskom mozgu sa tieto štruktúry ukladajú rozdielne. Každý synset obsahuje množinu synonymických slov alebo slovných spojení, ktoré produkujú špecifický význam (napr. Vysoká škola). Slová sa zvyčajne vyskytujú vo viacerých synsetoch naraz. Význam synsetov je ďalej doladený krátkymi poznámkami.

Typickým príkladom synsetu s poznámkou je: správny, vhodný, akurátny – (najvhodnejší pre dané špecifické účely; „správny čas orať“, „vhodný moment na ústup“, „akurátny prístup k danej problematike“).

Každý synset je relačne prepojený na iné synsety. Tieto relácie závisia od typu slova. Jedná sa o tieto typy relácií.

- Synonymá – synsety s podobným významom.
- Hypernymá - Y je hypernymom k X, ak každé X je (ako) Y.
- Hyponymá - Y je hyponymom k X, ak každé Y je (kind of) X.
- Holonymá - Y je holonymom k X, ak X je časťou Y.
- Meronymá - Y je meronymom k X, ak Y je časťou X.
- Antonymá – slová opačného významu.
- Rovnocenné pojmy – slová zdieľajúce bežné hypernymum.

Ako príklad nám môže poslúžiť anglické slovo „string“ myslené vo význame „struna“ (napríklad na gitare). Použitím on-line nástroja [57] po zadaní daného slova dostaneme zoznam možných výsledkov. Pre možnosť „a tightly stretched cord of wire or gut, which makes sound when plucked, struck, or bowed“ je možný dotaz „domain category“, ktorý doménovou reláciou dáva strunu do súvislosti s „music (an artistic form of auditory communication incorporating instrumental or vocal tones in a structured and continuous manner)“, čiže hudbou.

WordNet poskytuje hodnotu „polysemy count“, ktorá vyjadruje počet synsetov, ktoré dané slovo obsahujú. Ak sa dané slovo vyskytuje vo viacerých synsetoch (t.z. má viacero významov), tak je zrejmé, že niektoré významy sú bežnejšie ako iné. WordNet to kvantifikuje pomocou „frequency score“. V niekoľkých vzorkách textov sú všetky slová sématicky označené korešpondujúcim synsetom a neskôr je spočítané koľkokrát sa vyskytli s daným špecifickým významom. Databázové rozhranie vie z užívateľského vstupu dedukovať koreňovú formu slova a iba do databázy sa ukladajú iba tieto formy [43].

2.3.3 CYC

Cyc je projektom, ktorý sa snaží zhromaždiť komplexnú ontológiu a databázu tvorenú každodennými praktickými znalosťami s cieľom umožniť UI aplikáciám rozhodovanie podobné ľudskému.

Projekt vznikol v roku 1984 pod vedením Douga Lenata ako časť „Microelectronics and Computer Technology Corporation“. Názov „Cyc“ (odvodený z anglického slova encyclopedia) je registrovanou obchodnou značkou firmy Cycorp, Inc. v Texase. Báza znalostí, ktorú systém využíva, je majetkom spoločnosti, ale jej menšia verzia vydaná pod názvom OpenCyc je vydaná pod open source licenciou [56].

Typickými príkladmi znalostí reprezentovaných v databáze je napríklad veta „Každý strom je rastlina“ a „Rastliny hynú“. Pri otázke, či hynie strom, dôjde inferenčný systém k logickej odpovedi. Báza znalostí obsahuje vyše milióna ručne napísaných výrokov, pravidiel a jednoznačných myšlienok. Tie sú formulované v jazyku CycL, založenom na predikáto-

vom počte so syntaxou veľmi podobnou Lispu.

Názvy konceptov sa v Cyc označujú ako „konštanty“. Konštanty začínajú znakom „#\$“ a môžu označovať individuálne položky, množiny, pravdivostné funkcie a funkcie. Pravdivostné funkcie sa delia na logické spojky, kvantifikátory a predikáty. Funkcie vracajú nové termy na základe vstupných parametrov. Najdôležitejšími predikátmi sú #\$.isa a #\$.genls. Kým #\$.isa popisuje, že položka je prvkom nejakej kolekcie, #\$.genls vyjadruje, že množina je podmnožinou inej množiny. Fakty o konceptoch sú vyjadrené pomocou špecifických CycL viet. Predikáty sú zapísané pred argumentami v zátvorkách:

```
(#$isa #$.Marcel #$.Gitaristi)
```

„*Marcel patrí do množiny Gitaristi*“ a

```
(#$genls #$.Gitarista #$.Hudobnik)
```

„*Všetci gitaristi sú hudobníci*“.

Vety môžu obsahovať aj premenné označené reťazcom začínajúcim znakom „?“:

```
(#$implies
```

```
  ($and
```

```
    ($isa ?OBJ ?SUBSET)
```

```
    ($genls ?SUBSET ?SUPERSET)
```

```
  )
```

```
  ($isa ?OBJ ?SUPERSET)
```

```
)
```

čo interpretujeme ako „ak OBJ je inštanciou množiny SUBSET a SUBSET je podmnožinou množiny SUPERSET, potom OBJ je inštanciou množiny SUPERSET“ [43].

3 Tvorba a využitie konceptuálnych znalostných sietí

V dnešnej dobe je využitie znalostných sietí naozaj široké. Od pôvodného zámeru reprezentácie databázových systémov až po dolovanie zdrojového kódu alebo modelovanie sémantiky agentových systémov. V nasledujúcich kapitolách si bližšie popíšeme možnosti tvorby a využitia konceptuálnych znalostných štruktúr. Uvedieme pár reálnych príkladov a vzhľadom na zameranie tejto práce sa bližšie budeme zaoberať aj asociatívnym učeníím pojmov. Príklady boli vybrané tak, aby ilustrovali spektrum využitia, a zároveň sa sústreďovali na získavanie relácií medzi konceptami z textových dokumentov, prípadne ilustrovali zaujímavé postupy tvorby daných znalostných štruktúr. Kapitola neponúka podrobný prehľad všetkých možností tvorby a využitia konceptuálnych ZS, ale môže poslúžiť ako prvotný zdroj informácií pri ich skúmaní.

