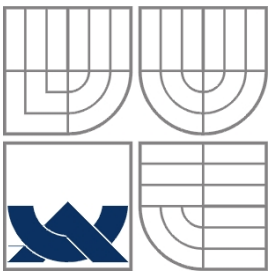
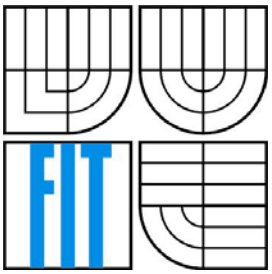


VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
BRNO UNIVERSITY OF TECHNOLOGY



FAKULTA INFORMAČNÍCH TECHNOLOGIÍ
ÚSTAV INFORMAČNÍCH SYSTÉMŮ

FACULTY OF INFORMATION TECHNOLOGY
DEPARTMENT OF INFORMATION SYSTEMS

METODY DOLOVÁNÍ V GRAFECH A JEJICH VYUŽITÍ V PROSTŘEDÍ IS PRO HISTORIKY

METHODS FOR GRAPH MINING AND THEIR USE IN AN IS FOR HISTORIANS

DIPLOMOVÁ PRÁCE
MASTER'S THESIS

AUTOR PRÁCE
AUTHOR

RADIM KAPAVÍK

VEDOUCÍ PRÁCE
SUPERVISOR

Ing. VLADIMÍR BARTÍK, Ph.D.

BRNO 2010

Zadání diplomové práce/10373/2009/xkapav01

Vysoké učení technické v Brně - Fakulta informačních technologií

Ústav informačních systémů Akademický rok 2009/2010

Zadání diplomové práce

Řešitel: **Kapavík Radim, Bc.**
Obor: Informační systémy
Téma: **Metody dolování v grafech a jejich využití v prostředí IS pro historiky**
Methods for Graph Mining and Their Use in an IS for Historians
Kategorie: Databáze

Pokyny:

1. Analyzujte požadavky na informační systém pro ukládání informací o historii s datovým modelem založeným na sémantické síti.
2. Prostudujte problematiku analýzy grafů včetně možností využití metod získávání znalostí pro grafy.
3. Definujte úlohy s využitím dolování v grafech na datech v navrženém informačním systému.
4. Implementuje navržený informační systém, včetně metody pro dolování dat, dohodnuté s vedoucím práce.
5. Ověřte funkčnost informačního systému i metody pro dolování. Proveďte experimenty a srovnání s jinými metodami.
6. Zhodnoťte dosažené výsledky a další možná rozšíření tohoto projektu.

Literatura:

- Han, J., Kamber, M.: Data Mining - Concepts and Techniques, 2nd Edition. Morgan Kaufmann Publishers, 2006.

Při obhajobě semestrální části diplomového projektu je požadováno:

- Body 1-3.

Podrobné závazné pokyny pro vypracování diplomové práce naleznete na adrese http://www.fit.vutbr.cz/info/szz/

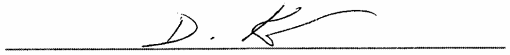
Technická zpráva diplomové práce musí obsahovat formulaci cíle, charakteristiku současného stavu, teoretická a odborná východiska řešených problémů a specifikaci etap, které byly vyřešeny v rámci ročníkového a semestrálního projektu (30 až 40% celkového rozsahu technické zprávy).

Student odevzdá v jednom výtisku technickou zprávu a v elektronické podobě zdrojový text technické zprávy, úplnou programovou dokumentaci a zdrojové texty programů. Informace v elektronické podobě budou uloženy na standardním nepřepisovatelném paměťovém médiu (CD-R, DVD-R, apod.), které bude vloženo do písemné zprávy tak, aby nemohlo dojít k jeho ztrátě při běžné manipulaci.

Vedoucí: **Bartík Vladimír, Ing., Ph.D.**, UIFS FIT VUT
Datum zadání: 21. září 2009
Datum odevzdání: 26. května 2010

VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
Fakulta informačních technologií
Ústav informačních systémů
612 66 Brno, Božetěchova 2

L.S.



doc. Dr. Ing. Dušan Kolář
vedoucí ústavu

Abstrakt

Tato diplomová práce představuje návrh a implementaci webového informačního systému pro historiky (Elektronická encyklopedie historie) s datovým modelem založeným na sémantické síti a několika nástrojů umožňujících analýzy dat v něm založených na grafových algoritmech a metodách dolování v grafech.

Abstract

This diploma work presents design and implementation of web information system for historians (called Electronic Encyclopaedia of History) with data model based on semantic network and of several tools for data analysis in such IS, using graph algorithms and graph mining methods.

Klíčová slova

Encyklopedie historie, sémantická síť, analýza grafů, dolování dat z grafů

Keywords

Encyclopaedia of history, semantic network, graph analysis, graph mining

Metody dolování v grafech a jejich využití v prostředí IS pro historiky

Prohlášení

Prohlašuji, že jsem tuto diplomovou práci vypracoval samostatně pod vedením Ing. Vladimíra Bartíka, Ph.D
Uvedl jsem všechny literární prameny a publikace, ze kterých jsem čerpal.

……
Radim Kapavík
25.5.2010

Poděkování

Děkuji Jiřímu Vrbovi za dlouhodobou spolupráci na návrhu datového modelu a na testování výsledné aplikace a Martinu Bulínovi za vytvoření designu webu, který je výsledkem této práce.

Obsah

Úvod

Na humanitní a přírodní (technické) vědy se často nahlíží jako na dva různé světy, někdy dokonce jako na světy protikladné. Tak tomu pochopitelně není a právě informatika se často považuje za vědu na hranici technických a humanitních věd. Jisté je, že jejich výsledků mohou využívat všechny ostatní vědní obory, ať už technické, nebo humanitní. Právě humanitní vědy ji však využívají v míře menší než by bylo prospěšné pro obě strany a v některých případech se jejímu využití dokonce aktivně brání[6]. Historie, jako jedna z věd používajících nejkonzervativnějších metod a postupů, mezi takovéto rozhodně patří.

Informatika a výpočetní technika má však historii co nabídnout. Rozvoj internetu přispěl k nebývalému zvýšení přístupnosti informací, tedy i historických, není výjimkou, že mnohé prameny jsou dnes dostupné elektronicky každému, kdo se dovede připojit k síti. V knihovnách dnes papírové kartotéční lístky slouží většinou jen jako dekorace, skutečné vyhledávání probíhá přes informační systémy. Málokterý historik spisuje své texty ručně nebo na psacím stroji… To všechno jsou však případy, kdy historici pro svou práci využili produktů informatiky, které nebyly vyvinuty speciálně pro ně*.

V této práci představím návrh aplikace nazvané Elektronická encyklopedie historie, speciálního informačního systému určeného pro historiky jakožto nástroj umožňující jim efektivnější provádění činnosti, která je podstatou historikova řemesla – historického bádání v nejvlastnějším slova smyslu.

První kapitola uvádí čtenáře do problematiky použití informatiky pro účely historické vědy. Druhá kapitola popisuje, co by vlastně Elektronická encyklopedie měla být a jaký je její smysl. Třetí kapitola pojednává o teorii sémantických sítí, neboť data budou v encyklopedii uložena právě ve formě sémantické sítě. Ve čtvrté kapitole je specifikován návrh aplikace (počínaje neformální specifikací) s využitím standardních návrhových prostředků. Pátá kapitola představuje návrh možností několika druhů analýzy dat v popisovaném systému. Šestá kapitola je věnována popisu implementace celé aplikace. Sedmá kapitola prezentuje výsledky testování aplikace. Závěrečná kapitola pak obsahuje celkové zhodnocení projektu a výhled do jeho budoucnosti.

* Snad jedině knihovnický informační systém by se dal považovat za aplikaci vyvinutou speciálně pro knihovnické účely (podobně jako např. bankovní informační systém byl vyvinut pro bankovnictví atd.).

1 Informatika ve vědě o historii

Stopovat podrobně využití poznatků a postupů informatiky historiky při jejich bádání není účelem této práce. V rámci uvedení do problematiky však považuji za vhodné zařadit tuto kapitolu, která ve stručnosti popisuje, jaké možnosti využití informatiky historiky vůbec existují, a dále krátce představuje některé konkrétní projekty z této oblasti.

1.1 Možnosti využití informatiky historiky

To, pro co lze asi jako nejvýstižnější použít anglický název „historical information science“[*], souvisí jednak se sociologickými a ekonomickými statistickými analýzami, a jednak s analýzou textů pomocí počítače. Výrazným impulsem v tomto směru je rozvoj internetu, který vedl k tomu, že v 90. letech začínají být digitalizovány sbírky některých světových archivů a muzeí, které tak mohou být badatelům přístupny vzdáleně.

Zavádění výpočetní techniky do oboru humanitních věd však nebylo přijato tak pozitivně, jako v jiných oborech, a někteří humanisticky zaměření badatelé o jejich přínosu dodnes pochybují. A tak se stalo, že až dodnes jsou tyto prostředky ze strany historiků využívány pouze sporadicky a nezpůsobily zásadní změnu v přístupu k bádání, ale pouze přiměly badatele přizpůsobit se novým technologiím na té nejnižší možné úrovni. Historik samozřejmě běžně píše na PC, vyhledává na internetu či v elektronickém katalogu knihovny a komunikuje pomocí e-mailu, ale k využití výpočetní techniky na hlubší úrovni dochází spíše výjimečně. Slibně se rozvíjející směr vývoje tak brzy poněkud ustrnul. Nestalo se tak proto, že „by snaha výpočetní techniky a informatiky poskytnout historii nástroje a metody, které by historikové mohli používat k rozšíření možností svého výzkumu a jeho zkvalitnění, byla neúspěšná, ale proto, že historie nedokázala využít mnoho z nástrojů, které výpočetní technika a informatika nabízí“[6].

Těžko spekulovat, co je důvodem odporu klasických historiků k významnějšímu využití poznatků informatiky a výpočetní techniky ve svém oboru. Je-li však tento odpor realitou, pak nezbývá než souhlasit, že „tradiční rozdělení práce mezi historiky a informatiky nevede k uspokojivým výsledkům pro ani jedny z nich“ a proto „by bylo třeba, aby vznikla nová generace odborníků vzdělaných jak v oboru historie, tak informatiky, konkrétně programování, databází a multimédií“.[6] Několik institucí, které se o naplnění této vize snaží, ve světě v současné době existuje, v České republice však žádná. Podívejme se proto nejdříve v obecné rovině na to, v jakých oblastech může informatika historii být nápomocná.

1) Digitalizace zdrojů – Do této oblasti se soustřeďuje většina úsilí, pokud se už nějaké na projekty historical information science vynakládá. Krom technických zařízení určených ke skenování, která jsou již dnes na dostatečné úrovni, aby potřeby historiků v této oblasti uspokojila, jde především o oblast OCR (Optical Character Recognition, optické rozpoznávání znaků) a následně problematiku

[*] Tento název, navrhovaný v [6], by měl vyjadřovat, že nejde ani o historii informatiky, jak naznačují některé jiné názvy (historical computing), ani o pouhé složení dvou oddělených oborů (jak naznačuje dříve používaný výraz computing and history), nýbrž o specifický obor, který autoři definují jako „Vědu s vlastní metodologií. Jejím předmětem jsou historické informace a nejrůznější způsoby, jak historickou informaci vytvořit, […] analyzovat a prezentovat s pomocí informačních technologií“. Do češtiny nelze tento pojem přeložit se zachováním přesného významu, proto ho budu v této kapitole používat v originálním anglickém tvaru.

zpracování a analýzy přirozeného jazyka. Celkovým záměrem v této oblasti je vytvořit systém schopný automatického zpracování předloženého textu od skenování až po zařazení do nějakého systému či databáze informací či znalostí, s čímž souvisí také problémy identifikace shodných a rozdílných objektů, které jsou ve zpracovávaném textu zmiňovány, automatické strukturování získaných informací apod.[6]. V praxi digitalizace zdrojů často končí jejich naskenováním a zpřístupněním na internetu, které je velice užitečné, ale novou metodu výzkumu na něm postavit nelze.

2) 3D modelování – Techniky modelování reality umožňují i historikům vytvářet 3D modely předmětů, ale i míst ve stavu, v jakém se nacházely v minulosti. Tento atraktivní způsob prezentace výsledků historického bádání je v současnosti jedním z obvyklých prvků moderních muzeí (zde jsou na čele tohoto směru vývoje archeologové, resp. muzea archeologická – u nás např. Mikulčice), ale sám o sobě může být těžko považován za kvalitativní přínos historickému bádání.

3) Využití GIS – Systémy GIS jsou využívány historickou obcí ke dvěma účelům. Prvním je vytváření historických atlasů na podkladě historických map, k němuž jsou používány standardní GIS systémy. Nikoliv bezvýznamným problémem, který je v této oblasti potřeba řešit, je podpora vkládání časových údajů do systémů GIS (tzv. tempoval GIS)[7]. Druhou oblastí využití GIS systémů v historii je archeologie. Archeologové jsou mezi historiky tou skupinou, která výpočetní techniku používá v největší míře*. Archeologové využívají specializované GIS systémy dvěma způsoby – za prvé k mapování jednotlivých archeologických nalezišť a k uchování přesných informací o poloze nálezů, které jsou přemístěny, a za druhé k mapování rozložení množství nalezišť v prostoru a k jeho analýze s ohledem na pravděpodobná dosud neprozkoumaná naleziště. Velkým rozdílem oproti snaze popsané výše je v případě archeologie fakt, že ta se nepokouší o nějaké automatické převádění existujících údajů do digitální podoby, ale digitální model vytváří zadáváním údajů přímo do něj, což považuji za perspektivnější než výše uvedenou snahu po automatickém digitalizování. Využití GIS systémů archeology lze označit za kvalitativně nový způsob historického bádání.

4) Analýza historických dat – výpočetní technika umožňuje díky možnosti digitalizace provádět statistické analýzy velkého množství dat týkajících se historie (typicky registr obyvatel, vojáků), což bývá v současné době historiky v zahraničí už častěji využíváno než dříve (o žádném projektu tohoto typu u nás mi není nic známo). Díky nástrojům informatiky (např. metodám získávání znalostí) je však možno provádět i složitější a komplexnější analýzy dat, než pouze statistické. Tyto možnosti jsou ze strany historiků zatím téměř zcela nevyužity. Přitom analýza historických dat s pomocí výpočetní techniky nabízí, podobně jako využití GIS v archeologii, možnosti skutečně nových badatelských postupů.

1.2 Podobné projekty

V této podkapitole bych rád představil některé konkrétní existující projekty realizující výše uvedené možnosti s tím, že vybírám ty, které vykazují podobnost se zamýšleným charakterem Elektronické encyklopedie historie (který je podrobně popsán v kapitole 2), a upozorňuji, ve kterých ohledech by se Elektronická encyklopedie historie měla od těchto projektů lišit a proč.

* Snad to souvisí s tím, že podle některých, jako např. Michaela Foucaulta (viz jeho Archeologie vědění), je archeologie nejvědečtější ze všech historických disciplín a její metody by měly sloužit za vzor ostatním historikům

1.2.1 Wikipedia

Žádný podobný projekt se nemůže vyhnout srovnání s nejoblíbenější internetovou encyklopedií současnosti. Protože koncept Wikipedie nepovažuji za nutné představovat, zaměřím se pouze na výčet těch vlastností systému Wikipedie, které ji dělají pro zamýšlený projekt nepoužitelnou.

Základním nedostatkem pochopitelně je to, že Wikipedia nedisponuje datovým modelem založeným na sémantické síti, resp. nedisponuje datovým modelem prakticky žádným. Tudíž veškeré přehledy, seznamy či tabulky je nutno vytvářet ručně, stejně tak provazování jednotlivých záznamů (které se děje pouze pomocí hypertextových odkazů) je nutno provádět ručně, vyhledávat v jejím obsahu lze jen fulltextově a možnosti analýz vložených dat jsou velmi omezené.

Významný koncept, který Elektronická encyklopedie historie nehodlá přijmout, a který Wikipedii dělá Wikipedií, je právě princip tzv. wiki – tedy umožnění komukoliv přidávat do encyklopedie nové záznamy nebo libovolně měnit stávající. Tento systém vyžaduje neustálé zásahy a velké množství redakční práce od správců celého projektu, což je něco, s čím zamýšlená aplikace nepočítá. Zatímco obsah Elektronické encyklopedie historie by měl být stejně jako u Wikipedie otevřený a přístupný komukoliv, možnost měnit jej musí zůstat vyhrazena jen registrovaným redaktorům, aby bylo zaručeno transparentní autorství jednotlivých záznamů a tedy zodpovědnost za konkrétní obsah; tedy ani registrovaný autor nesmí mít možnost zasahovat do příspěvků jiného autora – natožpak libovolný návštěvník.

1.2.2 Valka.cz*

Nejrozsáhlejším a nejpopulárnějším českým projektem podobného druhu je valka.cz, server zaměřený primárně, ale ne výhradně, na vojenskou historii. V jeho rámci je provozována encyklopedie informací o vojenské historii (osobnosti, technika, jednotky atd.), která se vyvinula původně z diskusního fora. Ačkoliv některé prvky (tabulky údajů, odkazy mezi záznamy, strukturovaný obsah encyklopedie) vzbuzují na první pohled zdání, že data by mohla být uložena v nějakém druhu sémantické sítě, není tomu tak. Systém této encyklopedie je stále založen na fóru, které není už z velké části používáno ke svému původnímu účelu, ale právě k vytváření databáze informací. Data jsou uložena jako jednotlivé příspěvky v databázi sloužící pro ukládání příspěvků v diskusích, veškeré tabulky a další podobné prvky jsou vytvářeny pomocí formátovacích značek uvnitř uložených textů a veškeré provazování příspěvků mezi sebou se děje ručním vkládáním hypertextových odkazů. Podobně jako u Wikipedie, údržba tohoto systému si vyžaduje velké množství redakční práce od dobrovolníků starajících se o obsah encyklopedie.

Ostatní vlastnosti tohoto systému v obecné rovině vesměs odpovídají zamýšlené Elektronické encyklopedii historie – systém je snadný na ovládání, veškerý obsah encyklopedie je volně přístupný, vkládat a měnit data mohou pouze registrovaní uživatelé, kterých se na jednom příspěvku může podílet více, ale nemohou si při tom zasahovat do svých textů (rozšíření záznamu jiného autora se dosáhne přidáním dalšího příspěvku do daného tématu).

Jednou z možností budoucího vývoje tohoto projektu je právě přechod k datovému modelu založenému na sémantické síti, což s sebou však nese obrovské problémy s přesunem veškerého obsahu z nestrukturované a často nesjednocené formy do sémantické sítě. Pokud k tomu dojde, bude

* www.valka.cz, resp. http://forum.valka.cz/ – databáze informací se ze setrvačnosti – a také proto, že je smíšena se skutečným diskusním fórem – stále nazývá fórem. Projekt byl založen a je veden soukromou osobou za pomoci skupiny nadšenců, do níž měl autor tu čest jistou dobu také patřit. Právě zkušenosti z tohoto projektu stály u zrodu nápadu vytvořit Elektronickou encyklopedii historie.

to – i s poměrně rozsáhlým zázemím dobrovolníků, ochotných dělat i „černou práci“ – běh na velmi dlouhou trať.

1.2.3 Freebase*

Freebase je jeden z několika projektů, jehož cílem je vytvořit sémantickou obdobu Wikipedie. Disponuje tudíž, na rozdíl od ní, datovým modelem založeným na sémantické síti. Uživatel může procházet a zobrazovat obsah sítě, vyhledávat v něm fulltextově, ale také tvořit a měnit obsah – Freebase zachovává princip wiki (viz výše), což je první rozpor s požadavky na Elektronickou encyklopedii historie.

Freebase dává uživateli přístup na úroveň datového modelu – uživatel si může vytvářet a upravovat typy a třídy objektů a vztahů mezi nimi a vytvářet si tak svůj vlastní datový model (zde není princip wiki dodržován, neboť uživatelé si vzájemně nemají právo zasahovat do svých datových modelů). To vyžaduje, aby sami uživatelé byli schopni udržovat datový model smysluplný a konzistentní (Freebase jim k tomu poskytuje několik nástrojů, které jim to usnadní). Navíc se tím přesunuje odpovědnost za vytváření zobrazovacích nástrojů (tzv. pohledů – views) a případných analytických nástrojů na uživatele, neboť neexistuje žádný administrátor systému, který by znal datový model (ten se navíc může v průběhu času měnit) a mohl tudíž takovéto nástroje obecně vytvářet.

Freebase nabízí základní vyhledávací rozhraní, základní editor a základní zobrazovací nástroj (prostý výpis objektu, jeho vlastností a vazeb) a nástroje pro tvorbu vlastního API. Ty využívají dotazovacího jazyka MQL, který umožňuje přístup do obsahu databáze Freebase. Jazyk MQL, využívající formátu JSON (JavaScript Object Notation), lze využít jak ke čtení dat z databáze, tak k jejich zápisu. Dotaz v tomto jazyce lze na server odeslat buď přímo v rámci HTTP hlavičky**, nebo přes platformu Acre, která umožňuje umisťovat přímo na server Freebase HTML stránky, v jejichž rámci je možno využívat jednak javascript a jednak jakýsi „acre-skript“, který je zpracován na serveru a umožňuje přímé zadávání MQL dotazů v rámci kódu stránky a možnost zobrazení výsledků na určeném místě stránky.

Práce s těmito nástroji vyžaduje nemalou programátorskou zdatnost (znalost HTML a souvisejících technologií, schopnost naučit se rozumět MQL, schopnost zpracovat výsledek dotazu…) a navíc zůstává otevřená otázka, jak se vypořádat s měnícím se datovým modelem. Elektronická encyklopedie historie proto hodlá jít cestou centrální správy datového modelu, který bude v rukou tvůrce a který bude vyvíjen a udržován společně se systémem samotným.

1.2.4 Histcross***

Projekt s názvem Histcross – the Semantic Database for Historians je ze všech zde uváděných projektů nejbližší tomu, čím by měla být zamýšlená Elektronická encyklopedie historie. Jedná se o implementaci sémantické sítě použitou původně pro jeden konkrétní projekt (Zkoumání obchodních vazeb mezi Evropou a Asií v 16. století), schopnou však širšího využití. Uživatel může procházet a zobrazovat obsah sítě, vyhledávat v něm fulltextově nebo zadáváním komplexních dotazů nad

---

* http://www.freebase.com/, vyvíjená skupinou Metaweb Technologies. Stručný popis projektu je k dispozici na http://wiki.freebase.com, dokumentace k projektu je dostupná na adrese http://www.freebase.com/docs/.

** jako požadavek typu GET na http://www.freebase.com/api/service/mqlread?query=[dotaz v jazyce MQL]

*** http://www.histcross.org/ - autorem je Maxmilian Kalus z Friedrich-Schiller-Universität v Jeně (Německo). Oproti Freebase je projekt Histcross mnohem komornějším.

sémantickou sítí, registrovaný uživatel pak měnit její obsah. Většina principů, na kterých je Histcross postaven, v zásadě odpovídá principům aplikace představované v této práci (viz kapitola 2 a 3), zaměřím se proto na těch několik odlišností, které lze vypozorovat.

Nejvýznamnější z nich je skutečnost, že i Histcross dává uživateli přístup na úroveň datového modelu – uživatel může zavádět a upravovat typy a třídy objektů a vztahů mezi nimi; tím pádem si uživatelé v rámci konkrétní instance Histcrossu definují svůj vlastní datový model. Tento přístup navíc v případě Histcrossu znemožňuje jakékoliv případné specifické analýzy dat v systému uložených (tedy analýzy, k jejichž provedení je potřeba znalosti sémantiky dat), protože tvůrce systému nezná datový model, který si uživatelé vytvoří (a který se navíc postupem času může měnit), a nemůže proto žádné takovéto analytické nástroje vytvářet, a uživatel, i kdyby toho byl schopen, k tomu zase nedisponuje potřebnými nástroji.

Dalšími dvěma méně zásadními rozdíly jsou u histcrossu prakticky neexistující uživatelské rozhraní, které by jistě bylo nutno dotvořit v případě, že by ho měli využívat ve větším měřítku historikové ne-informatici, a fakt, že Histcross nepodporuje práci s netextovým obsahem, jako například fotografiemi, mapami, plánky apod., což by nemělo v zamýšlené aplikaci chybět. Na rozdíl od Freebase si ale Histcross neklade za cíl stát se jednou velkou univerzální encyklopedií (ve stylu Wikipedie), ale spíše být k dispozici pro jednotlivé menší projekty. Zamýšlená Elektronická encyklopedie historie by sice měla být teoreticky schopna stát se univerzální encyklopedií historie, ale v praxi rozhodně minimálně v první fázi své existence půjde cestou využití v rámci menších ohraničených projektů. Na rozdíl od Histcrossu však nepočítá s existencí více instancí aplikace, ale s jednou, a to především kvůli centrálnímu uložení dat, které je potřebné k jejich provazování.

2 Elektronická encyklopedie historie

Po krátkém představení možností využití informatiky v oboru historie a projektů, které se pokoušejí o něco podobného jako Elektronická encyklopedie historie, je čas konečně si představit, čím by Elektronická encyklopedie historie měla být.

2.1 Smysl Elektronické encyklopedie historie

K vysvětlení smyslu Elektronické encyklopedie historie (dále také EE) je nutný stručný úvod týkající se zkoumání historie.

To, co se většinou myslí obecným pojmem „historie“ nebo „zkoumání historie“, je možno rozčlenit ve tři fáze:

1) samotná historie (myšleno věda o historii)
2) aplikace historických poznatků
3) užívání historických poznatků

Důvodem, proč zde uvádím toto členění, je uvědomit si fakt, že skutečnou vědou ve vlastním smyslu tohoto pojmu je pouze první z těchto tří fází. Její aplikace (2. fáze) je analogií technologie (tedy použití vědeckých poznatků v praxi) a užívání (3. fáze) je pak už úplně něčím jiným (odpovídá využívání technického prostředku v praxi).[*] To je tedy základním východiskem filosofie návrhu EE – Elektronická encyklopedie historie je zamýšlena jako nástroj právě a výhradně pro první (vědeckou) fázi historického zkoumání.

Elektronická encyklopedie má umožnit historikům, aby při své práci využili některé výhody výpočetní techniky. Historikova práce vždy začíná shromážděním pramenů k vybranému tématu (heuristika) a jejich kritikou, a pokračuje „vytěžením“ těchto pramenů – historik si na jejich základě sestavuje nejpravděpodobnější obraz o tom, jak se „to“ odehrálo. Formálněji řečeno vytváří si určitý model minulé reality, který je sestaven na základě stop této reality, které se dochovaly do současnosti. Tento model posléze prezentuje, vyvozuje z něj závěry, publikuje apod. Smyslem EE je umožnit historikovi efektivnější práci s oním modelem minulé reality.

Klasickými prostředky, které historik při vytváření modelu využívá, jsou poznámky, výtahy z pramenů a vlastní paměť, která uchovává nejen část informací, ale především jejich souvislosti a význam. Tyto tradiční postupy mají jednu velkou a jednu menší nevýhodu. Tou menší nevýhodou je roztříštěnost reprezentace takovéhoto modelu – jeho část se nachází přímo v pramenech, část v poznámkách a část v mysli historika. Zvládání těchto potíží však tvoří významnou část know-how každého historika a domnívám se, že dobrý historik se s nimi dokáže vypořádat. Tou větší nevýhodou, se kterou se nevypořádá sebelepší historik, je však nepřenositelnost takovéhoto modelu.

Reprezentace tohoto modelu je čistě individuální záležitostí každého historika. I kdybychom připustili, že je snad teoreticky možné veškeré informace reprezentovat fyzicky (tedy nemít ani část

[*] Takovéto členění jsem v žádné teoreticko-historické práci nenašel a samotná historická obec nemá jasno, co se pojmem historie má správně rozumět. Často se do tohoto pojmu proto zahrnují všechny tři uvedené fáze a způsobují tím nedorozumění mezi svou komunitou a nehistoriky, neboť jen těžko může být aplikace vědeckých poznatků v praxi, tím méně užívání výsledků takovéto aplikace, vědou. Podrobnější pojednání o těchto problémech však není součástí této práce.

informací pouze ve vlastní paměti), stále tu zůstává ta snad nejdůležitější část modelu, kterou lze z historikovy mysli „vytáhnout“ velmi těžko, ba je to snad zcela nemožné – a tou jsou souvislosti a význam jednotlivých informací. Díky tomu je model v podstatě nepřenosný, nelze ho sdílet s jinými historiky, ba dokonce často ani historik sám ho po delší době od výzkumu uplynulé není schopen ze svých záznamů rekonstruovat v té podobě, v jaké existoval v době, kdy se výzkumem zabýval. Výstupem historikovy práce je totiž klasický text, tedy jakýsi lineární popis, kterým mnohodimenzionální model není možné zachytit. Navíc historik často řeší nějakou konkrétní otázku a jeho cílem ani není sdělovat ostatním celý svůj model, nýbrž jen určité konkrétní informace, které z něj získal. Důsledkem pak je, že mnoho historiků, kteří se zabývají podobnými nebo příbuznými tématy, si potřebné modely vytvářejí mnohonásobně (každý svůj), přičemž tyto se mnohdy z větší či menší části překrývají.

Elektronická encyklopedie proto nabízí badatelům v oblasti historie nástroj pro tvorbu těchto modelů, který by měl, aspoň částečně, odstranit oba zmíněné problémy. Díky centralizaci informací na jednom místě, dostupném navíc v dnešní době prakticky odkudkoliv (je-li k dispozici internetové připojení), odstraňuje roztříštěnost reprezentace historikova modelu. Navíc umožňuje jeho sdílení mezi více lidmi. Aby toho bylo dosaženo, je nutné vytvořit odpovídající datový model, který umožní zachytit co nejpřesněji souvislosti a sémantiku dat, tak aby tento model mohl historikům plnohodnotně nahradit jejich myšlený model. To je asi nejsložitější problém celé Elektronické encyklopedie, a nelze očekávat, že se ho podaří v krátké době a definitivně vyřešit. Ten však není předmětem této práce – tím je spíše představit celkový koncept a ukázat jeho možné využití.

2.1.1 Požadavky na aplikaci

Obecné požadavky na EE by se daly shrnout do několika bodů:

- možnost spolupráce více autorů na tvorbě obsahu encyklopedie – znamená centrální ukládání dat a jejich sdílení mezi všemi autory, ovšem se zachováním autorství jednotlivých záznamů
- snadné a lehce přístupné vkládání údajů – znamená jednak že autor nebude nucen instalovat si žádnou speciální aplikaci, a jednak že nebude počítáno s existencí (lidského) editora, který by autory vkládaná data musel jakkoliv zpracovávat nebo technicky či jinak upravovat
- efektivní využití vložených dat – znamená možnost automaticky generovat nejrůznější přehledy, seznamy, atd. na základě dat v encyklopedii se nacházejících; pokud jsou v encyklopedii zadány všechny údaje potřebné k vytvoření určitého přehledu, musí být možné takový přehled skutečně vytvořit – jedná se vlastně o požadavek na vhodný datový model pro ukládání obsahu encyklopedie
- možnost přímé prezentace obsahu encyklopedie na internetu – znamená, že veškerý aktuální obsah encyklopedie je v každém okamžiku přístupný jakémukoliv uživateli přes webové rozhraní a po autorech se nebude vyžadovat vytváření jakýchkoliv speciálních aplikací pro zobrazování obsahu encyklopedie

2.1.2 Principy Elektronické encyklopedie historie

Na základě výše uvedených obecných požadavků lze stanovit základní principy, na nichž bude návrh a implementace aplikace stát.

EE bude čistě webová aplikace, což umožní velmi snadnou přístupnost jak pro redaktory (je nutno počítat s tím, že historici patří k nejkonzervativnějším uživatelům výpočetní techniky), tak pro čtenáře.

EE bude disponovat redakčním systémem umožňujícím spolupráci více autorů na tvorbě obsahu, přičemž však bude zachováno autorství jednotlivých záznamů. Redaktor obecně nemá právo zasahovat do záznamů jiného redaktora.

Redakční systém bude co nejjednodušší, v největší možné míře bude vycházet ze zobrazovací části aplikace, aby práce s ním byla co nejintuitivnější.

A konečně hlavním principem EE je provázanost dat – jednotlivé záznamy budou v co nejvyšší míře provázány vazbami, přičemž redaktor může svůj záznam provázat s jakýmikoliv jinými existujícími záznamy, bez ohledu na to kdo je jejich autorem. Struktura obsahu encyklopedie bude striktně definována, díky čemuž bude tento strojově srozumitelný, což umožní efektivní využití dat v encyklopedii se nacházejících, jejich přehledné zobrazování, automatickou tvorbu přehledů a jejich analýzu.

Datový model EE bude založen na sémantické síti.

3 Sémantická síť

Sémantická síť se skládá z uzlů, které reprezentují koncepty (konkrétní objekty nebo abstraktní ideje), a hran, které reprezentují vztahy mezi nimi. Vznikla původně pro účely zpracování přirozeného jazyka. Je však použitelná obecněji jako způsob reprezentace znalostí.[1] Pro naše účely je sémantická síť mimořádně vhodným nástrojem, protože odpovídá fungování sémantické paměti člověka. Sémantická paměť je ta složka dlouhodobé paměti člověka, která „není vázána na prostor a čas […], obsahuje významy pojmů a termínů, fakta, znalosti, vědomosti, vztahy mezi nimi".[2] Je to tedy právě ten typ paměti, který je zapojen do práce s historikovým modelem reality. Sémantická síť tak teoreticky umožňuje reprezentaci tohoto modelu mimo mysl člověka.

3.1 Sémantická síť jako graf

V matematickém smyslu je sémantická síť grafem. Jeho přesné vlastnosti mohou záviset na konkrétní implementaci, pro potřeby Elektronické encyklopedie historie bude použit orientovaný ohodnocený multigraf, proto se v této části zaměřím na tento druh grafu.