3.1 Dolovanie bayesovských sietí z textov

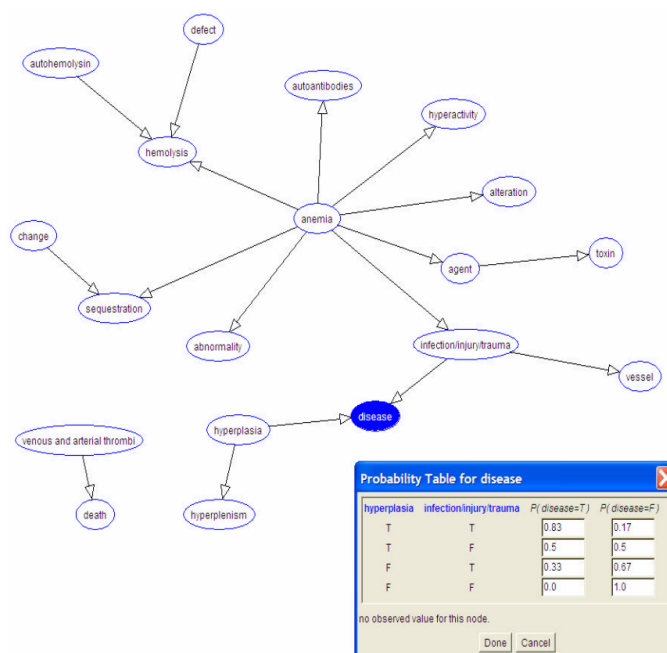
Pomerne aktuálnym problémom je dnes dolovanie kauzálnych relácií z textov v prirodzenom jazyku. Existencia takého automatického procesu by uľahčila prehľad trebárs v doméne medicínskych dokumentov [53] týkajúcich sa vzájomného pôsobenia génov.

V [46] je predstavená metóda dolovania kauzálnych vzorov z textov. Za kauzálny vzor sa v tomto prípade považuje myšlienka, prípadne časť vety, ktorá špecifikuje kauzálnu reláciu medzi príčinou a následkom. Tie sa v textoch vyhľadávajú na základe slovných spojení s funkciou spojok. Systém predpokladá, že príčina a následok sú oddelené v myšlienke nejakou spojkou, ktorá by mohla vykazovať funkciu relačného operátora.

Systém bol testovaný na množine textových súborov v anglickom jazyku a medzi príklady takýchto spojení v anglickom jazyku patria napríklad „*because*“, „*the cause of*“, „*causes*“ alebo „*is a result of*“. Po akumulácii kauzálnych vzorov v texte ďalej nasleduje ich analýza. Tá spočíva v bližšom preskúmaní príčin a následkov na základe spojok „*a*“ a „*alebo*“, interpunkčných znamienok a zisťovaní existencie negačného operátora pri

slovnom spojení, ktoré kauzálnu reláciu delí na príčinu a následok (v anglickom jazyku je to slovo „not“).

Nasleduje zovšeobecňovanie kauzálnych vzorov na základe podobností slov definovaných vo Wordnete [11]. Ak sú potom napríklad v príčinnej (kauzálnnej) časti nejakých rozdielnych vzorov slová, ktoré sú prvkami jedného synsetu, systém to berie do úvahy a môže ich vnímať ako jednu entitu. Do úvahy sa berú okrem synonym aj hypernymá. V poslednom kroku sa generuje bayesovská sieť. Uzliami sú v tomto prípade udalosti príčiny a následku a pravdepodobnostné hodnoty hrán, ktoré ich prepojujú a získavajú sa z početností výskytov kauzálnych vzorov, v ktorých sa vyskytli. Na obr. 3 – 1 je ilustrácia výstupu daného prístupu.



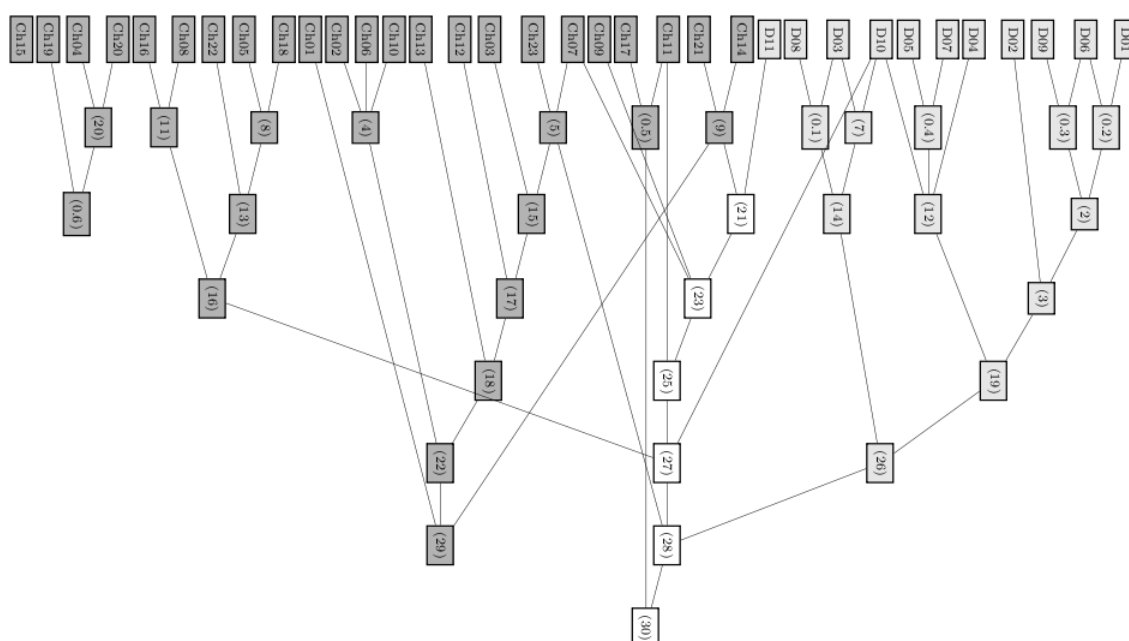
Obr. 3–1 Bayesovská sieť získaná z textu

V tomto prístupe bolo na riešenie úlohy využitých niekoľko ZS (bayesovské siete, Wordnet), taktiež ako metódy boli využité prístupy z rôznych oblastí (dolovanie v textoch, učenie pravdepodobnostných modelov).