Orientovaný graf G je dvojice:

$$G = (V,E),$$

kde V je konečná množina uzlů a $E \subseteq V^2$ je množina hran

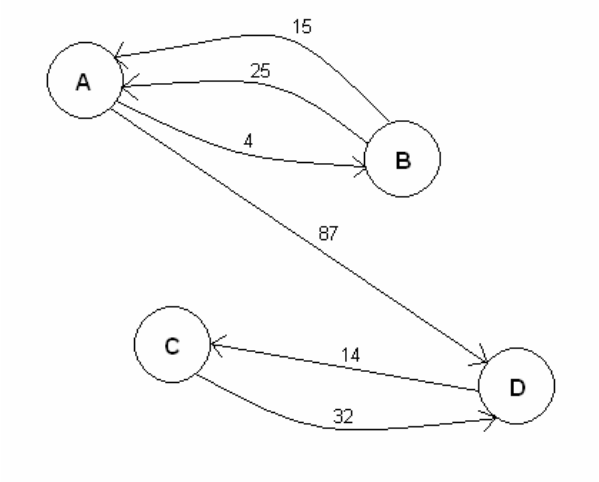
Skutečnost, že graf je orientovaný, znamená, že hrana (u,v) není totožná s hranou (v,u), nýbrž označuje se za invertovanou hranu (v,u). Hrana (u,v) v takovémto zápisu vede z uzlu *u* do uzlu *v*. Uzel *u* se nazývá přímým předchůdcem uzlu *v* a uzel *v* pak přímým následníkem uzlu *u*. Uzly *u* a *v* se v případě, že jsou propojeny hranou, nazývají sousedící uzly.

Ohodnocený graf je takový graf, jehož každá hrana má přiřazenu hodnotu určenou váhovou funkcí $w : E \to R$.[3]

Konečně multigraf je graf, v němž může existovat mezi dvěma uzly více než jedna hrana.*

V grafické reprezentaci takového orientovaného ohodnoceného multigrafu se uzly znázorňují zpravidla kružnicí (oválem), v němž může být název uzlu, hrany mezi nimi šipkou propojující oba uzly, směřující od předchůdce k následníku, ohodnocení hran se pak zapisuje k dané hraně. Příklad grafické reprezentace ohodnoceného orientovaného multigrafu představuje následující ilustrace.

* V odborné literatuře neexistuje shoda na terminologii týkající se multigrafů (někteří je nazývají též pseudografy), proto se zde zdržím formálních definic. Pro účely této práce stačí definovat multigraf jako graf, v němž je povoleno spojit dva uzly libovolným počtem hran, přičemž každá z nich může mít samozřejmě jiné ohodnocení.

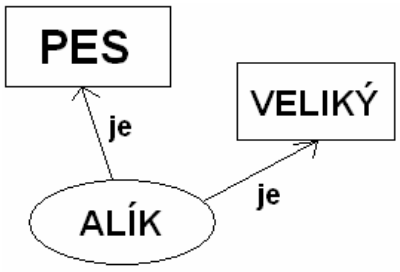


Obr. 3.1. – ilustrace ohodnoceného orientovaného multigrafu

3.2 Co to je sémantická síť

Sémantická síť je tedy orientovaný ohodnocený multigraf, v němž uzly představují objekty a hrany představují vztahy mezi nimi. Uzly v sémantické síti mohou představovat buď konkrétní objekty nebo třídy objektů, přičemž v jedné sémantické síti se mohou vyskytovat oba typy uzlů – v takovém případě je však nutné je rozlišovat (v grafické reprezentaci typicky tím, že třídy jsou označovány obdélníkem namísto kruhu/oválu).

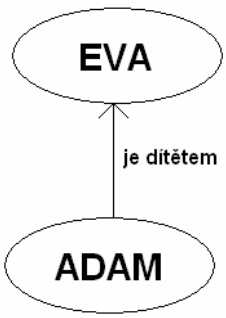
Ohodnocení hrany v sémantické síti vyjadřuje druh vztahu mezi dvěma uzly, které jsou hranou spojeny. Nevyjadřuje se proto zpravidla reálným číslem, jak by odpovídalo definici, ale textově (formálně by se to dalo vyjádřit jako funkce mapující váhy hran na textově vyjádřený význam hrany). Je však třeba mít na paměti, že při textovém ohodnocení hran přesná sémantika hrany závisí nejen na jejím ohodnocení, ale i na uzlech, které spojuje (což vychází z nejednoznačnosti přirozeného jazyka).



Obr. 3.2. – ilustrace závislosti sémantiky hrany na uzlech, které spojuje; v přirozeném jazyce jsou obě hrany ohodnoceny výrazem „je“, který má však v obou případech různý význam (v prvním případě zařazuje konkrétní objekt do určité třídy objektů, ve druhém přiřazuje objektu určitou vlastnost)

Dále je důležité si povšimnout, že přestože hrana mezi dvěma uzly je orientovaná, zavádí ve skutečnosti vztah mezi těmito uzly v obou směrech, nikoliv jen v tom, který odpovídá orientaci hrany. Oba vztahy jsou v takovém případě spojeny a nelze uvažovat jeden bez druhého, neboť vyjadřují sémanticky totéž (jen v přirozeném jazyce existují dva pojmy pro takovýto vztah, podle toho

z pohledu kterého z uzlů se na vztah díváme). Proto se do sémantické sítě explicitně zavádí vždy pouze jeden z nich, ale implicitně je tím vyjádřen i druhý.



Obr. 3.3. – ilustrace implicitního vztahu mezi dvěma uzly; explicitně je vyjádřen vztah „být dítětem“, implicitně je zde obsažen vztah (opačně orientovaný) „být rodičem“

Důležitým přínosem sémantické sítě jako způsobu reprezentace znalostí je možnost její pomocí odvozovat či odhalovat informace, které nejsou v síti přímo uvedeny. Implicitní inverzní vazby jsou asi tím nejjednodušším příkladem – nezdá se sice na první pohled, že by přinášely novou informaci, ale v praxi se stává, že některé vztahy mezi jistými objekty je zvykem uvádět pouze v jednom směru. Abych zvolil příklad z oboru, který je předmětem Elektronické encyklopedie historie – u českého geologa a biologa Jana Kašpara Palackého se i v nejstručnějším životopise člověk vždy dočte, že byl synem Františka Palackého. V životopisu Františka Palackého se však zmínka o synovi-geologovi nevyskytuje téměř nikdy[*]. Pokud je však taková informace reprezentována sémantickou sítí, jsou podobné nedůslednosti automaticky vyloučeny.

Druhým, a v praxi ještě přínosnějším, způsobem odvozování nových informací v sémantické síti je procházení sítě po hranách, neboli skládání (kompozice) hran. Jsou-li uzly *u* a *v* spojeny hranou *a*, a uzly *v* a *w* hranou *b*, vzniká automaticky složená hrana *ab* mezi uzly *u* a *w*. Je důležité si uvědomit, že informace o vztahu mezi uzly *u* a *w* (ať už vztah *ab* znamená cokoliv) se v sémantické síti explicitně vyjádřená vůbec nenachází.[4] Tento fakt nabývá na významu v případě rozsáhlých sítí, u nichž lze předpokládat, že ani člověk, který ji vytváří, si všech nepřímých vazeb nemůže být vědom, a obzvláště v případě, že síť není tvořena z informací z jednoho zdroje, nýbrž z několika. V takovém případě skládání hran umožňuje automatické skládání informací pocházející z různých zdrojů a tedy potenciální objevování skutečně nových souvislostí, které se v sémantické síti, ale ani ve zdrojích dat pro tuto síť, původně vůbec nevyskytovaly.

3.3 Použití sémantické sítě pro Elektronickou encyklopedii historie

V této části práce se budu zabývat konkrétním způsobem návrhu datového modelu pro Elektronickou encyklopedii historie jakožto sémantické sítě. Základním úkolem návrhu je stanovit, na jaké úrovni bude sémantická síť použita. Ačkoliv koncept sémantické sítě byl navrhnut původně pro zpracování přirozeného jazyka a většinou se proto předpokládá jeho použití na úrovni jednotlivých slov, maximálně pojmů, pro účely EE je potřebné použít sémantické sítě na úrovni o něco vyšší – na

[*] Viz životopisy těchto dvou mužů na Wikipedii

úrovni jednotlivých entit uložených v encyklopedii (kde se entitami rozumí osoby, události atd.). Uzly tedy v EE představují tyto entity a hrany vztahy mezi nimi (např. osoba způsobila událost). Každý uzel sémantické sítě je tudíž značně rozsáhlou a komplikovanou strukturu, neboť může obsahovat velké množství informací o dané entitě. Tyto informace budeme v EE nazývat atributy uzlu.

Hrany mezi těmito uzly budeme v Elektronické encyklopedii nazývat vazbami, a podobně jako uzly, i vazby mohou nést větší množství informace, než jen o typu vztahu. Proto i vazby mají své atributy.

Uzly sémantické sítě budou představovat konkrétní objekty, nikoliv třídy. Třídy objektů budou sice součástí návrhu, ale v sémantické síti nebudou vyjádřeny jako zvláštní uzly a vazby konkrétních objektů na tyto uzly, nýbrž jiným způsobem. Za tím účelem je nutno zavést analogii ohodnocení hran pro uzly – ohodnocení uzlů. Formálně se jedná o funkci, přiřazující každému uzlu právě jeden prvek z množiny typů.

Uzel v EE bude tedy definován svým typem a svými atributy. Vazba pak svým typem, svými atributy a uzly, které spojuje.

4 Návrh Elektronické encyklopedie historie

Návrh Elektronické encyklopedie historie se dá rozdělit na dvě části – jednou je samotný systém (webová aplikace) jako takový, a druhou je datový model pro reprezentaci obsahu encyklopedie. Návrh datového modelu je problém daleko vybočující z tématu této práce, budu mu proto věnovat jednu podkapitolu, v níž se pokusím nastínit aspoň základy tohoto modelu, ale ve zbytku kapitoly se budu zabývat pouze návrhem informačního systému EE. Ten je navržen tak, aby byl na použitém datovém modelu co nejméně závislý.

4.1 Neformální specifikace

Elektronická encyklopedie historie má být webový informační systém sloužící k ukládání a prezentování informací o historii. Za tímto účelem bude vytvořen datový model založený na sémantické síti umožňující zachytit informace, se kterými se v historii běžně pracuje, a jejich vztahy.

Základním charakteristickým znakem encyklopedie historie je nutnost pracovat s časem – každý objekt i vazba musí mít určení času existence (interval od-do). Přitom je potřeba umožnit časové údaje zadávat na různých úrovních od přesnosti na roky až po přesnost na hodiny a minuty. Zároveň je nutno umožnit zadávání neúplných, nepřesných i nejistých časových údajů. Součástí aplikace budou i nástroje na převod dat mezi různými kalendáři (gregoriánský, juliánský, muslimský…).

Ty objekty, u kterých to má smysl, by měly být ukotveny i geograficky. Aplikace však nemá imitovat GIS a k prostorovému ukotvení postačí umožnit zadávání GPS souřadnic, přičemž by bylo vhodné zajistit možnost zobrazení objektů opatřených GPS souřadnicemi v nějaké volně dostupné geografické aplikaci, nejlépe Google Earth.

Každý objekt v encyklopedii bude mít (krom časového a případně místního určení):

- název – identifikátor záznamu, pod kterým se bude v encyklopedii zobrazovat (nemusí ovšem být jedinečný)
- další užitečné atributy – v závislosti na návrhu datového modelu
- textové pole – prostor pro zadání jakýchkoliv informací o daném objektu, které nebudou zadány jiným způsobem
- redakční pole – prostor pro zapisování redakčních poznámek, což je jediná část záznamu, která nebude viditelná pro čtenáře
- vazby – propojení s ostatními objekty v encyklopedii, mající definovanou sémantiku a nesoucí informaci (především časovou, ale i jinou)
- zdroje – ke každému záznamu (objektu i vazbě) je nutno uvádět zdroje, z nichž použité informace pocházejí

Dále budou veškerým záznamům v encyklopedii (tedy objektům i vazbám) přiděleny filtry, které budou mít stromovou strukturu a budou vyjadřovat tématické okruhy. Strukturu filtrů bude mít právo vytvářet a měnit pouze redaktor s rozšířenými pravomocemi. Každý záznam může být přiřazen libovolnému počtu filtrů (přiřazení záznamu filtru může provádět každý redaktor, který má právo se záznamem manipulovat). Filtry rozdělují obsah encyklopedie na tematické celky (které se můžou libovolně překrývat).

Návštěvník encyklopedie si má možnost zvolit tematické okruhy a časové rozmezí, které ho zajímají, v závislosti na čemž mu bude nabídnut seznam vyhovujících objektů encyklopedie. Základní jednotkou encyklopedie je pak jednotlivý objekt, který obsahuje veškeré výše zmíněné informace včetně výpisu všech vazeb (i s informacemi náležejícími k vazbě, neboť vazby budou v encyklopedii obrazovány pouze v rámci objektů, které spojují, nikoliv samostatně). Přes vazby se pak uživatel dostane na související objekty, čehož může využít k procházení obsahu encyklopedie. V encyklopedii bude možno také vyhledávat, a to podle nadpisů objektů (názvů).

Aplikace bude také umožňovat ukládání dokumentů, tedy nahrávání souborů – textů i obrázků k objektům. Přitom by mělo být možno tyto dokumenty seskupovat, aby se mohl například několikastránkový dokument vložit do encyklopedie jako celek, ale zároveň umožňoval rozlišení jednotlivých stránek v odkazech na něj.

U vkládaných informací musí být zachováno autorství, to znamená že každý záznam (nejen objekt, ale i vazba) bude mít svého autora, který bude mít právo obsah záznamu měnit. Pouze redaktor s rozšířenými pravomocemi bude mít právo měnit obsah objektů, jejichž autorem sám není (v takovém případě je třeba uchovat informaci o tom, kdo do záznamu krom původního autora zasahoval). Protože provázanost objektů je základním principem EE, každý redaktor musí mít právo ke všem objektům, s nimiž má právo manipulovat, přidávat vazbu na jakýkoliv jiný objekt v encyklopedii, bez ohledu na jeho autora. Může také upravovat již existující vazbu mezi objektem, jehož je autorem, a jakýmkoliv jiným – vazba mezi dvěma uzly dvou různých autorů je jediný případ, kdy redaktor má právo zasahovat do objektu vytvořeného jiným redaktorem.

U všech záznamů v encyklopedii je také potřeba uchovávat informaci o datu poslední změny.

4.2 Analýza požadavků

Ze specifikace požadavků na aplikaci vyplývá, že systém bude navržen pro dva základní typy uživatele – čtenáře a redaktory, přičemž druhý z nich bude mít ještě speciální variantu s rozšířenými pravomocemi, kterou nazveme šéfredaktorem.

Čtenář je jakýkoliv návštěvník aplikace, aniž by se musel přihlašovat. Má právo prohlížet si kompletní obsah Elektronické encyklopedie historie s výjimkou redakčních poznámek, ale nemá právo do obsahu encyklopedie jakkoliv zasahovat.

Redaktor má právo v systému vytvářet nové objekty a upravovat objekty, jejichž je autorem. Nemá však právo zasahovat do objektů jiných redaktorů, s výjimkou přidávání vazeb mezi objektem svým a jakýmkoliv jiným.

Šéfredaktor má práva redaktora s tím, že navíc má možnost měnit záznamy všech ostatních redaktorů (tedy nejen ty, jejichž je on sám autorem) a manipulovat s filtry.

4.3 Datový model

Navrhnout datový model EE předpokládá znalosti z oboru historie a souvisejících oborů a jeho podoba není konečná (těžko lze předpokládat, že by kdy konečnou byla). V této práci se proto omezím pouze na jeho části dostatečně obecné a základní na to, aby se dalo s jistou důvěrou prohlásit, že tyto se nebudou měnit (nebo aspoň ne příliš často).

V encyklopedii existuje několik základních entit, které nazýváme sekce. Ačkoliv jejich počet ani složení nemusí být neměnné, uvedu zde pro pochopení významu tohoto pojmu jejich výčet: Událost, Osobnost, Organizace, Věc, Dokument, Místo. Je zřejmé, že každá z nich má úplně jiný

význam. Tyto entity jsou však příliš obecné a je užitečné kvůli větší přehlednosti obsahu encyklopedie rozdělit každou na několik kategorií (přičemž tyto nejsou výlučné a jeden uzel může patřit do více než jedné kategorie). Např. Organizace mohou být rozděleny na státní útvary, vojenské jednotky, politické strany, atd.