3.2 Prístupy tvorby sociálnych sietí

3.2.1 Formálna fuzzy analýza

Metóda generovania sociálnej siete odzrkadľujúcej osobné vzťahy medzi žiakmi triedy základnej školy bola prezentovaná v [22]. Vstup predstavovala matica hodnotení, kde každý žiak ohodnotil svoj osobný postoj k ostatným spolužiakom na sedembodovej stupnici, ktorá odzrkadľovala tri stupne pozitívneho vzťahu, tri stupne negatívneho vzťahu a neutrálny vzťah. Takáto matica hodnotení (v prenesenom význame matica podobností) potom poslúžila ako vstup do zhlučovacieho algoritmu formálnej konceptuálnej analýzy. Jej výstupom bol dendrogram podobný tomu na obrázku 3 – 2. Výsledný graf nielen skoro dokonale odrážal pohlavie jednotlivcov, ale najmenšie zhluky s veľkou pravdepodobnosťou zodpovedali dvojiciam, ktoré spolu sedeli v lavici. Z podobných grafov je potom možné ľahšie vyčítať aj informácie o osobných vzťahoch, ktoré na prvý pohľad (napr. pre pedagóga) nemusia byť jasné.



Obr. 3 – 2 Sociálna sieť zobrazujúca osobné vzťahy medzi spolužiakmi

3.2.2 Identifikácia tématických sociálnych sietí

Otázka definície odborných väzieb medzi jednotlivcami môže byť zaujímavá v oblasti vedeckého bádania aj v komerčnej sfére. Pán Jelínek vo svojej práci [19] približuje metódu dolovania odborných väzieb medzi jednotlivcami z informácií z bežných webových vyhľadávacích systémov. Systém pracuje v troch krokoch:

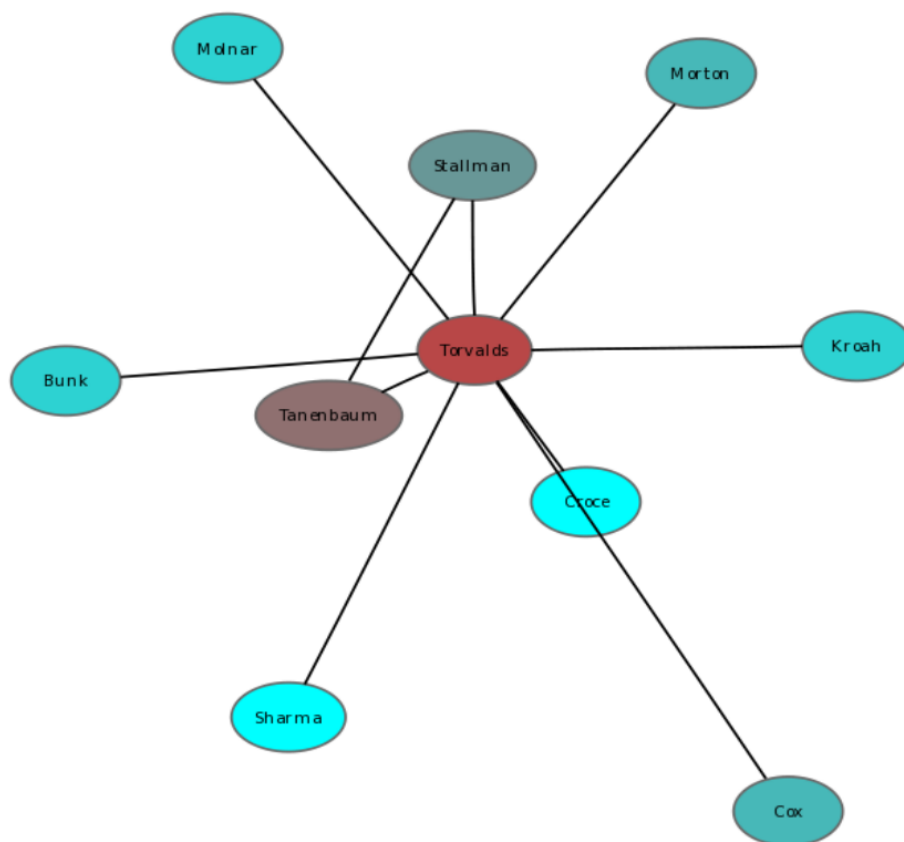
1. detekcia mien osôb,
2. spresnenie významu,
3. detekcia väzieb medzi danými osobami.

Cieľom prvého kroku je detekovať skutočné mená osôb v texte v prirodzenom jazyku. To sa deje opäť vo viacerých krokoch. V textoch sa hľadajú vzory zodpovedajúce maske pre meno (napr. „meno1 meno2 priezvisko“), kde by dané mená naozaj mohli patriť skutočným osobám. Identifikácia krstných mien a priezvísk je realizovaná nielen nástrojmi tretej strany (Named Entity Tagger, Wordnet), ale aj štatistickým prístupom učenia z prechádzajúcich príkladov. Spresnenie významu spočíva v hľadaní rôznych možných zápisov daných mien. Identifikácia väzieb medzi jednotlivými osobami je potom daná ich spoločným výskytom vo vstupných dokumentoch. Výstupom je potom sieť zobrazujúca odborné väzby medzi osobami z danej oblasti.

Na obr. 3 – 3 je ilustrovaný výstup daného prístupu vystavaný nad doménou „linux“.

3.3 Dolovanie zdrojového kódu

Komplexná úloha získavania, klasifikovania a extrahovania informácií zo zdrojových kódov – Code retrieval (CR) – je nevyhnutnou zložkou pri procese vývoja veľkých softvérových systémov. Keďže zdrojové kódy sú formou štruktúrovaného textu, to celý proces dolovania zdrojového kódu uľahčuje. Aj keď štrukturalizácia môže byť pre získavanie znalostí prínosná, je to za cenu veľkej výpočtovej náročnosti pri manipulácii s danou

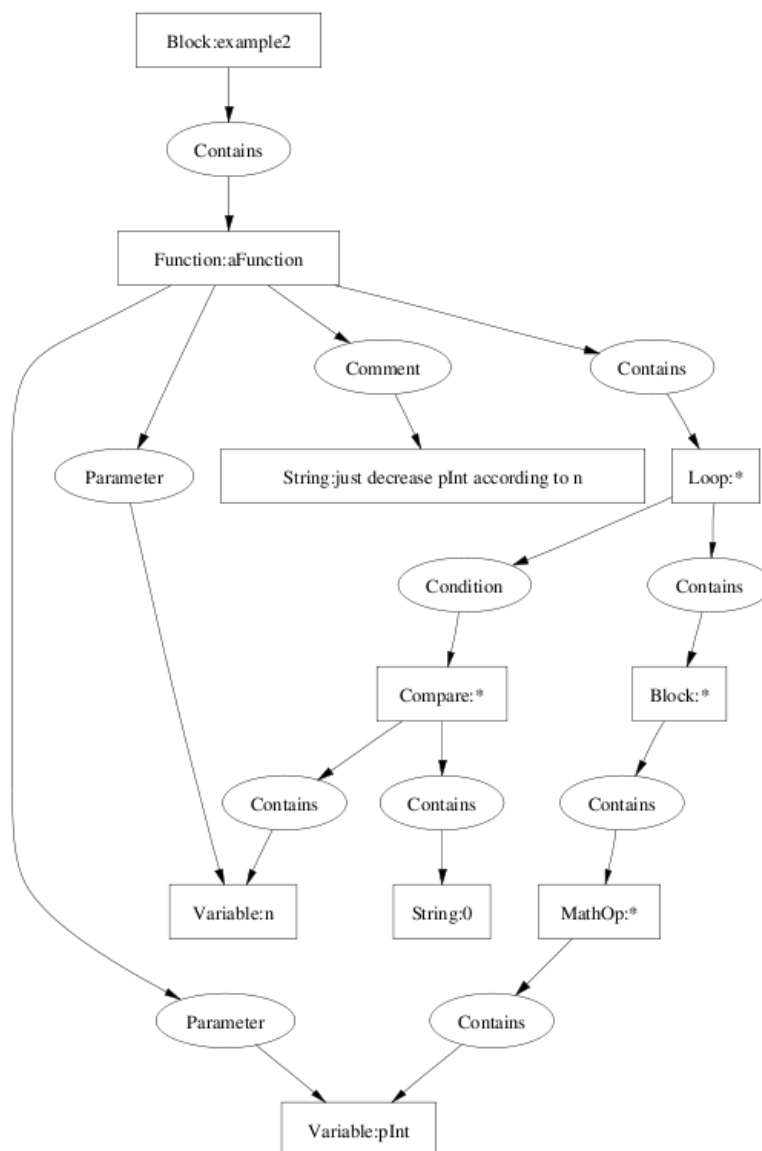


Obr. 3–3 Sociálna sieť vygenerovaná nad doménou „linux“

štruktúrou, resp. pri extrahovaní znalostí z nej. Jedným z možných prístupov k riešeniu tejto výpočtovej náročnosti je reprezentácia zdrojových kódov v podobe KG. V [28] sú popísané metódy na vytváranie a porovnávanie takýchto KG. Tie majú v konečnom dôsledku priaznivý dopad na riešenie problému duplicity v zdrojových kódoch a na odhaľovanie plagiátorstva.

Systém teda používa ako vnútornú reprezentáciu zdrojových kódov KG. Každý z uzlov konceptuálneho grafu patrí do určitej triedy vybranej z vopred pripravenej taxonómie. Uzol tak môže byť napríklad premenná, podmienka, cyklus alebo matematická operácia. Relatívne typy sú tiež dané vopred a obsahujú napríklad relácie „podmienka“, „obsahuje“ alebo „vracia“. Analýzou zdrojového kódu a vyhľadaním kľúčových slov pre programovací jazyk (v tomto prípade C) dokáže systém vytvoriť KG reprezentujúci danú časť programu.

Príklad krátkej funkcie jazyka C zobrazuje obr. 3 – 4.



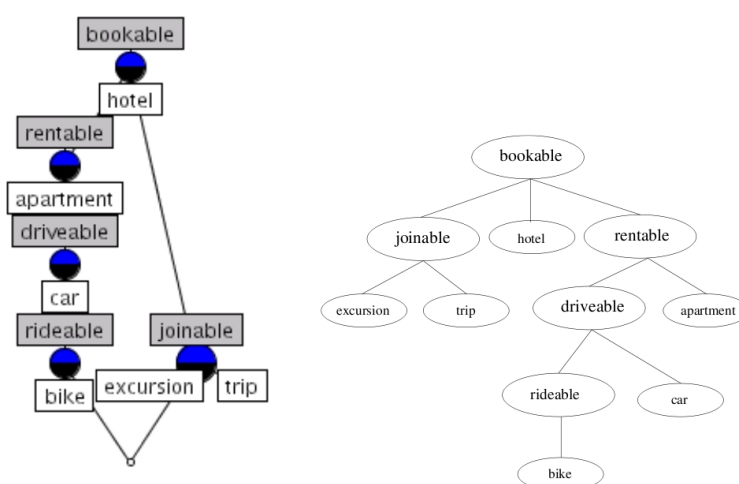
Obr. 3–4 Reprezentácia jednoduchkej funkcie jazyka C vo forme KG

Rôzne metódy na porovnávanie KG boli uvedené už v práci [51]. Pomocou týchto metód je potom odhaľovanie plagiátorstva, prípadne duplicít v kóde značne uľahčené.

3.4 Generovanie hierarchií konceptov pomocou FKA

Vzhľadom na vysokú pracnosť manuálneho budovania taxonómií konceptov pri výstavbe ontológií je výskum v oblasti ich automatického generovania pomerne živý. Jedna z možností budovania týchto štruktúr automatickými metódami z textových korpusov bola predstavená v [7].

Textový korpus je najprv otagovaný metódou part-of-speech pomocou TreeTaggeru a spracovaný pomocou LoPar2, a výstupom tejto fázy sú spracované syntaktické stromy pre každú vetu. Tie sú ďalej upravované lingvistickými nástrojmi a štatistickými metódami až do podoby tzv. formálneho kontextu, ktorý je vstupom do FKA (Fuzzy konceptuálna analýza). Autori porovnávajú FKA s inými zhlučovacími technikami a to hierarchickým aglomeratívnym zhlučovaním a Bi-section K-mean algoritmom, ako aj ďalšími zhlučovacími postupmi v ich nadväzujúcej práci [8]. Za pozitívnu vlastnosť FKA označujú vlastnosť jeho výstupu, ktorým nie je iba samotné pozhlučovanie konceptov, ale poskytuje aj ich popis. Výstup je samozrejme silne závislý na doméne, na ktorej bol postavený text, ktorý poslužil ako vstup. Príklad výstupu udáva obr. 3–5.



Obr. 3–5 Ukážka výstupu FKA zhlučovacieho algoritmu (vľavo) a jeho úpravy do taxonómie konceptov (vpravo)

3.5 Asociatívne učenie pojmov

Vzhľadom na naše skúsenosti a prechádzajúci výskum bude v tejto kapitole venované asociatívnu učenie pojmov (AUP) viac miesta ako predchádzajúcim prístupom. AUP sa nám javí ako veľmi perspektívny prístup a radi by sme sa aj naďalej venovali jeho využitiu pri tvorbe ZS.