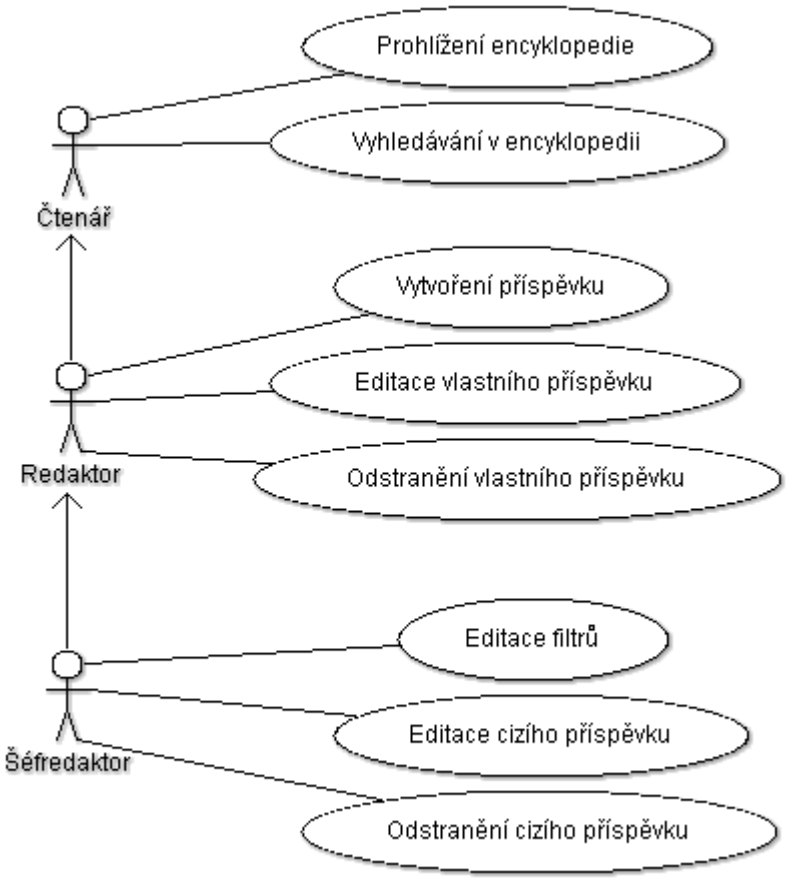
Zvláštní postavení mezi entitami mají Dokumenty – na počátku historického bádání stojí prameny, z nichž jsou informace o minulosti získávány, a tyto jsou v EE reprezentovány právě entitou Dokument. Dokumenty jsou jednou ze sekcí encyklopedie (stejně jako Události, Osobnosti atd.) a také se tak k nim přistupuje, navíc však umožňují vkládání souborů (obrazových či textových) a je možno je vázat na všechny zbylé uzly encyklopedie speciální vazbou, vyjadřující odkaz na zdroj informací v uzlu uvedených.

Mezi všemi entitami v encyklopedii pak existuje poměrně velký počet nejrůznějších vazeb, které jsou hlavním nástrojem přizpůsobování datového modelu novým potřebám, a proto se nejčastěji mění. Nebudu je zde proto podrobněji popisovat a omezím se na obecné vlastnosti – stejně jako uzly, i jednotlivé vazby jsou rozděleny do kategorií – např. vazba mezi organizacemi vyjadřující jejich závislost může mít kategorie „podřízená“, „součást“ apod. Na rozdíl od uzlů, kategorie vazeb jsou výlučné a každá vazba je právě jedné kategorie.

Veškeré uzly jsou pak přiřazeny libovolnému množství tematických okruhů. Jejich struktura je pro celou encyklopedii jednotná a rozděluje veškeré záznamy (bez ohledu na to, do které sekce patří) do množin uzlů, jejichž společným znakem je příslušnost k jednotlivým tematickým okruhům.

4.4 Případy užití

Diagram případů užití je poměrně jednoduchý. Zachycuje vztahy mezi třemi druhy uživatelů a každému přiřazuje několik málo případů užití. Případy „Editace vlastního příspěvku“ a „Editace cizího příspěvku“, resp. „Odstranění vlastního příspěvku“ a „Odstranění cizího příspěvku“ jsou totožné (liší se jen tím, jaký uživatel k nim má oprávnění) a v následující části, kde budou podrobně rozepsány, jsou zahrnuty pod jeden případ užití „Editace příspěvku“ (ID 4), resp. „Odstranění příspěvku“ (ID 5). Případ užití „Editace filtrů“ v sobě zahrnuje operace vytvoření nového filtru, změna stávajícího filtru (včetně změny jeho umístění ve stromové struktuře filtrů) a smazání filtru. Tento případ užití nepovažuji za nutné podrobně specifikovat, neboť filtry je na této úrovni abstrakce možno chápat jako dodatečné záznamy v Encyklopedii (k manipulaci s nimiž má přístup pouze šéfredaktor) a jednotlivé operace s nimi tak probíhají v principu stejně jako tytéž operace se záznamy (jsou tedy v podstatě totožné s případy užití „Vytvoření příspěvku“, ID 3, „Editace příspěvku“, ID 4 a „Odstranění příspěvku“, ID 5).



Obr. 4.1. – Diagram případů užití

4.4.1 Specifikace případů užití

Specifikace případu užití „Prohlížení encyklopedie“

ID	1
Název	Prohlížení encyklopedie
Popis	Zobrazí obsah encyklopedie podle požadavků čtenáře
Primární aktéři	Čtenář
Sekundární aktéři	---
Předpoklady	---
Následné podmínky	Je zobrazen žádaný záznam
Akce pro spuštění	Čtenář vstoupí do encyklopedie
Hlavní tok	1. systém zobrazí nabídku sekcí 2. čtenář zvolí sekci 3. systém zobrazí nabídku filtrování obsahu encyklopedie 4. uživatel zvolí tematické okruhy, kategorie a časové rozmezí 5. uživatel potvrdí výběr 6. systém zobrazí seznam vyhovujících záznamů encyklopedie 7. uživatel vybere žádaný záznam 8. systém zobrazí kartu záznamu
Alternativní tok	Pokud je uživatelem přihlášený redaktor-autor požadovaného

	příspěvku nebo šéfredaktor, může v bodu 7 zvolit u požadovaného záznamu „změnit“ a systém mu (bod 8) zobrazí namísto karty záznamu formulář pro změnu záznamu, kde uvidí obsah příspěvku v redaktorském modu (tedy včetně redakčních poznámek) a případně může obsah příspěvku editovat (viz případ užití „Editace příspěvku“, ID 4)
Výjimky	Storno
Frekvence	často

Specifikace případu užití „Vyhledávání v encyklopedii“

ID	2
Název	Vyhledávání v encyklopedii
Popis	Vyhledá záznamy
Primární aktéři	Čtenář
Sekundární aktéři	---
Předpoklady	
Následné podmínky	Je zobrazen seznam záznamů vyhovujících vyhledávacím kritériím
Akce pro spuštění	Čtenář zadá vyhledávací řetězec a zvolí „hledat“
Hlavní tok	1. systém zobrazí seznam záznamů odpovídajících kritériím 2. uživatel zvolí požadovaný záznam 3. systém zobrazí kartu záznamu
Alternativní tok	Pokud čtenář v okamžiku vyhledávání již zvolil sekci encyklopedie, zobrazí systém v bodu 1 pouze záznamy ze zvolené sekce, které odpovídají vyhledávacím kritériím
Alternativní tok 2	Pokud je uživatelem přihlášený redaktor-autor požadovaného záznamu nebo šéfredaktor, může v bodu 2 zvolit u požadovaného záznamu „změnit“ a systém mu (bod3) zobrazí namísto karty záznamu formulář pro změnu záznamu … (viz alternativní tok případu užití „Prohlížení encyklopedie“, ID 1)
Výjimky	Storno
Frekvence	střední

Specifikace případu užití „Vytvoření příspěvku“

ID	3
Název	Vytvoření příspěvku
Popis	Vytvoří nový uzel
Primární aktéři	Redaktor
Sekundární aktéři	---
Předpoklady	Redaktor je přihlášen do systému
Následné podmínky	V systému je uložen nový uzel
Akce pro spuštění	Redaktor zadá název nového příspěvku a zvolí „vytvořit“ na hlavní stránce sekce
Hlavní tok	1. systém uloží nový uzel ve vybrané sekci se zadaným názvem 2. systém uzlu přiřadí filtry, které měl redaktor v okamžiku vytvoření uzlu zvoleny 3. systém přejde do případu užití „Editace příspěvku“ pro nový

	příspěvek (ID 4)
Alternativní tok	
Výjimky	Storno
Frekvence	střední

Specifikace případu užití „Editace příspěvku“

ID	4
Název	Editace příspěvků
Popis	Změní obsah jednoho uzlu encyklopedie
Primární aktéři	Redaktor
Sekundární aktéři	---
Předpoklady	Redaktor je přihlášen v systému
Následné podmínky	Údaje u záznamu jsou změněny
Akce pro spuštění	Redaktor zvolí „změnit“ u záznamu, na jehož editaci má právo
Hlavní tok	1. systém zobrazí formulář na editaci uzlu 2. Pokud chce uživatel měnit atributy uzlu 2.1. uživatel změní libovolné atributy uzlu 2.2. uživatel potvrdí změněné údaje 2.3. systém uloží změněné údaje a informuje o tom uživatele 3. Dokud hodlá uživatel měnit vazby uzlu 3.1. uživatel vybere typ vazby, kterou chce změnit 3.2. systém zobrazí uživateli existující vazby vybraného typu 3.3. uživatel vybere vazbu, kterou chce změnit 3.4. systém zobrazí uživateli formulář na změnu vazby 3.5. uživatel změní údaje vazby 3.6. uživatel potvrdí změněné údaje 3.7. systém uloží změněné údaje a informuje o tom uživatele 4. Dokud hodlá uživatel přidávat vazby uzlu 4.1. uživatel vybere typ vazby, kterou chce přidat 4.2. systém zobrazí uživateli existující vazby vybraného typu 4.3. uživatel zvolí možnost přidat vazbu 4.4. systém nabídne uživateli hlavní stranu odpovídající sekce 4.5. uživatel zvolí tematické okruhy, kategorie a časové rozmezí 4.6. systém zobrazí seznam vyhovujících záznamů encyklopedie 4.7. uživatel u vybraného záznamu zvolí možnost přidat vazbu 4.8. systém uloží novou vazbu 4.9. systém zobrazí formulář na zadání informací k nové vazbě 4.10. uživatel zadá informace k vazbě a potvrdí 4.11. systém uloží změněné údaje a informuje o tom uživatele 4.12. uživatel se vrátí do formuláře na editaci uzlu 5. Pokud hodlá uživatel měnit filtry uzlu 5.1. uživatel zvolí, že chce měnit filtry 5.2. systém zobrazí strom filtrů a označí přiřazené filtry 5.3. uživatel změní označení libovolných filtrů 5.4. uživatel potvrdí změnu 5.5. systém uloží změněné údaje a informuje o tom uživatele
Alternativní tok	Pořadí bodů 2-5 je pouze orientační a znázorňuje rozmístění jednotlivých částí formuláře na obrazovce, nikoliv pořadí, v němž

	musí být prováděny. Uživatel může tyto body provádět v libovolném pořadí i opakovaně
Výjimky	Storno
Frekvence	často

Specifikace případu užití „Odstranění příspěvku“

ID	5
Název	Odstranění příspěvku
Popis	Smaže požadovaný uzel (včetně všech vazeb)
Primární aktéři	Redaktor
Sekundární aktéři	---
Předpoklady	Redaktor je přihlášen do systému
Následné podmínky	Daný uzel je odstraněn ze systému
Akce pro spuštění	Redaktor zvolí „smazat“ ve formuláři pro změnu záznamu
Hlavní tok	1. systém odstraní daný příspěvek, včetně všech vazeb mezi tímto příspěvkem a jakýmkoliv jiným 2. systém zobrazí hlavní stránku sekce, jejíž příspěvek byl odstraněn
Alternativní tok	
Výjimky	Storno
Frekvence	zřídka

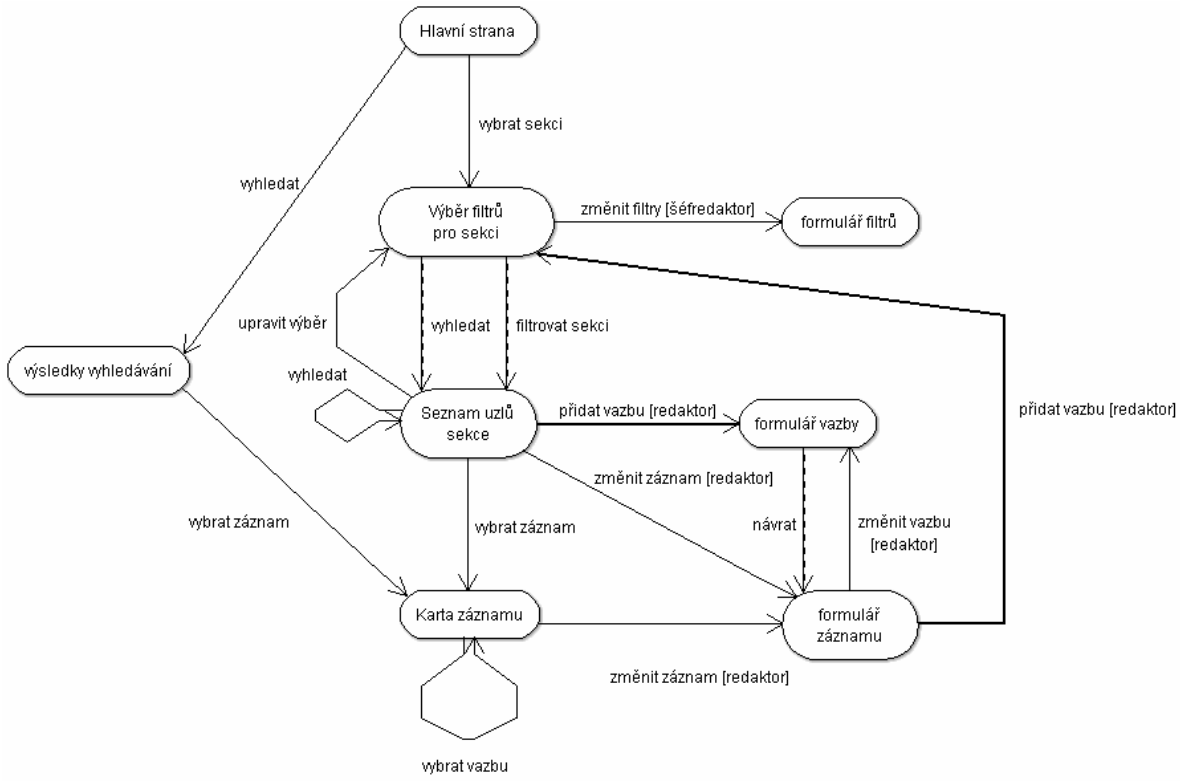
Obecná výjimka (ke všem případům užití) „Storno“

ID	E1
Název	Storno
Popis	Uživatel ukončí provádění případu užití před jeho dokončením
Primární aktéři	Uživatel
Sekundární aktéři	---
Předpoklady	---
Následné podmínky	Případ užití není proveden až do konce
Akce pro spuštění	Uživatel kliknutím na jakoukoliv možnost, kterou systém nabízí (např. položku menu) přejde do jiného případu užití
Tok	1. Systém přeruší provádění případu užití a přejde na začátek jiného (vybraného uživatelem). Veškeré dosud provedené změny v rámci původního případu užití zůstávají v platnosti.
Frekvence	zřídka

4.5 Diagram obrazovek

V diagramu návaznosti obrazovek není pro přehlednost znázorněna skutečnost, že z libovolné obrazovky se lze vrátit na hlavní stranu, nebo na hlavní stranu libovolné sekce (ve schématu nazvanou „výběr filtrů pro sekci“). Z tohoto důvodu u některých obrazovek chybí přechod na další obrazovku – zde se předpokládá, že uživatel využije některé z těchto dvou možností.

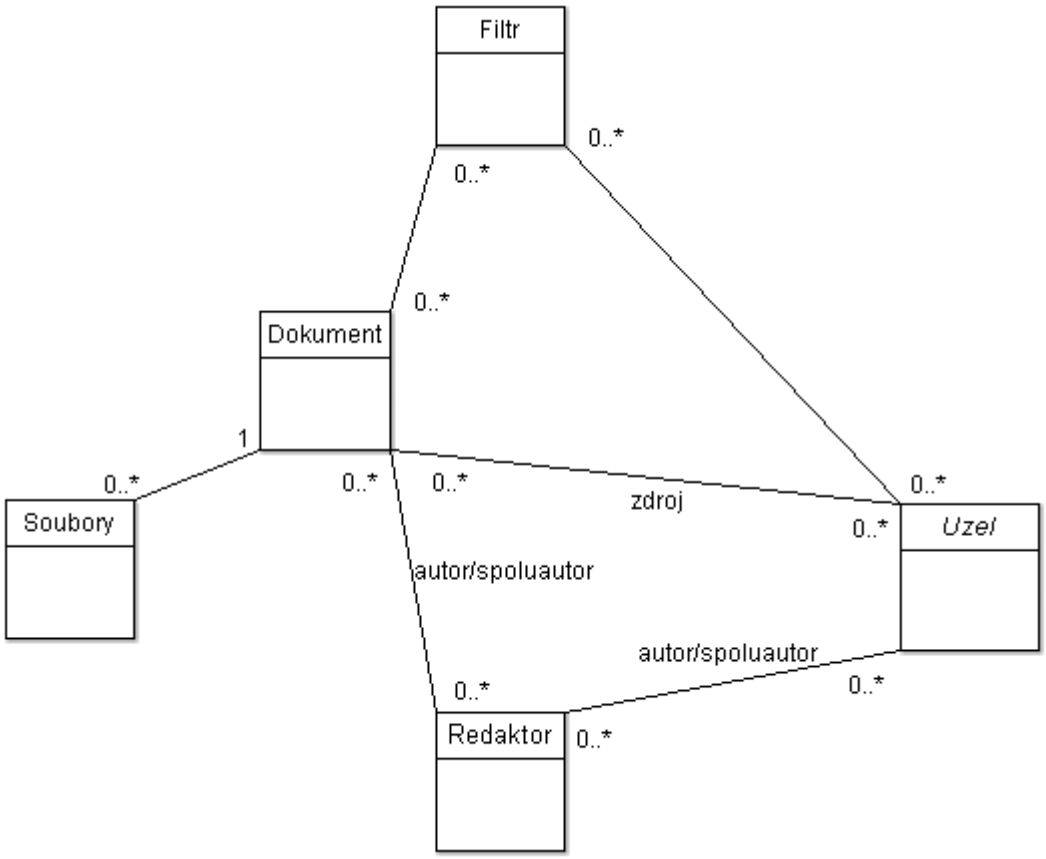
Pro větší přehlednost jsem také v diagramu obrazovek zvýraznil (ztučněním příslušných šipek) posloupnost obrazovek pro případ přidávání nové vazby mezi dvěma záznamy. Tato akce využívá pro výběr uzlu, k němuž má vazba vést, zobrazovací část aplikace. Redaktor, který se rozhodne přidat vazbu (nachází se tedy na obrazovce „formulář záznamu“) vybere typ vazby, kterou chce přidat, dostane se na hlavní stránku sekce (odpovídající typu uzlu, který je pro danou vazbu vyžadován), kde si podobně jako při běžném procházení encyklopedie vyfiltruje, případně vyhledá, uzly, z nichž požadovaný vybere pro přidání vazby; následuje formulář pro zadání údajů k vazbě a návrat do původního formuláře záznamu.