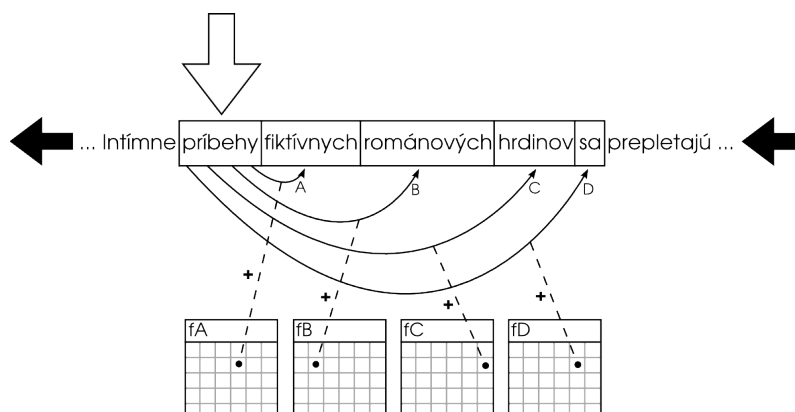
Asociatívne učenie pojmov je prístup založený na neuro-fyziologickom modeli spracovania talamo-kortikálnej informácie od R. Hecht Nielsena [31]. Model predpokladá existenciu fixného lexikónu „symbolov“ v ľudskom talame, ktorý sa vytvorí v skorom vývinovom štádiu jedinca. Učenie potom pozostáva z vytvárania asociácií medzi neurónmi, odrážajúc tak neurónové spojenia medzi kortikálnymi regiónmi. Najnovšie znalosti o teórii talamu sa nachádzajú v [32]. Dynamika učenia je namodelovaná podľa hebbovskej metódy. Asociatívne učenie pojmov je teda nekontrolované učenie nad prúdom symbolov. Cieľom je naučenie sa znalostnej reprezentácie štruktúry do podoby sémantických sietí, prípadne hierarchií konceptov. Metóda je založená na indukcii asociácií medzi symbolmi vzhľadom na ich spoločné výskyty v kontextovom okne, cez ktoré samotné učenie prebieha. Nutná podmienka pre vytvorenie asociácie medzi párom tokenov je daná ich štatisticky nenáhodným spoločným výskytom. Tento sa zvyčajne implementuje podľa vzorca na výpočet signifikancie:

$$S(i, j) = \frac{p(i, j)}{p(i) \cdot p(j)} \quad (3.1)$$

kde i a j sú diskkrétne náhodné premenné. Vzájomná signifikancia $S(i, j)$ je potom definovaná ako podiel vzájomnej pravdepodobnosti symbolov (diskrétnych premenných) i, j a pravdepodobnosti ich výskytu. Ak hodnota $S(i, j)$ prekročí vopred určenú prahovú hodnotu, považujeme dané symboly za asociované. Váha asociácie medzi dvoma symbolmi i a j sa počíta iba v prípade, že sú dané symboly asociované a je daná vzorcom:

$$w(i, j) = \frac{p(i, j)}{p(j)} \quad (3.2)$$

Učenie prebieha napríklad nad predspracovaným textovým korpusom. V prvom kroku sa vytvorí lexikón známych symbolov. Podľa [43] 3500 najfrekventovanejších tokenov vo veľkom (napr. 200 MB) súbore textových dokumentov v prirodzenom jazyku (angličtine) pokrýva 90% všetkých tokenov v texte. V druhom kroku sa v textovom korpuse nahradia všetky symboly, ktoré sa nevyskytujú v lexikóne za značky označujúce neznámy symbol. Takto predspracovaný korpus slúži ako vstupný parameter pre učiaci algoritmus. Učenie prebieha v kontextovom okne, ktorého veľkosť je ďalším vstupným parametrom učenia. Podľa [31] pokrýva veľkosť kontextového okna $n=4$.



Obr. 3–6 Učenie v kontextovom okne

V kontextovom okne sa vypočítavajú početnosti spoločných výskytov tokenov, na ktoré počas učenia algoritmus narazil a tie poslúžia neskôr na výpočet pravdepodobností ich spoločných výskytov pre n kontextových vzdialeností, ktoré ďalej slúžia na výpočet signifikancií (3.1) a neskôr váh (3.2). Matice váh pre danú kontextovú vzdialenosť l pri určitom prahu sa nazývajú fascikle F_l . Učenie systému cez kontextové okno je znázornené na obr. 3–6.

3.5.1 Počítanie podobnosti slov

V [45] je popísaný postup počítania sémantickej príbuznosti dvoch slov pomocou asociatívneho učenia pojmov. S využitím nahromadených znalostí počas procesu učenia je potom možné vypočítať sémantickú podobnosť dvoch slov $R(x,y)$ podľa vzorca :

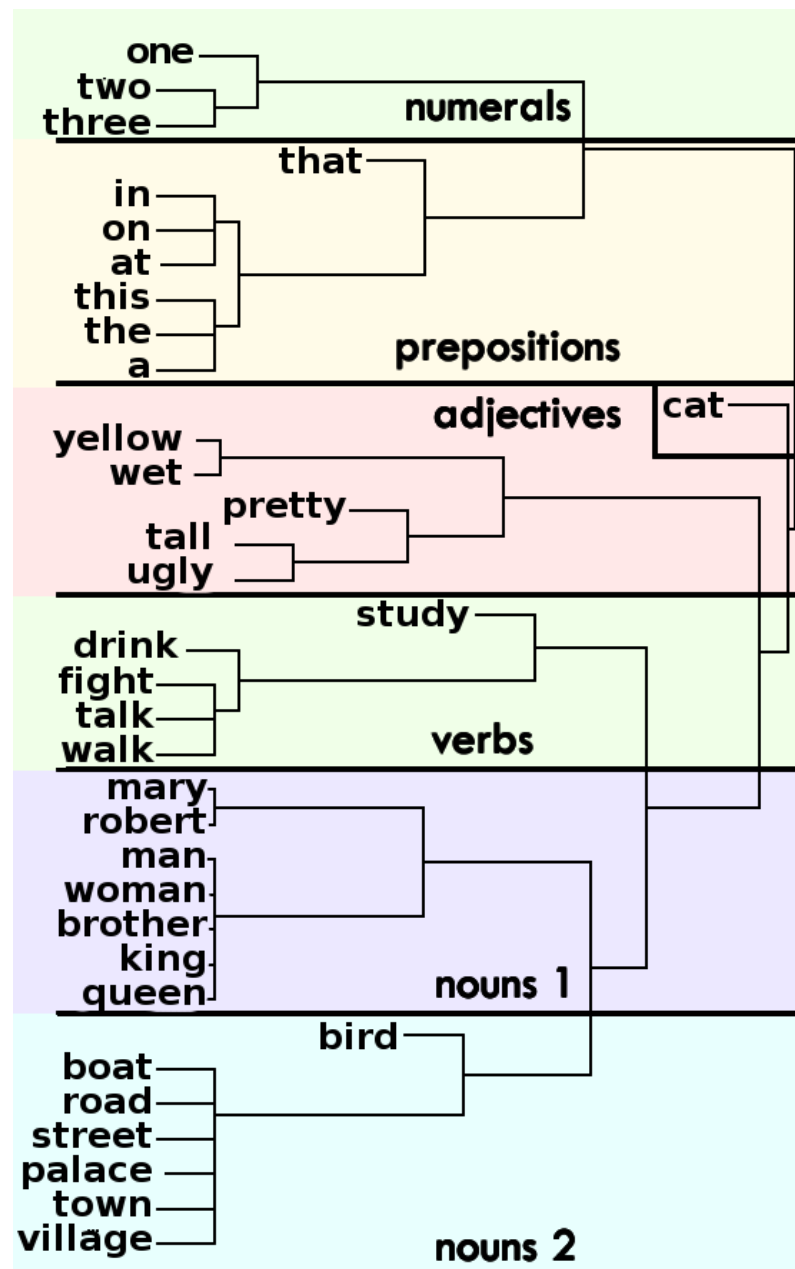
$$R(x,y) = \frac{O(x) \cap O(y)}{O(x) \cup O(y)} \quad (3.3)$$

kde $O(x)$ označuje sémantické okolie symbolu x z lexikónu L . $O(x)$ je taká množina symbolov, kde pre každý token t z lexikónu L existuje asociácia medzi tokenom t a symbolom x . Je teda množinou všetkých tokenov asociovaných s tokenom x . Asociácie sa vytvárajú až potom, ako signifikancia prekročí určitú vopred definovanú prahovú hodnotu. Pomocou týchto jednoduchých vzorcov autor vytvoril maticu podobností pre množinu tokenov, ktorá poslúžila ako vstup pre zhlukovací algoritmus. Jeho výstupom bol dendrogram – hierarchická štruktúra konceptov zoradených podľa podobnosti. Daný dendrogram je znázornený na obr. 3 – 7.

3.5.2 Výpočet konsezu

Konsenzus sa využíva na rozpoznávanie fráz a jeho popis je potrebný pre pochopenie procesu generovania synonym. Fráza je v [44] definovaná ako „syntakticky“ validný vstup rozpoznaný systémom. Výpočet sa vykonáva nad oknom (niekoľkých tokenov) a je vlastne hľadaním odpovede, ktorá najviac potvrdzuje najslabšiu hypotézu. Je dané okno n tokenov t_1, t_2, \dots, t_n . Maximálna kontextová vzdialenosť v takom okne bude $n - 1$. Posledný token v okne, t_n , nazvime dotazovacím tokenom. Všetky ostatné tokeny potom hlasujú cez relevantné fascikle $(F_1, F_2, \dots, F_{(n-1)})$ určené ich kontextovou vzdialenosťou a vytvárajú množiny odpovedí $M_1, M_2, \dots, M_{(n-1)}$. Prienik týchto množín potom tvorí množinu možných odpovedí v danom okne a nazývame ho „plná množina konsezov“:

$$t_n \in M; M = M_1 \cap M_2 \cap \dots \cap M_{(n-1)} \quad (3.4)$$



Obr. 3–7 Ukážka hierarchickej štruktúry vygenerovanej pomocou AUP

Množina M obsahuje m tokenov m_1, m_2, \dots, m_m , ktoré sú asociované s každým tokenom v okne cez príslušnú kontextovú vzdialenosť. Tieto asociácie majú obvykle rozdielne váhy (index u váhy vyjadruje, nad akým fasciklom bola vypočítaná). Aby sme podporili

najslabšiu hypotézu, musíme vyjadriť silu konsenzu $s(m_k)$:

$$s(m_k) = \min(w_{(n-1)}(t_1, m_k), w_{(n-2)}(t_2, m_k), \dots, w_1(t_{(n-1)}, m_k)); k = 1, 2, \dots, m \quad (3.5)$$

Odpoveď, ktorá potom najviac podporuje najslabšiu hypotézu, je potom t_{answer} :

$$t_{answer} = m_k; s(m_k) = \max(s(m_1), s(m_2), \dots, s(m_k)) \quad (3.6)$$

Pri učení autor použil okrem dopredných fasciklov aj fascikle inverzné. S inverznými fasciklami bolo možné vypočítať konsenzus pre dotazovaný token umiestnený na akomkoľvek políčku okna.

3.5.3 Vytváranie synonymým

V tejto práci sú synonymá definované ako tokeny prislúchajúce do nejakej konkrétnej triedy. Názov synonymum bol podľa autora najbližšou aproximáciou medzi existujúcimi názvami. Vstupom do algoritmu na výpočet synonymým je nami zvolený známy token. Výstupom algoritmu je množina tokenov, ktoré sú k tokenu zadanému syntakticky blízke, ale rozšírené aj o tokeny z jedného sémantického poľa, čo implicitne definuje samotnú triedu. Algoritmus na výpočet množiny synonymým:

1. Pre zadaný token \mathbf{T} vyber všetky asociácie z fasciklov F_{-2}, F_{-1}, F_1, F_2 a vlož ich do množín M_{-2}, M_{-1}, M_1, M_2 .
2. Nájdi úplnú množinu konsenzov za dotázaný token \mathbf{T} pre všetky kombinácie $M_{-2} \times M_{-1} \times T \times M_1 \times M_2$. Výsledok pridaj do množiny \mathbf{M} .
3. V množine \mathbf{M} sa nachádza množina všetkých synonymým k dotazovanému tokenu.