Obr. 4.2. – Schéma návaznosti obrazovek

4.6 Konceptuální schéma

Konceptuální schéma nezobrazuje veškeré entity, které v systému budou reprezentovány. Namísto všech entit odpovídajícím uzlům sémantické sítě, kterých bude několik v závislosti na datovém modelu, je v něm pouze jedna abstraktní entita „Uzel“, která zastupuje všechny ostatní. V kompletním schématu by mělo být takovýchto uzlů několik, všechny se stejnými relacemi mezi nimi a ostatními entitami znázorněnými ve schématu, a navíc s velkým množstvím různých relací mezi sebou navzájem. Všechny tyto vztahy jsou součástí datového modelu, cílem tohoto konceptuálního schématu je znázornit obecnou část návrhu entit, která je nezávislá na konkrétní podobě datového modelu.



Obr. 4.3. – Konceptuální schéma obecné části systému

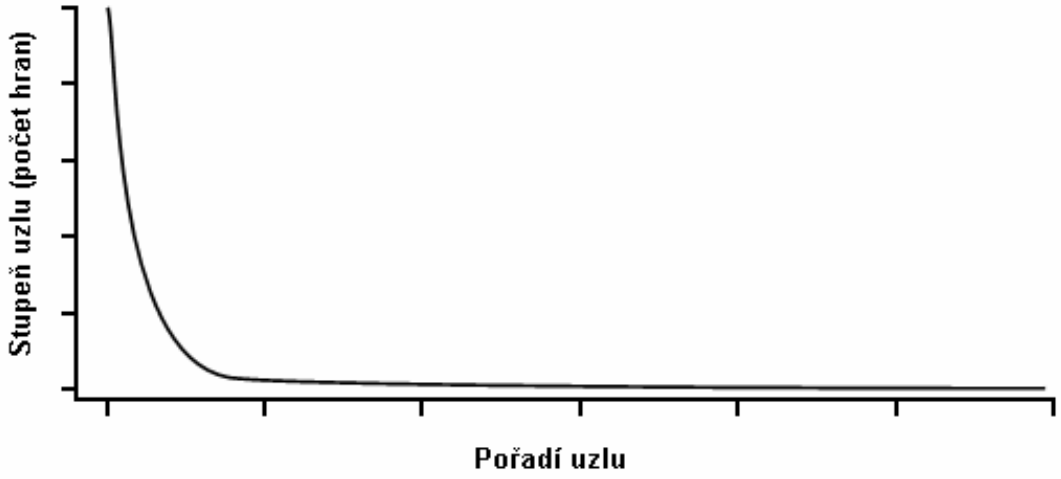
5 Analýzy v Elektronické encyklopedii historie

Jedním z přínosů Elektronické encyklopedie historie a zároveň hlavním tématem této práce je možnost analyzovat data v encyklopedii uložená. Analýza sémantické sítě odpovídá analýze grafu, proto i analýza obsahu Elektronické encyklopedie musí být založena na metodách analýzy grafů, resp. na metodách dolování v grafech.

Protože datový model Elektronické encyklopedie historie je založen na sémantické síti, splňuje charakteristiky toho, co se nazývá sociální síť (social network) – jde o heterogenní množinu dat reprezentovanou grafem, který je typicky velmi rozlehlý, s uzly odpovídajícími objektům a hranami odpovídajícími vazbám nebo interakcím mezi objekty; uzly i hrany mají atributy a objekty mohou být různých tříd[5]. I přes svůj název sociální sítě nemusejí nutně reprezentovat sítě vztahů mezi osobami, své uplatnění nalézají v nejrůznějších oborech od biologie přes sociologii až po technologii (např. inženýrské sítě).

Pro praktickou práci s takovouto sítí a pro možnosti analýzy dat v ní je důležitým poznatkem to, že sociální sítě zpravidla splňují charakteristiky tzv. sítě malých světů (small world network). To znamená[5][8]:

- Průměrný stupeň uzlu (tedy průměrný počet hran na jeden uzel) je ve srovnání s celkovým počtem uzlů velmi malý (pokud má například síť řádově tisíce uzlů, každý uzel je spojen s ostatními průměrně řádově jednotkami až desítkami hran). Průměrný stupeň uzlu tedy ani v jedné a téže síti není v průběhu času konstantní, jak by se dalo čekat u náhodně vytvářeného grafu, nýbrž s přibývajícími uzly zvolna stoupá. Jinými slovy hustota sítě se s přibývajícími uzly zvyšuje, závislost počtu hran v síti na počtu uzlů se dá vyjádřit tak, že počet hran závisí na n^a, kde n je počet uzlů a a nabývá hodnot mezi 1 a 2 (v případě že by bylo a = 1, znamenalo by to lineární závislost počtu hran na počtu uzlů, tedy konstantní průměrný stupeň uzlů v síti, hodnota a = 2 je opačným extrémem ukazujícím na mimořádně hustý graf, kde každý nový uzel je hranami spojen s konstantním počtem už existujících uzlů).
- Poloměr sítě je – i přes velký počet uzlů a malý průměrný stupeň uzlu – překvapivě malý. Poloměr sítě lze definovat buď jako největší v síti se nacházející, a nebo jako průměrnou, vzdálenost (tedy nejkratší možnou cestu) mezi libovolnými dvěma uzly. Jinými slovy jako počet uzlů, přes které je minimálně nutno projít, abychom se dostali z kteréhokoliv uzlu sítě do kteréhokoliv jiného. Poloměr sítě nestoupá ani zdaleka lineárně s počtem uzlů, jak by se dalo čekat u náhodně vytvářeného grafu, ale mnohem pomaleji. Uvádí se logaritmická závislost, ba dokonce poloměr někdy může s rostoucím počtem uzlů i klesat.
- Rozložení uzlů podle stupně je exponenciální. Seřadíme-li uzly sítě podle jejich stupně (počtu hran), dostáváme takovýto průběh závislosti stupně uzlu na jeho pořadí v tomto seřazeném seznamu uzlů:



Obr. 5.1. – Závislost stupně uzlů (počtu hran z nich vedoucích) na pořadí uzlu v seznamu uzlů seřazeném podle stupně

Jinými slovy, nejvíce uzlů má malý (podprůměrný) počet hran a uzlů s velkým (nadprůměrným) počtem hran je málo; funkce, která tuto závislost popisuje, klesá exponenciálně.

- Vysoká „shlukovitost“ – koeficient vyjadřující míru vytváření shluků v síti je definován jako poměr všech trojic sousedních uzlů, které jsou spojeny každý s každým, ku všem trojicím uzlů v síti. Jinými slovy jako pravděpodobnost, že vybereme-li náhodně ze sítě uzel a dva jeho sousedy, tyto dva sousední uzly budou také propojeny hranou. Tento koeficient (pravděpodobnost) je značně vyšší, než by se dalo čekat u náhodně vytvářeného grafu s tak velkým počtem uzlů s tak malým průměrným stupněm uzlů, jako sémantické sítě typu malý svět mívají. Znamená to, že uzly v síti typu malý svět mají tendenci vytvářet lokální shluky, tedy podsítě tvořené uzly, které jsou vzájemně hustě propojeny hranami[*].

Tyto vlastnosti, které bude sémantická síť, vzniklá po naplnění encyklopedie daty, s největší pravděpodobností splňovat, je potřeba zohlednit při návrhu a implementaci analýz dat v ní.

5.1 Analýzy grafů

Základním prostředkem analýzy grafu, který je použitelný pro naše účely, je hledání cest v grafu. Posloupnost $<v_0,v_1,v_2,\ldots,v_k>$, kde $(v_{i-1},v_i) \in E$ pro i=1,2,…,k se nazývá sled (délky *k*). Cesta je pak sled, v němž se neopakují uzly. Pokud existuje sled mezi dvěma uzly *v* a *v'*, je uzel *v'* dosažitelný z *v*. Vzdálenost těchto dvou uzlů se pak rovná délce nejkratšího sledu (resp. nejkratší cesty), který mezi nimi existuje.[3]

Při hledání cest v grafech je nutno využít algoritmů pro prohledávání grafů. Existují v zásadě dva takovéto algoritmy, a to BFS (breadth first search – prohledávání do šířky) a DFS (depth first search – prohledávání do hloubky).

[*] Zde je třeba upozornit, že shluky o kterých hovořím v tomto oddíle nemají nic společného se shlukováním jakožto analýzou navrhovanou na konci této kapitoly. Tam jde o shlukování uzlů na základě jejich struktury, zde o shluky uzlů na základě topologie sítě, resp. vzájemné provázanosti uzlů v síti. Tyto shluky souvisejí s tím, co navrhuji v rámci podkapitoly 5.1 jakožto Hledání souvisejících uzlů.

5.1.1 BFS

Algoritmus prohledávání do šířky postupně prochází všechny uzly náležející do souvislé komponenty*, do níž náleží výchozí uzel (kořen) prohledávání. V každém kroku prozkoumá všechny následníky aktuálního uzlu a ty zařadí do fronty (typu FIFO). Poté nahradí aktuální uzel prvním uzlem z této fronty a opakuje činnost, dokud se fronta nevyprázdní. Každý uzel, který už byl jednou prozkoumán, je označen a není opakovaně zpracováván.

Princip činnosti algoritmu je následující (*provedením operace* nazývám samotnou akci, kvůli které je graf procházen, např. výpis uzlu, porovnání uzlu s hledaným apod.):

1) vlož do fronty výchozí uzel (kořen)
2) vyber z fronty první uzel a *proveď operaci*
3) označ vybraný uzel za prozkoumaný
4) vlož všechny následníky vybraného uzlu neoznačené dosud za prozkoumané do fronty
5) pokud je fronta prázdná, ukonči činnost
6) jinak skoč na bod 2)

Algoritmus BFS je výpočetně úplný, to znamená, že projde každý uzel vstupního grafu. Přitom každý uzel v grafu bude zpracován právě jednou. Zpracování uzlu spočívá v hledání všech jeho přímých následníků, proto bude každá hrana v grafu existující prozkoumána právě dvakrát (jednou z každého z obou uzlů, jež spojuje). Časová složitost algoritmu je tedy O(|V| + |E|), tedy lineární vzhledem k počtu uzlů a hran v grafu. Prostorová složitost algoritmu je daná především použitím fronty pro ukládání uzlů, které mají být prozkoumány. Do té je nutno uložit všechny následníky všech uzlů z aktuální úrovně, tedy všechny uzly následující úrovně. Nutný prostor je tedy lineárně závislý na počtu uzlů v nejhlubší úrovni. To znamená v nejhorším případě prostorovou složitost O(|V|).

5.1.2 DFS

Algoritmus prohledávání do hloubky postupně prochází všechny uzly náležející do souvislé komponenty, do níž náleží výchozí uzel (kořen) prohledávání. Z aktuálního uzlu hledá prvního přímého následníka a z něj pokračuje stejným způsobem až do maximální hloubky a teprve když narazí na uzel, z něhož už nelze pokračovat (nevede z něj žádná hrana krom té, kterou do něj algoritmus přišel), vrací se o úroveň výš a pokračuje další hranou v pořadí. Každý uzel, který byl už jednou prozkoumán, je označen a není opakovaně zpracováván.

Princip činnosti algoritmu je následující:

1) spusť DFS(kořen)

DFS(uzel):

1) *proveď operaci*
2) označ *uzel* za prozkoumaný
3) pro všechny bezprostřední následníky uzlu *uzel* neoznačené dosud za prozkoumané spusť DFS(následník)

Algoritmus DFS je výpočetně úplný, to znamená že projde každý uzel vstupního grafu. Přitom každý uzel v grafu bude zpracován právě jednou. Zpracování uzlu spočívá v prozkoumání všech jeho

* Souvislá komponenta grafu je jeho podgraf, v němž mezi libovolnými dvěma uzly existuje cesta; formálněji řečeno je to třída ekvivalence množiny uzlů podle relace „je dosažitelný z“[3]

přímých následníků, proto bude každá hrana v grafu existující prozkoumána právě dvakrát (jednou z každého z obou uzlů, jež spojuje). Časová složitost je tedy O(|V| + |E|), tedy lineární vzhledem k počtu uzlů a hran. Prostorová složitost je dána v případě DFS nutností ukládat na zásobník potřebné informace při rekurzivním volání. To znamená v nejhorším případě prostorovou složitost O (|V|).

5.1.3 Návrh analýz pro Elektronickou encyklopedii historie

Na základě existence algoritmů pro vyhledávání cest v grafech pak lze navrhnout několik analytických úloh vhodných pro implementaci v rámci EE:

- hledání cest mezi zadanými uzly – uživatel zvolí dva uzly v encyklopedii a systém nalezne a zobrazí cesty mezi těmito uzly od nejkratší z nich po nejdelší (přičemž nejdelší zde znamená s délkou rovnou parametru, který vyjadřuje jaké nejdelší cesty nás ještě zajímají, nikoliv nejdelší v síti se nacházející)
- hledání dosažitelných uzlů – uživatel zvolí uzel v encyklopedii a systém nalezne a zobrazí všechny uzly, které jsou dosažitelné ze zadaného uzlu cestou délky 1, cestou délky 2 atd. (opět až po hranici zadanou jako parametr)
- hledání souvisejících uzlů – uživatel zvolí uzel v encyklopedii a systém nalezne a zobrazí určitý počet uzlů, které s vybraným uzlem v rámci sémantické sítě nejvíce souvisí (souvislost dvou uzlů lze určit pomocí tzv. Katzovy míry)

5.2 Dolování v grafech – shlukování

Dolování v grafech je součástí širšího oboru nazývaného Získávání znalostí. Řeší obdobné úlohy jako klasické dolování v databázích, ale za zdroj dat namísto standardní databáze považuje grafovou strukturu. Pro potřeby Elektronické encyklopedie historie jsem zvolil jako nejvhodnější z metod, které se v rámci dolování v grafech uvádějí, shlukovou analýzu.

Shlukování je proces rozdělování objektů do tříd (shluků) na základě své podobnosti. Podobnost objektů je definována na základě podobnosti hodnot jednotlivých atributů objektů, přičemž podobnost atributů se určuje pomocí vzdálenostní funkce. V grafickém vyjádření pak objekty představují body v n-rozměrném prostoru, kde n je počet atributů a souřadnice v každém rozměru je dána hodnotou příslušného atributu. Cílem shlukování je rozdělení objektů do tříd tak, aby podobnost objektů v rámci jedné třídy byla co nejvyšší a podobnost objektů z různých tříd co nejnižší.

Při aplikování tohoto postupu na graf a jeho strukturu budeme za objekty považovat uzly v grafu a za atributy budeme považovat počty hran jednotlivých typů, které z uzlu vedou. Každý uzel budeme proto reprezentovat uspořádanou n-ticí přirozených čísel (včetně nuly), kde n je celkový počet různých typů hran, které se v grafu nacházejí. Každé číslo v této n-tici je rovno počtu hran odpovídajícího typu, které z uzlu vedou. Tím jsme zavedli transformaci grafu na množinu objektů reprezentovaných atributy, na kterou můžeme aplikovat standardní algoritmy pro shlukování.

Abychom mohli shlukování provádět, musíme si zavést způsob, jak určit vzdálenost dvou objektů. Jelikož naše transformace uzlů grafu na objekty zavádí pouze celočíselné atributy objektů, jejichž vzdálenost budeme muset počítat, není potřeba provádět žádné další úpravy atributů a je možno použít některou ze standardních vzdálenostních funkcí – např. eukleidovskou či Manhattanovskou.

5.2.1 Metody shlukování

Existuje celá řada metod shlukování, které vycházejí z několika různých principů. Představíme si proto stručně tyto principy, jejich výhody a nevýhody[9].

Metody založené na rozdělování jsou nejjednoduššími metodami shlukování. Rozdělují množinu objektů do *k* shluků tak, že zvolí náhodně *k* center shluků a pak každý objekt přiřadí k nejbližšímu centru. Poté přepočítají centra shluků vzniklých v prvním kroku a provedou nové přiřazení všech objektů k nejbližšímu (novému) centru. Celý postup se opakuje, dokud se přepočítáváním center shluků mění přiřazení jednotlivých objektů ke shlukům. Nevýhodami těchto metod jsou to, že nachází jen shluky kulových tvarů, a že výsledek závisí na náhodně zvolených centrech shluků v prvním kroku – metoda obecně nenalézá řešení optimální, ale jen suboptimální. Navíc je nutno při spuštění zadat požadovaný počet shluků, což předpokládá, že už před provedením shlukování známe aspoň částečně jeho výsledek.

Metody hierarchické jsou založeny na jiném principu – provádějí hierarchický rozklad množiny objektů a vytvářejí strom shluků, kde na nejnižší úrovni je každý objekt samostatným shlukem a na nejvyšší úrovni jsou všechny objekty součástí jednoho shluku, přičemž každá úroveň mezi těmito dvěma extrémy představuje rozdělení do počtu shluků o jeden menšího, než na bezprostředně nižší úrovni. Výhodou těchto metod oproti předchozím je to, že nevyžadují zadání požadovaného počtu shluků, nýbrž nabízejí celou hierarchii výsledků, z níž je možno si vybrat. Nevýhodou je to, že při postupném budování hierarchie shluků (ať už shora dolů nebo zdola nahoru) nelze v dalším kroku už měnit jednou stanovené rozdělení objektů do shluků (přesunout objekt mezi shluky), i když by to bývalo vedlo k lepšímu výsledku. Navíc tyto metody jsou díky tomu, že vytvářejí celou hierarchii shluků, výpočetně náročnější než ostatní (v každé nové úrovni je nutno přepočítat vzdálenost nově vytvořeného shluku ke všem ostatním objektům/shlukům).

Metody založené na hustotě jsou nejpříměji navázány na výše uvedenou definici toho, co to vlastně shlukování je – objekty považují za body v n-rozměrném prostoru (kde n je počet atributů) a hledají oblasti v prostoru s velkou hustotou objektů, které považují za shluky, oddělené oblastmi s malou hustotou objektů. Konkrétním zástupcem tohoto druhu metod shlukování je metoda zvaná DENCLUE (Density-based Clustering neboli Shlukování založené na hustotě), která modeluje vliv každého objektu na své okolí tzv. funkcí vlivu, což je jakákoliv funkce odvozená od vzdálenosti objektů, v nejjednodušším případě např. funkce nabývající hodnoty 1 ve vzdálenosti menší než σ od objektu a hodnoty 0 všude jinde. Dále je zavedena funkce hustoty, definovaná nad každým bodem v prostoru dat jako součet všech funkcí vlivu v daném bodě. Centra shluků jsou pak místa, kde tato funkce dosahuje lokálního maxima, a každý objekt je přiřazen ke shluku na základě průběhu funkce hustoty a jejího stoupání k lokálnímu maximu. Metoda vyžaduje zadání jednoho parametru, a to σ, který určuje vzdálenost, do jaké působí funkce vlivu každého objektu, a tím jednoznačně, ale nepřímo, určuje počet shluků, které budou nalezeny.

5.2.2 Návrh využití metod dolování v grafech pro Elektronickou encyklopedii historie

Na základě existence algoritmů pro shlukování pak lze navrhnout několik analytických úloh vhodných pro implementaci v rámci EE:

- hledání podobných uzlů – uživatel zvolí uzel v encyklopedii a systém nalezne uzly, které mají stejnou či podobnou strukturu (vazby na ostatní uzly) jako vybraný uzel; tato úloha je ve

skutečnosti podúlohou shlukování, neboť je třeba vypočítat vzdálenost zvoleného uzlu od všech ostatních uzlů a hledat takové uzly, jejichž vzdálenost od zvoleného je nejmenší

- shlukování uzlů – systém rozdělí uzly v encyklopedii do shluků na základě jejich podobnosti za použití některého z algoritmů pro shlukování

6 Implementace

Aplikace je implementována jako webová ve skriptovacím jazyce PHP a využívá databázového systému MySQL. Pro dosažení dynamického chování je použito javascriptu a jeho knihovny jQuery a technologie ajax. Je umístěna na serveru valka.cz, kde je k dispozici prostředí PHP verze 5.2.9 a MySQL 5.1.31, které však teprve v průběhu vývoje této aplikace nahradilo MySQL verze 4.0, která byla kvůli rychlosti na serveru používána až do nedávné doby. Při vytváření struktury databáze a při vytváření dotazů do ní bylo tedy nutno respektovat její omezení (např. absence podpory složených SQL dotazů), a také omezení vyplývající ze současného nastavení prostředí (absence kontroly referenční integrity).

6.1 Implementace Elektronické encyklopedie historie

6.1.1 Uzly a vazby

Každá entita, která se v encyklopedii zobrazuje (tedy všechny uzly) musí mít určenu svou etiketu, (plnící úlohu člověku srozumitelného identifikátoru), pod nímž bude uživateli zobrazována. U všech uzlů je tento identifikátor obecně tvořen

1) názvem uzlu – název uzlu je jediným povinně vyplňovaným atributem každého uzlu, zadává se v češtině (pouze u uzlů typu Osobnost je „název“ tvořen ve skutečnosti několika atributy zvlášť pro jméno, příjmení, šlechtický přídomek atd.)

2) originálním názvem uzlu – většina uzlů má jako další atribut originální název, tedy oficiální název, který se uvádí, pokud je v jiném než českém jazyce; tvoří součást etikety, neboť umožňuje rozlišit například organizace se stejným českým názvem (typicky: 1.pluk)

3) datem existence – atribut udávající období existence má každý uzel a i tento je součástí etikety, neboť umožňuje jednak rozlišovat entity stejného názvu existující v různých obdobích (například osoby z jedné rodiny, často mající v různých generacích totožná i křestní jména) a jednak jde o základní informaci pro každý uzel, jejíž zobrazení i mimo kartu uzlu zvyšuje přehlednost pro uživatele

Výjimky představují v tomto směru uzly některých typů (konkrétně Organizace a Místo), u nichž je nutno mít možnost zadávat více názvů v různých časových obdobích (typicky přejmenovávání měst – u nás např. Gottwaldov/Zlín). Nutit uživatele, aby v takovém případě vytvářel právě kvůli identifikaci v encyklopedii pro danou entitu více záznamů, byť by je měl možnost odpovídajícími vazbami provázat do vztahu „předchůdce-následník“, se ukázalo jako velmi nepraktické a navíc teoreticky chybné – bez ohledu na měnící se název jde stále o tutéž entitu, která má své atributy a vazby, které se vztahují k ní jako celku. U těchto uzlů jsou proto součástí etikety všechny názvy, které během své existence nesl, a zobrazují se vždy ty názvy, které odpovídají časovému období, které si uživatel zvolil (u zobrazování cílů vazeb pod názvem, který je aktuální v době, kdy existovala daná vazba). Důsledkem je, že tentýž uzel se může v rámci encyklopedie na různých místech vyskytovat pod různým názvem (etiketou), což přesně odpovídá tomu, že entita, kterou uzel reprezentuje, v různých dobách existovala pod různými názvy.

Hlavním vstupním bodem do encyklopedie je úvodní strana, kde je z menu možno vybrat sekci (Události, Osobnosti apod.) a poté pomocí filtrů, kategorií a časového určení (viz oddíl 6.1.4.) vyfiltrovat seznam uzlů, které uživatele zajímají. V uzlech v encyklopedii je také možno vyhledávat, a to dvěma způsoby – buď pouze mezi uzly vybraného typu (tedy ve vybrané sekci), a nebo mezi všemi uzly v encyklopedii (v takovém případě jsou zobrazeny výsledky vyhledávání ve všech sekcích, avšak každé sekce zvlášť). Vyhledávání probíhá v názvech a originálních názvech uzlů. Vyhledávání probíhá tak, že vyhledávaný výraz je rozdělen na základě mezer na jednotlivá slova a vyhledávají se takové uzly, v jejichž atributech název a originální název se vyskytují všechna slova z vyhledávacího výrazu (bez ohledu na to, která z nich se vyskytují v názvu a která v originálním názvu).

Vazby mezi uzly jsou zásadní součástí Elektronické encyklopedie, neboť jsou podstatou sémantické sítě. Jelikož se v této práci vyhýbám konkrétnímu popisu datového modelu Elektronické encyklopedie historie, zmíním jen jednu základní vlastnost vazeb – každá vazba v sémantické síti nese definovanou sémantiku, která závisí na tom, mezi jakými typy uzlu vazba vede. Mezi týmiž typy uzlu však může existovat několik sémanticky rozdílných vazeb, a proto jsou zavedeny kategorie vazeb, které je rozlišují. Ty pak umožňují v kartě uzlu zobrazovat vazby na ostatní uzly roztříděné podle jejich sémantického významu. Veškeré vazby jsou v rámci Encyklopedie prezentovány etiketou uzlu, na něhož vazba míří, který je samozřejmě hypertextovým odkazem na kartu onoho uzlu.

6.1.2 Časové údaje

Časové údaje jsou pro aplikaci typu Elektronické encyklopedie historie zásadní. K jejich uložení do databáze MySQL není možno použít speciálních datových typů pro ukládání dat, které MySQL nabízí, neboť rozsah hodnot těchto typů je pro EE naprosto nedostatečný (typ DATE má spodní hranici 1.1.1000). Proto nezbývá, než datum ukládat pomocí klasických celočíselných typů a pro práci s těmito údaji si vytvořit několik speciálních funkcí. Krom samotné doby je u každého data nutno udržovat i informaci o tom, v jakém kalendáři má být datum zobrazováno.

Každý záznam v encyklopedii (uzel i vazba) musí nést časové určení ve formě intervalu, tedy dvou dat udávajících dobu existence (od – do). Každé z těchto dat může nabývat přesné hodnoty, ale také nejisté či neznámé. Proto je nutno navrhnout takový způsob ukládání a reprezentace dat, který to umožní. Přesné hodnoty pochopitelně nepředstavují problém. Pro neznámou hodnotu je nutno vyčlenit jednu z možných hodnot. Vzhledem k tomu, že v kalendářích běžně neexistuje rok 0[*], nabízí se tato hodnota pro reprezentaci neznámé hodnoty. Největším problémem však je reprezentace nejistých hodnot – zde by, teoreticky, bylo možno vytvářet systém, který by umožnil zadávat rozmezí hodnot (znamenající „víme pouze, že inkriminované datum se nachází v intervalu od X do Y“), seznam hodnot („inkriminované datum je jedno z následujících: …) a v extrémním případě i odstupňování pravděpodobnosti („je možné, že inkriminované datum je buď X, a nebo Y, přičemž X je pravděpodobnější než Y“). Takovýto systém bych však považoval i pro uživatele za příliš komplikovaný a ve svém důsledku kontraproduktivní. Proto jsem navrhnul zjednodušenou verzi, která však zachová hlavní rysy a potřeby.

Datum má u každého záznamu tři funkce – jednak samozřejmě informovat o době existence, nicméně pro tyto účely stačí některé dodatečné informace (jako jsou právě alternativní data) zaznamenat do textového pole a není nutno je za každou cenu zadávat přímo do atributu určeného pro datum. Dále datum tvoří součást etikety – a jako takové by nebylo vhodné, aby bylo příliš

[*] Prvním rokem gregoriánského kalendáře je podle definice rok 1. Po roce 1 př.n.l. tak následuje rok 1 n.l.

komplikované a dlouhé, neboť etiketa se zobrazuje na mnoha místech v encyklopedii a její komplikovaný tvar by snižoval přehlednost karet záznamu. A konečně datum plní důležitou funkci tím, že umožňuje řazení seznamu uzlů, ať už v seznamu všech uzlů nějaké sekce, nebo v seznamu vazeb určitého typu v rámci karty uzlu – a aby mohlo tuto úlohu plnit, musí být možno jeho jakkoliv komplikovaný tvaru zjednodušit na prosté jednoznačné datum (byť nemusí být v takovém případě jisté, že je správné), které jednoznačně určí pozici záznamu v seřazeném výpisu uzlů.

Na základě těchto požadavků jsem proto zvolil následující kombinovanou variantu zápisu dat: Každé datum (tedy obě data udávající začátek a konec intervalu doby existence) je tvořeno jedním datem (které může být buď zadáno přesně či neznámé). Zadávání nejistých nebo nepřesných dat je pak umožněno jednak tím, že lze zadávat jen část celého data (pouze rok, nebo pouze měsíc a rok), a jednak přidáním příznaků ke každému z dat, které umožňují modifikovat přesný význam zadaného data tím, že ho označí buď za horní („před“) či spodní („po“) hranici možného data a nebo za přibližné datum („asi“). Jakékoliv další upřesnění či podrobnosti pak lze zadávat přímo do textového pole záznamu.

U těch uzlů, u kterých to má smysl, je možno k datům zadávat i časy (v případě zadávání časů v lokálním časovém pásmu i včetně určení onoho časového pásma). Pro zobrazení takto komplikovaně reprezentovaného data jsem pak implementoval funkci, která provádí formátování data předávaného jako vstupní parametry podle toho, co obsahuje (s časem/bez času, úplné datum/jen rok, známé/neznámé datum atd.), do odpovídajícího tvaru. Tato funkce se pak používá při zobrazování jakéhokoliv data kdekoliv v encyklopedii.

Aplikace obsahuje také nástroj umožňující převádět data mezi nejčastěji používanými kalendáři (gregoriánský, juliánský, muslimský, hebrejský s tím, že je možno v budoucnu podle potřeby doplnit další). Tato funkcionalita je implementována na základě volně dostupných algoritmů[*] a slouží jednak k tomu, aby umožnila převedení jakéhokoliv data v Encyklopedii se nacházejícího do libovolného kalendáře, hlavně však k tomu, aby bylo možno veškerá data do Encyklopedie zadávat v jiném než gregoriánském kalendáři, pokud jsou takto uvedeny v pramenech[**] (aby tedy odpadla nutnost převádět je ručně). Aby však bylo možno data v encyklopedii používat ke všem výše uvedeným účelům (především tedy k chronologickému řazení uzlů a také k analýzám, popsaným dále), musí být v databázi ukládána všechna v tomtéž kalendáři (resp. kdyby byla ukládána v různých kalendářích, musela by být při každém použití převáděna). Nabízí se samozřejmě použít rovnou kalendář gregoriánský, v němž lze očekávat nejvíce zadaných dat. Systém tedy zajišťuje při uložení jakéhokoliv data jeho převedení do gregoriánského kalendáře s tím, že při zobrazování jakéhokoliv data je naopak převedeno z gregoriánského kalendáře do kalendáře, v němž bylo zadáno (což je určeno příznakem u každého data).

Krom toho je součástí Elektronické encyklopedie i nástroj pro výpočet dne v týdnu pro dané datum (založený na výše uvedených algoritmech), který umožňuje ve vybraných případech zobrazit k datům odpovídající den.

6.1.3 Redakční systém

Redakční systém je interní část aplikace, k níž mají přístup pouze registrovaní redaktoři. Jejich identifikaci zajišťuje přihlašovací systém. Ten využívá cookies a funguje tak, že redaktor se přihlásí

[*] Konkrétně z http://www.fourmilab.ch/documents/calendar/

[**] Například naprostá většina dat ze středověku je udávána v historických publikacích běžně v juliánském kalendáři, který se přestal používat v Evropě až v novověku, v Rusku dokonce až v roce 1918

svým uživatelským jménem a heslem, a pokud takovýto uživatel v databázi uživatelů existuje (přihlášení proběhne úspěšně), vygeneruje se náhodná hodnota jakožto dočasný identifikátor uživatele (přičemž se kontroluje, aby se dvěma současně přihlášeným uživatelům náhodou nevygenerovaly stejné hodnoty), která je uložena jednak do databáze uživatelů a jednak do cookies. Každá stránka patřící do redakční (interní) části aplikace pak kontroluje, zda na počítači, ze kterého přichází požadavek na její zobrazení, existuje cookie s identifikátorem, který se nachází v databázi, a pokud ano, rozpozná uživatele jako přihlášeného redaktora, umožní mu stránku zobrazit a zároveň získá jeho identifikaci, která je používána při zaznamenávání změn v encyklopedii. Platnost cookie je časově omezená, čímž se dosahuje vypršení platnosti přihlášení po jisté době nečinnosti.[*]

Ke každému uzlu i vazbě je zaznamenávána informace o autorovi záznamu. Tato informace slouží k určení, zda daný redaktor má či nemá právo zasahovat do záznamu, a u uzlů je zobrazována i na kartě záznamu. U uzlů je navíc udržována informace i o všech ostatních redaktorech, kteří do záznamu zasáhli, která slouží k informaci pro autora záznamu o tom, který z ostatních redaktorů (resp. šéfredaktorů) do jeho záznamu zasahoval.

Redakční systém slouží k editaci a vytváření záznamů v encyklopedii. Uzly jsou vytvářeny ve dvou krocích – nejdřív je nutno vybrat typ uzlu, který chce redaktor vytvořit, a zadat jeho název (jediný povinný atribut každého uzlu). Po potvrzení dojde k vytvoření tohoto uzlu v databázi a je zobrazen standardní formulář, používaný pro editaci jakéhokoliv uzlu, který umožňuje doplnit veškeré další atributy. K možnosti vytvořit uzel se redaktor dostává na hlavní stránce vybrané sekce. To umožňuje jakémukoliv nově vytvořenému uzlu automaticky přidat filtry (viz následující oddíl 6.1.4.), které má redaktor právě zaškrtnuty (cílem je zefektivnit práci redaktora, u něhož se předpokládá, že si nejdříve zobrazí seznam uzlů dané sekce, aby zjistil, zda uzel, který potřebuje vytvořit, již existuje, a pokud ho nenalezne, tak ho vytvoří; proto je praktické, aby uzlu při vytvoření byly rovnou přiřazeny zvolené filtry).