Systém sa učil nad množinou anglických textov pozostávajúcich z náhodne vybraných titulov beletrie z elektronickej knižnice gutenberg.org. Korpus pozostával z 1 GB textu v anglickom jazyku. Veľkosť lexikónu bola 3000 slov a prah pre jednotlivé príklady bol použitý v rozsahu 6.0 - 10.0 vzhľadom na mohutnosť výslednej množiny odpovedí s váhou konsenzu väčšou ako 0.001. Príklad výstupu zo systému vyzeral napríklad takto:

- moon → star, stars, moon, lights, bright
- white → brown, grey, black, white, red, yellow, blue, silk, gray, green, purple
- william → joseph, james, william, edward, smith, thomas
- tree → trees, plants, tree, cloud, plant, pine, branches, green, grass
- two → two, fifteen, eight, twelve, seven, three, forty, twenty, six, four, five, thirty

AUP sa ukázalo ako nádejný prístup, na ktorom je možné stavať. V budúcnosti by mohlo poslúžiť ako nástroj na automatizované budovanie ontológií z rozličných textových zdrojov.

4 Tézy dizertačnej práce

Problematika tvorby konceptuálnych sietí je pomerne široká a ponúka veľa priestoru na realizáciu. Predmetom dizertačnej práce by mal byť návrh modelu, implementácia a experimentálne overenie systému na tvorbu takýchto konceptuálnych sietí. Použité konceptuálne siete budú navrhnuté ako hybridné, kde jednou zo zložiek budú práve KG alebo siete ontologického typu, keďže majú všetky potrebné vlastnosti na zachytávanie znalostí z reálneho sveta [50]. Potenciálnymi prínosmi a zatiaľ tézami dizertačnej práce teda bude:

1. Vyhodnotenie metódy AUP z hľadiska jej možného využitia pre tvorbu konceptuálnych sietí z textov.
2. Návrh metódy na vytvorenie konceptuálnej siete s prihliadnutím na minimalizáciu zásahov človeka. Sieť bude zachytávať v texte relevantné koncepty a bude navrhovať a pomenovávať relácie, ktoré medzi nimi existujú.
3. Implementácia navrhutej metódy na vytvorenie konceptuálnej siete na základe existujúcej textovej informácie. Vstupom do systému bude text v prirodzenom jazyku a výstupom konkrétna znalostná sieť.
4. Ukážka použitia danej implementácie v rôznych doménových oblastiach a oblastiach záujmu.
5. Analýza vytváraných sietí, porovnanie navrhnutého a implementovaného prístupu s inými známymi prístupmi

Zoznam použitej literatúry

- [1] Abrahamson E. - Rosenkopf L.: *Social network effects on the extent of innovation diffusion: A computer simulation*, Organization Science 8(3), 289-309, 1997
- [2] Brachman R. J.: *On the Epistemological Status of Semantic Networks*, Associative Networks: Representation and Use of Knowledge by Computers, Academic Press, 3-50, 1979
- [3] Bruza P. D. - van der Weide Th.P.: *The Semantics of Data Flow Diagrams*, Proceedings of the International Conference on Management of Data, N. Prakash, Hyderabad, India, 1989
- [4] Carnap R. - Meyer W. H.: *Introduction to Symbolic Logic and Its Applications*, Dover Publications, 1958
- [5] Cao T.-H. - Sanderson A. C.: *Intelligent Task Planning Using Fuzzy Petri Nets*, Series in Intelligent Control and Intelligent Automation - Vol. 3, World Scientific Publishing Company, 1996
- [6] El Choubassi M. M. - El Khoury H. E. - Alagha C. E. J. - Skaf J. A. - Al-Alaoui M. A.: "Arabic Speech Recognition Using Recurrent Neural Networks," Proc. 3rd IEEE International Symposium on Signal Processing and Information Technology, IEEE ISSPIT 2003, Darmstadt, Germany, December 2003
- [7] Cimiano P. - Hotho A. - Staab S.: *Comparing conceptual, divisive and agglomerative clustering for learning taxonomies from text*, Proceedings of the European Conference on Artificial Intelligence (ECAI), 435-439, 2004
- [8] Cimiano P. - Hotho A. - Staab S.: *Learning concept hierarchies from text corpora using formal concept analysis*, Journal of Artificial Intelligence Research Vol. 24, 305 - 339, 2005

-
- [9] Dean T. - Kanazawa K.: *Persistence and probabilistic inference*, IEEE Transactions on Systems, Man, and Cybernetics, 19(3), 574-585, 1989
- [10] Doyle J.: *A Truth Maintenance System*, Artif. Intell. 12(3), 231-272, 1979
- [11] Fellbaum C.: *WordNet: An Electronic Lexical Database*, The MIT Press, Cambridge, MA, 1998
- [12] Frege G.: *Begriffsschrift: eine der arithmetischen nachgebildete Formelsprache des reinen Denkens.*, Halle, 1879
- [13] Friedman, N. - Linial, M. - Nachman, I.: *Using Bayesian networks to analyze gene expression data*, J. Comput. Biol., 7, 601–620, 2000
- [14] Fridman N. N. - Hafner C. D.: *The State of the Art in Ontology Design: A Survey and Comparative Review*, AI Magazine 18(3), 53-74, 1997
- [15] Furdík K.: *Získavanie informácií v prirodzenom jazyku s použitím hypertextových štruktúr*, Doktorandská dizertačná práca, Katedra kybernetiky a umelej inteligencie FEI, Technická univerzita v Košiciach, Košice, 2003
- [16] Gruber, T. R.: *Toward Principles for the Design of Ontologies Used for Knowledge Sharing*, International Journal Human-Computer Studies, 43(5-6), 907-928, 1995
- [17] Guarino, N. - Giaretta P.: *Ontologies and Knowledge Bases: Towards a Terminological Clarification*, In N. Mars (Ed.), *Towards Very Large Knowledge Bases: Knowledge Building and Knowledge Sharing*, Amsterdam. 1995
- [18] Janota, A.: *Bayesovské siete – rozšírené možnosti analýzy spoľahlivosti (1)*, A&TP Journal, č. 4, 2003, ISSN 1335-2237
- [19] Jelínek J.: *Identifikace tématických sociálních sítí*, In: *Znalosti 2008*, Bratislava, SK, Vyd. STU, 2008
- [20] Katrák M.: *Rozpoznávanie reči použitím neurónových sietí*, písomná práca k di-
-

zertačnej skúške, Katedra elektroniky a multimediálnych telekomunikácií, Fakulta elektrotechniky a informatiky, Technická univerzita v Košiciach, Košice, 2007