Krom atributů má každý uzel množství vazeb. Tyto je možno zadávat v redakčním systému od libovolného z obou uzlů, které vazba spojuje. Praktické řešení vypadá tak, že u každého uzlu jsou krom jeho atributů vypsány i všechny druhy vazeb, které se k němu váží. Pomocí technologie ajax pak lze každý druh vazby rozbalit a zobrazit seznam všech konkrétních vazeb, které jsou u aktuálního uzlu zadány (resp. seznam všech uzlů, na něž vede vazba daného typu z aktuálního uzlu). Zde lze také údaje patřící ke každé vazbě měnit (s výjimkou cílového uzlu, ten nelze změnit, vazbu lze smazat a vytvořit novou). Z tohoto místa lze také vytvářet novou vazbu, přičemž ta vznikne po kliknutí na „vytvořit novou vazbu“ vybráním cílového uzlu ze zobrazovací (veřejně přístupné) části aplikace, načež ji lze měnit stejně jako existující vazby. Pro zefektivnění práce je redaktor po vytvoření vazby ihned přesměrován do formuláře pro změnu právě vzniklé vazby, aby nemusel znovu načítat seznam vazeb (což je jinak nutno udělat pro zobrazení seznamu všech vazeb včetně nově zadané, neboť protokol HTTP neumožňuje vynutit znovunačtení stránky ze strany serveru). Navíc při vytváření nové vazby je možno, v případě že dosud v encyklopedii neexistuje, rovnou vytvořit i uzel, na něhož má vést. V takovém případě systém umožní redaktorovi namísto výběru uzlu, který má tvořit druhý konec vazby, vytvořit uzel nový, přičemž zároveň je do databáze uložena i vazba mezi aktuálním a tímto nově vytvořeným uzlem, kterou redaktor zadává.

[*] Tento systém byl zvolen namísto standardního řešení pomocí sezení (session), neboť na rozdíl od něj umožňuje kontrolu doby, po níž bude uživatel v případě nečinnosti automaticky odhlášen. V praxi se totiž ukázalo být užitečným prodloužit tuto dobu řádově na několik hodin, neboť práce s encyklopedií vede někdy k delším prolukám v aktivitě uživatele. Pokud by se však v budoucnu měla ukázat nedostatečná bezpečnost tohoto způsobu autentizace, bylo by nutno se této výhody vzdát a přejít na jiný způsob přihlašování redaktorů.

6.1.4 Filtry

Filtry slouží k tématickému členění obsahu encyklopedie, vytvářejí jakési řezy v sémantické síti. Každý uzel může být přiřazen do libovolného počtu tématických okruhů, tedy filtrů, a proto je nutno informaci o příslušnosti uzlu k filtru udržovat způsobem, který to umožňuje. Konkrétně u každého uzlu je v databázi uložen v jednom ze sloupců příslušné tabulky seznam příslušných filtrů oddělených čárkou, což umožňuje pracovat s nimi pomocí funkce MySQL *find_in_set()*. S filtry je možno pracovat třemi různými způsoby, které se liší ve způsobu zobrazování seznamu uzlů při zvolení filtrů, přičemž v encyklopedii se – v závislosti na typu uzlu – používají všechny tři. Společné mají to, že při zvolení některých filtrů uživatelem jsou zobrazeny všechny uzly, které mají v množině svých filtrů aspoň jeden ze zvolených.

Nejjednodušší způsob zpracování seznamu uzlů je použit u těch typů uzlů, u nichž je seznam uzlů zobrazován jako prostý seznam (nikoliv jako stromová struktura). Zde se seznam zobrazuje tak jak bylo popsáno výše – zobrazí se prostě všechny uzly, mezi jejichž filtry je aspoň jeden ze zvolených.

O něco složitější způsob zpracování seznamu je třeba použít u těch typů uzlů, kde je potřeba seznam uzlů zobrazovat jako stromovou hierarchii. Ta je definována vztahem typu nadřízený-podřízený, přičemž je ovšem zajištěno, že každý uzel má maximálně jeden nadřízený uzel a celá hierarchie je statická. Zde je potřeba definovat, které uzly se mají zobrazit, má-li některý z podřízených uzlů uzlu X mezi svými filtry jeden ze zvolených, kdežto ostatní podřízené uzly uzlu X nikoliv a samotný uzel X také ne. Hierarchie se v tomto případě nesmí rozpadnout, nesmí se tedy objevit například některé podřízené uzly uzlu X, aniž by byl zobrazen uzel X. Je tedy nutno, aby se v takovém případě zobrazil uzel X (přestože sám zvolený filtr přiřazen nemá) a ty jeho podřízené uzly, které zvolený filtr přiřazen mají (ne tedy všechny). Toho je možno docílit pomocí šíření filtrů – každý uzel takovéhoto typu (jejichž seznam je třeba zobrazovat jako stromovou hierarchii) tedy má uloženo nejen to, jaké filtry má zadány on sám, ale i to, jaké filtry mají zadány všechny jeho podřízené uzly (což se aplikuje rekurzivně). Prakticky je to realizováno uložením seznamu filtrů, které mají nastaveny podřízené uzly, do databáze stejným způsobem, jakým jsou uloženy vlastní filtry. Tyto údaje je nutno při každé změně ať už v zadání filtrů některým uzlům nebo při změně hierarchie uzlů aktualizovat, aby byly neustále v konzistentním stavu (redundance uložených informací je daní za snížení zátěže databáze při vypisování seznamu uzlů, které by si jinak vyžadovalo větší počet dotazů do databáze a hlavně složitější zpracovávání jejich výsledků v samotném php kódu stránky). To si vyžaduje udržování dodatečné informace o tom, kolika různým podřízeným uzlům je každý z filtrů přiřazen, která umožňuje správně rozhodovat, zda při odebrání filtru některému z podřízených uzlů je možno jej odebrat také jeho nadřízenému uzlu, a nebo zda má ještě jiný z uzlů podřízených tomuto uzlu inkriminovaný filtr zadán a v seznamu filtrů nadřízeného uzlu tedy musí zůstat.

Vůbec nejsložitějším případem pak jsou ty typy uzlů, které tvoří hierarchii dynamickou – to znamená, jejichž vztahy nadřízený-podřízený (které hierarchii definují) se mění v čase (tedy v závislosti na období, které uživatele zajímá) a každý uzel tak může mít libovolný počet nadřízených uzlů. Takovouto hierarchii lze sestavit teprve v okamžiku, kdy uživatel zadá rozsah období, které ho zajímá, a proto nelze výše popsanou metodu zaznamenávání všech filtrů podřízených uzlů k jejich nadřízenému uzlu použít. Zde se nabízejí dvě možnosti – jedna je sestavovat hierarchii skutečně dynamicky až poté, kdy si uživatel zvolí období, které ho zajímá, což s sebou nese výše zmíněné nároky na počet dotazů do databáze a složitost zpracování výsledků kódu stránky, a druhá je rezignovat v tomto případě na sestavování kompletní hierarchie a zobrazovat skutečně jen ty její části,

které odpovídají zvoleným filtrům. Zvolil jsem variantu druhou*, která znamená, že v případě, že některé podřízené uzly uzlu X mají zadán zvolený filtr a jiné nikoliv a sám uzel X také ne, zobrazí se v seznamu uzlů ty podřízené uzly, které zvolený filtr zadaný mají, aniž by se zobrazoval uzel X; hierarchie se tedy v tomto případě nezobrazí kompletně, uzly podřízené uzlu X se zobrazí jakoby byly na nejvyšší úrovni hierarchie.

Při zásahu do stromu filtrů je nutno zohlednit změnu u všech uzlů, které daný filtr mají přiřazen – konkrétně pokud je filtr smazán, je u všech uzlů nahrazen filtrem, který je ve stromu filtrů rodičovským uzlem mazaného filtru.

Krom filtrů lze uzly přiřazovat i jednotlivým kategoriím. Kategorie uzlů jsou obdobou kategorií vazeb – každý typ uzlu má vlastní strom kategorií, neexistují tedy (na rozdíl od filtrů) napříč celým obsahem encyklopedie. Jejich cílem je podrobnější rozdělení uzlů daného typu do tříd (například uzel typu Osobnost může mít kategorie Voják, Politik apod.). Na rozdíl od kategorií vazeb, kategorie uzlů nejsou disjunktní a jeden uzel může patřit do více než jedné kategorie. Krom toho strom kategorie uzlů, stejně jako filtrů, smí vytvářet a editovat šéfredaktor, kdežto kategorie vazeb jsou součástí datového modelu Encyklopedie, který žádný redaktor měnit nemůže. Je to dáno tím, že zatímco kategorie vazeb upřesňují či přímo určují sémantiku dané vazby, kategorie uzlů mají význam více podobný filtrům – zpřehledňují obsah encyklopedie a třídí uzly. Není však vyloučeno, že v případná realizace některých specifických nástrojů na analýzu obsahu encyklopedie (které by byly úžeji svázány s jejím datovým modelem, a které nejsou součástí této práce) si v budoucnu vynutí změnu přístupu ke kategoriím uzlů, která je učiní naopak bližšími kategoriím vazeb.

Nakonec je nutno ještě definovat způsob zobrazování uzlů na hlavní stránce sekce, respektive jak se do tohoto zobrazování promítne uživatelem provedený výběr filtrů a kategorií. V případě, že uživatel zvolí několik filtrů, zobrazí se seznam všech uzlů, které mají přiřazen aspoň jeden z těchto filtrů. To samé platí pro kategorie. Pokud uživatel vybere jak filtry, tak kategorie, zobrazí se uzly, které mají přiřazen aspoň jeden z vybraných filtrů a zároveň aspoň jednu z vybraných kategorií. V jazyce logických operátorů to zle vyjádřit jako zobrazení všech uzlů, které mají přiřazen ($filtr_1$ OR $filtr_2$ OR … OR $filtr_n$) AND ($kategorie_1$ OR $kategorie_2$ OR … OR $kategorie_n$), kde $filtr_1$ až $filtr_n$, resp. $kategorie_1$ až $kategorie_n$ jsou uživatelem vybrané filtry, resp. kategorie.

6.1.5 GIS

Propojení Elektronické encyklopedie historie s GIS systémem je řešeno s využitím programu Google Earth (šířeného jako freeware) a toho, že tento program umožňuje poměrně snadné a variabilní přidávání uživatelských dat pomocí jazyka KML.

KML (Keyhole Markup Language) je značkovací jazyk podle standardu XML, vyvinutý firmou Keyhole, která byla průkopníkem ve zpracování a prezentaci geografických dat. V roce 2004 ji koupil Google a z její aplikace EarthViewer se stal Google Earth. Vývoj Google Earth i KML od té doby pokračuje pod hlavičkou firmy Google. Pomocí KML je možné reprezentovat grafické objekty, které lze vkreslit do Google Earth, na podklad tam používaných map (satelitních snímků Země). Jelikož je to otevřený formát založený na XML, lze tyto soubory ručně editovat, případně automaticky generovat ze zadaných údajů. Pomocí KML lze reprezentovat bod umístěný na zadaných souřadnicích, ale i čáry a plochy nebo třeba poloprůhledné obrázky přiložené na zadané místo zeměkoule a 3D modely objektů umístěné tamtéž. Důležité je, že KML je podporován nejrůznějšími GIS programy, takže není závislý konkrétně na Google Earth a je možno v budoucnu v případě

* Důvody souvisejí s typy uzlů, u nichž je dynamická hierarchie zobrazována, tedy s datovým modelem a zvolenou metodologií vkládání dat, které v této práci neprezentuji.

potřeby využít tyto soubory i jinde (KML používá i nejrozšířenější komerční GIS program ArcGIS či třeba AutoCad).

V Elektronické encyklopedii historie je implementována poměrně jednoduchá funkcionalita, která umožňuje zobrazit si jakékoliv do ní zadané místo v aplikaci Google Earth. Při každém vložení místa je na základě zadaných GPS souřadnic automaticky vygenerován KML soubor, který je místu přiřazen, a v případě, že GPS souřadnice nejsou prázdné, se zobrazuje na kartě záznamu. Při každé změně GPS souřadnic je pochopitelně tento KML soubor nutno vygenerovat znovu. Lze si do budoucna představit i širší využití KML, ale tato jednoduchá varianta má tu výhodu, že díky automatickému generování KML souborů nepředstavuje pro redaktora vůbec žádnou práci navíc (krom zadání GPS souřadnic, které lze ale u geografických objektů tak jako tak očekávat) a nevyžaduje po něm jakoukoliv znalost ať už GIS systémů nebo KML. Vygenerovaný KML soubor obsahuje základní informace o místu a má následující tvar:

```
<?xml version='1.0' encoding='UTF-8'?>
<kml xmlns='http://earth.google.com/kml/2.2'>
  <Placemark>
    <name>Název místa</name>
    <description>Originální název místa</description>
    <Point>
      <coordinates>zeměpisná_délka,zeměpisná_šířka</coordinates>
    </Point>
  </Placemark>
</kml>
```

Zeměpisnou délku a šířku je nutno podle specifikace KML zadávat v desetinné soustavě, nikoliv v šedesátkové, tedy jako jedno číslo od -180 do 180, resp. od -90 do 90 s libovolným množstvím desetinných míst, nikoliv jako trojici stupně-minuty-sekundy jak jsou zadávány do EE, proto je nutno je pro účely vytvoření KML přepočítat:

Zeměpisná_souřadnice_pro_KML = stupně + minuty/60 + sekundy/3600.

6.1.6 Dokumenty se soubory

Speciální uzel typu „Dokument“ umožňuje, aby bylo možno do Elektronické encyklopedie historie vkládat i jiný než textový obsah – konkrétně obrázky (ve formátech JPEG, GIF a PNG) a jiné dokumenty (ve formátu PDF). Tato funkcionalita je realizována tak, že v redakční části aplikace, ve formuláři umožňujícím editovat uzel typu „Dokument“, je – krom běžných možností – ještě navíc přidána možnost vložit obrázek a možnost vložit PDF dokument. Každý redaktor může ke všem záznamům, na jejichž editaci má právo, nahrávat tyto soubory. Soubor je uložen na server a pokud jde o obrázek, je zároveň vytvořena jeho zmenšenina (o velikosti delší strany 150px). Tato zmenšenina se poté používá jako náhled v kartě záznamu, přes nějž je možno si zobrazit obrázek v plné velikosti. Každý obrázek/dokument má svůj název, který se v kartě záznamu zobrazuje, a který slouží k informování uživatele, co se v přiloženém souboru nachází.

Kvůli efektnějšímu zobrazování karet záznamu se u všech ostatních uzlů, které mají vazbu na dokument s přiloženým obrázkem, zobrazuje na kartě záznamu krom názvu provázaného dokumentu i tento obrázek, resp. jeho náhled. Pokud je u jednoho dokumentu přiloženo více obrázků, má autor možnost zvolit, který z nich je reprezentativní a má se tudíž zobrazovat na kartách svázaných uzlů. Aby bylo zobrazování karet záznamu ještě uživatelsky přívětivější, existuje možnost při zadávání

vazby mezi uzlem a dokumentem zvolit, že tento dokument (resp. jeho náhled) nemá být zobrazen na kartě záznamu mezi ostatními vazbami, ale vložen na začátek textového pole uzlu náležejícího, čímž je dosaženo efektu vložení obrázku do textu, který je na internetu běžný a uživatelům proto zřejmě připadá sympatickým.

Jelikož na rozdíl od obsahu databázových tabulek, s jejichž pravidelným zálohováním se počítá, zálohování souborů bude probíhat méně často, rozhodl jsem se implementovat pojistku proti neúmyslnému smazání souboru (který nemusí být v případě jeho ztráty snadné získat zpět). Ta spočívá ve způsobu odstraňování souborů, který je dvoufázový – první fáze, kterou může provádět každý redaktor, je z pouze zdánlivým smazáním souboru – po kliknutí na „smazat“ je soubor odstraněn ze seznamu souborů dokumentu přiřazených a nikde ve veřejné části aplikace dále nefiguruje. K fyzickému odstranění souboru ovšem nedošlo (a proto nebyl odstraněn ani odpovídající záznam z databázové tabulky, který byl ve skutečnosti pomocí příznaku pouze označen za neplatný). Skutečné fyzické odstranění souboru lze vynutit v oddělené části redakčního systému, do níž má ovšem přístup pouze šéfredaktor, a kde je možno zobrazit si všechny soubory označené za smazané a buď je obnovit (pokud dosud existuje původní dokument), a nebo definitivně odstranit a smazat ze serveru.

6.2 Implementace analýz v Elektronické encyklopedii historie

6.2.1 Analýzy grafů

Základem implementace všech tří definovaných úloh analýzy grafů, tedy hledání dosažitelných uzlů, hledání cest mezi zvolenými uzly a hledání souvisejících uzlů, jsou algoritmy pro procházení grafu. Pro všechny tři úlohy jsem zvolil algoritmus DFS (prohledávání do hloubky – depth-first search), který na rozdíl od BFS nevyžaduje použití fronty k ukládání seznamu uzlů právě procházené úrovně. Jak vyplyne z následujícího, pro danou úlohu, která není standardním procházením grafu, nýbrž její modifikací, je DFS vhodnější.