- [21] Koller D. - Pfeffer A.: *Object-oriented Bayesian networks*, In Proc. UAI, 1997
- [22] Krajčí S. - Krajčiová J.: *Sociálne siete a fuzzy formálna konceptová analýza*, In: Znalosti 2008, Bratislava, SK, Vyd. STU, 2008
- [23] Lee E. A. - Parks T. M.: *Dataflow Process Networks*, Proceedings of the IEEE, vol. 83(5), 773-801, 1995
- [24] Lee G. J. - Poole L.: *Diagnosis of TCP overlay connection failures using bayesian networks*, In Proceedings of the 2006 SIGCOMM Workshop on Mining Network Data (Pisa, Italy, September 11 - 15, 2006), MineNet '06. ACM, New York, 305-310. 2006
- [25] Mayer R.E.: *Models for understanding*, Review of Educational Research, 59 (1), 43-64, 1989
- [26] Meyers L.A.: *Contact network epidemiology: Bond percolation applied to infectious disease prediction and control*, Bulletin of the American Mathematical Society 44 (1), 63-86, 2007
- [27] Liao M.H.: *Chinese to English Machine Translation Using SNePS as an Interlingua*, M.A. Thesis, Technical Report 97-16, Department of Computer Science, State University of New York at Buffalo, Buffalo, NY, 1997
- [28] Gilad M.: *Source code retrieval using conceptual graphs*, Master's thesis, University of Amsterdam, 2004
- [29] Mozer M. C.: *Neural network music composition by prediction: Exploring the benefits of psychoacoustic constraints and multiscale processing*, Connection Science, vol. 6, 1994.

-
- [30] Murata T.: *Petri Nets: Properties, Analysis and Applications*, Proceedings of IEEE, vol. 77(4), 541-580, 1989
 - [31] Nielsen R. H.: *A Theory of the Cerebral Cortex*, ICONIP, 1998
 - [32] Nielsen R. H.: *Confabulation Theory: The Mechanism of Thought*, Springer, 2007
 - [33] Pearl J.: *Bayesian networks*, Tech. rep. R-216, Computer Science Department, University of California, 1994
 - [34] Pearl J.: *Causality: models, reasoning and inference*, Cambridge press, 2000
 - [35] Peleg M. - Rubin D. - Altman R. B.: *Using Petri Net Tools to Study Properties and Dynamics of Biological Systems*, Journal of the American Medical Informatics Association 12(2), 181-199, 2005
 - [36] Pérez U. A. - Sanchez E.: *Neural Networks Structure Optimization Through On-line Hardware Evolution*, World Congress on Neural Networks WCNN'96, San Diego, sept 15-18, 1996, 1041-1044
 - [37] Petri C. A.: *Kommunikation mit Automaten*, Bonn: Institut für Instrumentelle Mathematik, Schriften des IIM Nr. 2, 1962
 - [38] Rana O. F. - Shields M. S.: *Performance Analysis of Java Using Petri Nets*, Lecture Notes in Computer Science, Springer Berlin / Heidelberg, 2008
 - [39] Rapp Ch. - Evens M. - Garfield D.: *Design of an Emotion Profiler Using SNePS*, Proceedings of the first annual SNePS workshop on Current trends in SNePS—semantic network processing system, Buffalo, New York, United States, 1990
 - [40] Richens R.H. - *Preprogramming for mechanical translation*, Mechanical Translation, Vol. 3(1), 20-25, 1956
 - [41] Richta K.: *Universal Modeling Language UML*, In: Systems Integration 2003. Praha: Česká společnost pro systémovou integraci, 386-393, 2003, ISBN 80-245-0522-3
-

-
- [42] Ripley B.: *Pattern Recognition and Neural Networks*, Cambridge University Press, Cambridge, 1996
- [43] V. Ročkai, 2005. *Mining of Concepts and Semantic Relations from Texts in Natural Language* Diplomová práca, Katedra kybernetiky a umelej inteligencie, Fakulta elektrotechniky a informatiky, Technická univerzita, Košice, 2005
- [44] V. Ročkai, 2007. *Automatické generovanie slov patriacich do jednej syntakticko-sémantickej triedy*, WIKT 2007 - Proceedings of 2nd Workshop on Intelligent and Knowledge oriented Technologies (Košice, November 15 - 16, 2007). Centre for Information Technologies, FEI TU Košice, Slovakia, 2008
- [45] V. Ročkai, 2008. *Asociatívne učenie pojmov*, In: Znalosti 2008, Bratislava, SK, Vyd. STU, 2008
- [46] Sanchez O. - Poesio M.: *Acquiring Bayesian Networks from Text*, In the 4th LREC Preceedings, Lisboan, Portugal, 2004
- [47] Schreiber G. - Wielinga B. - Jansweijer W.: *The KAKTUS View on the 'O' Word*, In Proceedings of IJCAI95 Workshop on Basic Onto-logical Issues in Knowledge Sharing. Montreal, Canada, 1995
- [48] Shapiro S. C.: *Case Studies of SNePS*, ACM SIGART Bulletin 2 (3), 128-134, 1991
- [49] Sklenák V., *Sémantický web*, In INFORUM 2003 [online], Praha : Albertina icome Praha, 2003
- [50] Sowa J. F. - Eileen C. W.: *Implementing a semantic interpreter using conceptual graphs*, IBM Journal of Research and Development 30(1), 57-69, 1986
- [51] Sowa J. F.: *Conceptual Structures: Information Processing in Mind and Machine*, Addison-Wesley, Reading, MA, 1984
- [52] Sowa J.F., 1987. *Semantic networks*, Encyclopedia of Artificial Intelligence, edited

by S. C. Shapiro, Wiley, New York, 1987; revised and extended for the second edition, 1992

- [53] Strehovský M.: *Moderní metody získávání a zpracování biomedicínských dat*, Bakalárska diplomová práca, Masarykova univerzita, Brno, 2008
- [54] Uschold U. - Gruninger M.: *Ontologies: Principles, Methods, and Applications*, Knowledge Eng. Rev., vol. 11(2), 1996
- [55] Vanderdonckt J. - Tarby J. C. - Derycke A.: *Using Data Flow Diagrams for Supporting Task Models*, DSV-IS (2), 1998
- [56] Open Cyc, *Open Cyc*, Dostupné online: <http://www.opencyc.org/>
- [57] Wordnet, *Wordnet online*, Dostupné online: <http://wordnet.princeton.edu/perl/webwn>
- [58] Sowa J.F.: *Conceptual graphs tutorial*, Dostupné online: <http://www.huminf.aau.dk/cg/index.html>
- [59] Sowa J.F.: *Knowledge Representation*, Porfýriov strom, Dostupne online: <http://www.jfsowa.com/krbook/>