Prvním zásadním rozdílem algoritmu pro hledání dosažitelných uzlů oproti standardním grafovým algoritmům je, že modifikovaný algoritmus nepracuje s grafem explicitně uloženým v typickém tvaru používaném k reprezentaci grafu (seznam následníků nebo matice sousednosti[3]), ale s grafem uloženým ve formě záznamů v relační databázi. Tyto záznamy během své činnosti nijakým způsobem nemodifikuje, nemůže si tudíž značit každý uzel jako nezpracovaný/zpracovaný, jak je u algoritmů pro procházení grafů běžné. Druhým rozdílem oproti standardnímu DFS je, že ve všech třech úlohách není cílem algoritmu projít všechny uzly grafu, tedy navštívit každý z nich právě jednou, nýbrž nalézt cesty mezi určitými uzly v grafu. Při tom můžou být jednotlivé uzly navštíveny opakovaně (každý uzel může být součástí mnoha různých cest mezi některými jinými uzly), takže obecně jakékoliv uchovávání informace o tom, zda byl daný uzel už navštíven či nikoliv, nemá smysl. Jediné, čemu je potřeba zabránit, je, aby algoritmus při procházení grafu negeneroval cesty s cykly, tedy cesty, v nichž by se tentýž uzel opakoval vícekrát[*]. Proto je potřeba mít v každém kroku

[*] V takovém případě by formálně vzato ani nešlo o cestu, nýbrž o sled, protože definice cesty opakování uzlu přímo vylučuje. Algoritmy pro hledání cest v grafech často však ve skutečnosti nepracují s cestami, ale sledy[3]. Pro účely všech tří implementovaných úloh je však třeba používat skutečně cest a nikoliv sledů.

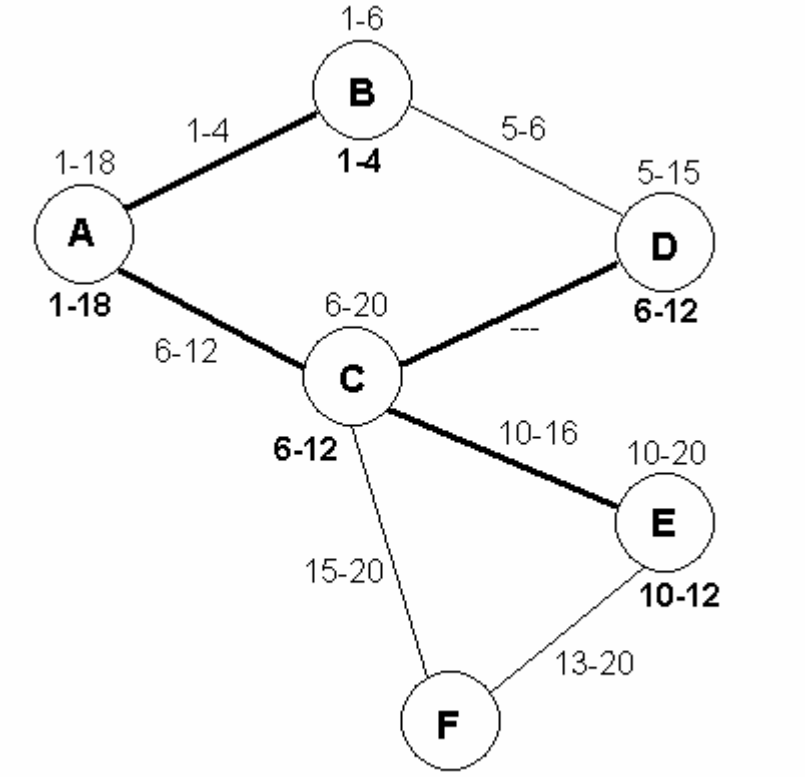
algoritmu k dispozici seznam uzlů tvořících dosud nalezenou část cesty, který bude využit právě k tomu, aby žádný z těchto uzlů nebyl zpracován vícekrát. V případě použití algoritmu BFS by takovýto seznam musel být udržován ve frontě společně s každým uzlem čekajícím na zpracování, kdežto při použití algoritmu DFS je možno tento seznam předávat algoritmu při každém rekurzivním zanoření jako dodatečný parametr. V obou případech se tak tento požadavek projeví na zvýšení prostorové náročnosti algoritmu.

Přestože časová a prostorová náročnost obou algoritmů (BFS i DFS) je teoreticky stejná, v praxi se skutečná prostorová náročnost obou algoritmů liší. Skutečná prostorová náročnost algoritmu DFS je totiž závislá na počtu úrovní, do nichž se algoritmus při procházení grafu zanořuje, kdežto skutečná prostorová náročnost algoritmu BFS je naopak závislá na počtu uzlů v nejhlubší z těchto úrovní. Jelikož sémantická síť je obecně hustě propojena, stupeň každého uzlu je spíše vyšší než nižší (navíc nejsou ze zpracování vylučovány už jednou navštívené uzly, jak standardní algoritmus předpokládá) a proto počet uzlů v nejhlubší úrovni bude obecně spíše vysoký než nízký. Hloubka, do níž je síť prohledávána, je naopak spíše malá – pro žádnou z navrhovaných úloh nemá smysl prohledávat síť hlouběji než do řádově jednotek úrovní (to souvisí s vlastnostmi sítě typu malý svět, viz kapitola 5). Proto je skutečná prostorová náročnost DFS při použití pro navržené analýzy v rámci Elektronické encyklopedie výrazně nižší než skutečná prostorová složitost BFS, což je důvodem pro použití právě tohoto algoritmu jako základu všech tří algoritmů pro analýzu obsahu EE.

6.2.2 Implementace analýz v rámci Elektronické encyklopedie historie

Při analyzování obsahu encyklopedie má uživatel možnost nastavit dvě modifikace procházení sítě. První z nich je možnost vybrat si pouze některé ze seznamu všech vazeb v systému existujících, které budou při procházení sítě používány, zatímco vazby ostatních typů budou ignorovány. U každé vazby lze nastavit možnost používat vazbu v obou směrech zvlášť (v seznamu se tedy nachází dvakrát, přičemž je nutno rozlišit, o který směr se v kterém případě jedná). To umožňuje používat nástroje pro analýzu sítě pro různé specifičtější účely, než je celková analýza sítě (např. zkoumat vztahy jen mezi vojenskými jednotkami či osobami, nebo naopak z obecnější analýzy některé vazby, které v daném okamžiku uživatel nepovažuje za relevantní, vynechat).

Druhou modifikací je volba, zda se při procházení sítě budou respektovat data existence uzlů a vazeb. Pokud není tato možnost zvolena, prochází se síť staticky, bez ohledu na jakákoliv data existence uzlů či vazeb a používají se tedy veškeré vazby (příp. vazby zvolených typů), které se v ní vyskytují. Pokud možnost respektovat data existence uzlů a vazeb zvolena je, považují se vazby v síti za rozdělené do dvou množin – vazby, které mají časové určení své existence (osoba zastávala funkci od doby X do doby Y) a vazby, které časové určení nemají (osoba X je rodičem osoby Y). U vazeb s časovým určením se v takovém případě berou při procházení sítě v úvahu pouze ty vazby, jejichž doba existence se překrývá s rozhodnou dobou. Ta je na počátku inicializována dobou existence uzlu, jež je vybrán jako počátek analýzy, a v každé další úrovni je dána jako průnik rozhodné doby z předchozí úrovně s dobou existence vazby, přes niž byl právě zpracovávaný uzel navštíven (viz následující schéma). Vazby, které časové určení nenesou, jsou rozděleny na další dva druhy. Ty, u nichž nemá smysl časové omezení uvažovat (například osoba X je rodičem osoby Y) jsou použity pro procházení všechny, bez ohledu na jakákoliv časová omezení. U těch, kde časové omezení roli hraje (např. osoba X se účastnila události Y), se berou v úvahu jen takové vazby, u nichž se s rozhodnou dobou překrývá doba existence uzlu, na něhož vedou. Následující schéma se pokouší tato pravidla ilustrovat.



Obr.6.1. – Ukázka části sítě ilustrující procházení sítě s ohledem na data existence. U každého uzlu i vazby je doba existence (u uzlu nahoře) a u každého uzlu dole (tučně) rozhodná doba. Tučně jsou vyznačeny vazby, které se při procházení sítě z počátečního uzlu A budou v případě respektování dat existence používat.

Z výchozího uzlu A budou použity obě vazby. V následujícím zanoření k uzlu B budou použity pouze vazby existující v období překrývajícím se s dobou, kdy existuje vazba mezi A a B, tedy žádná, neboť vazba na uzel D neexistuje v době, kdy existuje vazba mezi A a B. Podobně v případě uzlu C bude použita pouze vazba na uzel E, protože ta existuje aspoň částečně v době, kdy existuje vazba mezi uzly A a C, ale nikoliv vazba na uzel F. Vazba z uzlu C na uzel D bude použita v každém případě, protože doba existence uzlu D se překrývá s rozhodnou dobou z předchozího kroku, což je interval <6,12>. U uzlu E se pak nebude používat vazba na uzel F, protože ta neexistuje v době odpovídající průniku doby existence vazby mezi uzly C a E <10,16> a předchozího rozhodného intervalu, tvořeného v tomto případě dobou existence vazby mezi uzly A C <6,12>, což by byl interval <10,12>.

Ukázalo se také, že ačkoliv to tak na první pohled nemusí vypadat, vybírání uzlu k analýze pomocí možnosti zadat ručně jeho identifikátor (číslo) je pro uživatele při delší než demonstrační práci s těmito nástroji pohodlnější, než jeho vybírání pomocí procházení encyklopedie. Předpokládá se totiž, že si uživatel může v jednom okně otevřít obsah encyklopedie a ten si libovolně procházet a do druhého okna (kde má nachystány analýzy) jen přepisuje čísla uzlů, které chce analyzovat. Proto jsem nakonec zvolil tento způsob vybírání uzlů, přičemž pomocí s pomocí technologie ajax vždy po zadání identifikátoru uživateli zobrazím pro kontrolu, který uzel vybral.

6.2.2.1 Hledání dosažitelných uzlů

Nástroj pro hledání dosažitelných uzlů uživateli umožňuje zadat výchozí uzel a maximální hloubku, do níž má být síť prohledána, tedy maximální délku cesty v grafu. Krom výše zmíněných možností upravit chování nástroje (výběr použitých vazeb a respektování dat existence uzlů a vazeb) nabízí navíc možnost zvolit způsob zobrazování opakovaně nalezených uzlů, a to ze tří možností:

- ponechat – znamená, že opakovaně nalezené uzly budou prostě zobrazeny na svém místě, bez ohledu na to, kolikrát se ve výsledném seznamu budou vyskytovat; tato možnost umožňuje zobrazit kompletně část grafu sousedící se zvoleným uzlem až do zvolené vzdálenosti včetně překrývajících se částí (tj. takových, k nimž se dá ze zvoleného uzlu dostat více než jednou cestou); je potřeba si uvědomit, že to při zadání větších hloubek často vede k velice rozsáhlým seznamům, neboť celá síť je obecně hustě provázána a je v ní mnoho silně souvislých komponent, a proto se v takovémto seznamu bude množství uzlů vyskytovat vícekrát než jednou
- nezobrazovat – znamená, že opakovaně nalezené uzly budou z dalšího zpracování úplně vynechány, nebudou tedy ani vypsány, ani dále rozvíjeny; tato možnost umožňuje zobrazit množinu všech uzlů dosažitelných ze zvoleného uzlu cestou maximálně zvolené délky; v případě, že je zároveň požadováno respektování data existence vazeb/uzlů, je nutno si uvědomit, že může vést k vynechání některých uzlů, neboť do jednoho a téhož uzlu je možno se dostat několika různými cestami, z nichž s každou je spojena jiná rozhodná doba, přičemž síť je dále procházena jen s využitím první z nich
- nerozvíjet – znamená, že opakovaně nalezené uzly jsou ve výsledku zobrazeny, ale cesty z nich dále vedoucí jsou sledovány jen tehdy, pokud se při všech předchozích nalezení daný uzel vyskytoval na nižších úrovních než při aktuálním (tzn. pokud opakované rozvíjení cest z tohoto uzlu způsobí, že budou nalezeny ještě další uzly krom těch, které byli nalezeny při předchozím rozvíjení, neboť se bude postupovat do hlubší úrovně); tato možnost je jakousi kombinací obou předchozích a umožňuje zobrazit část sítě sousedící se zvoleným uzlem až do zvolené vzdálenosti a při tom částečně eliminovat opakované zobrazování překrývajících se částí[*]. Je-li požadováno respektování data existence uzlů, trpí stejným nedostatkem jako předchozí volba.

Díky použití algoritmu založeného na principu DFS je při procházení sítě možno nalezené uzly rovnou vypisovat do tabulky jakožto výsledek analýzy pro uživatele, který je dostatečně přehledný[**]. To znamená, že není nutno uchovávat strukturu celé nalezené části sítě v paměti pro pozdější výpis, což by představovalo velmi výrazné zvýšení paměťových i časových nároků.

6.2.2.2 Hledání cest mezi zvolenými uzly

Nástroj pro hledání cest umožňuje zadat dva různé uzly Encyklopedie a maximální délku cesty. Krom toho nabízí standardní možnosti úpravy chování (výběr používaných vazeb a zachování dat uzlů a vazeb).

---

[*] Aby mohlo být opakované zobrazování částí sítě eliminováno úplně, musel by být namísto DFS použit jako základ algoritmu BFS, což by umožnilo zobrazit každý podgraf jen jednou a při tom zobrazit sousedící část grafu kompletně, avšak použití BFS pro tento algoritmus by s sebou neslo výše zmíněné nevýhody

[**] Domnívám se, že graf lze, aspoň na dvourozměrné ploše monitoru, nejpřehledněji zobrazit pomocí tabulky sousedících uzlů. Veškeré grafické reprezentace grafu jsou, i přes svou vizuální přitažlivost, prakticky použitelné pouze pro velice malé grafy, neboť v okamžiku, kdy jejich velikost přeroste zobrazovací plochu, veškerá přehlednost se ztrácí a jejich grafické znázornění nepřináší oproti textovému žádnou výhodu, spíše naopak.

Algoritmus je podobný algoritmu pro Hledání dosažitelných uzlů, s tím že při procházení sítě pomocí DFS z počátečního uzlu nevypisuje nic na výstup, ale pouze si v případě, kdy narazí na zadaný koncový uzel, zapamatuje cestu k němu pro pozdější zobrazení. Výsledkem pak je seznam všech cest mezi zadanými dvěma uzly kratších nebo rovných zadané maximální délce, které jsou vypsány v pořadí od nejkratší po nejdelší.

6.2.2.3 Hledání souvisejících uzlů

Nástroj pro hledání souvisejících uzlů umožňuje zadat uzel a maximální délku cest, které se budou pro výpočet souvislosti uzlů používat. Navíc si uživatel může zvolit, kolik nejvíc souvisejících uzlů chce, aby mu nástroj zobrazil. Krom toho je opět možno zvolit používané vazby a zda se mají či nemají respektovat data existence uzlů a vazeb.

Souvislost dvou uzlů se počítá pomocí tzv. Katzovy míry jako vážený součet všech cest, které mezi oběma uzly existují:

$$s = \beta^0 * c_1 + \beta^1 * c_2 + ... + \beta^{n-1} * c_n,$$

kde c_x je počet cest délky *x*, *n* je maximální délka cesty a β je koeficient s hodnotou z intervalu $(0,1\rangle$, který určuje, jak rychle klesá důležitost cest s jejich rostoucí délkou

Algoritmus je prakticky stejný jako algoritmus pro hledání dosažitelných uzlů, při nalezení jakéhokoliv uzlu však není nutno nic vypisovat ani ukládat cestu k němu, nýbrž pouze poznačit si délku cesty, kterou se k uzlu došlo, aby po skončení procházení sítě mohl vypočítat souvislosti. Výsledkem je pak seznam uzlů, jejichž souvislost se zadaným uzlem je největší (to, kolik jich bude zobrazeno, si určí uživatel). U každého uzlu je zobrazena i hodnota jeho souvislosti s výchozím (Katzova míra), jež je však uváděna jen pro představu o rozdílech mezi souvislostmi jednotlivých uzlů, neboť nepředstavuje žádnou reálnou veličinu.

Hodnotu parametru β, používanou pro výpočet míry souvislosti, jsem zvolil na základě experimentování s nástrojem. Je zřejmé, že výsledek lze ovlivnit jednak nastavením maximální délky cest, které se budou pro výpočet souvislosti brát v potaz, a jednak právě nastavením hodnoty β, přičemž tato dvě nastavení by měla souviset – nízká hodnota β znamená, že příspěvek delších cest k míře souvislosti bude zanedbatelný, vysoká naopak způsobí, že cesty budou přispívat k míře souvislosti téměř bez ohledu na svou délku, což také není žádoucí. Proto jsem se rozhodl svázat hodnotu β s maximální délkou použitých cest, nastavenou uživatelem (toto nastavení je pro uživatele jistě průhlednější, proto zůstává možnost nastavit maximální délku cest a s hodnotou β uživatel přímo manipulovat možnost nemá). Hodnota β je vždy nastavena tak, aby příspěvek nejdelší cesty, která se při počítání míry souvislosti podle uživatelova zadání ještě má brát v potaz, k této míře byl násoben zhruba hodnotou 0,3. Jinými slovy tak, aby $\beta^n \approx 0{,}3$, kde *n* je uživatelem zvolená maximální délka cesty.

6.2.3 Shlukování

Pro implementaci úloh týkajících se shlukování bylo třeba zvolit jednak funkci vzdálenosti a jednak metodu shlukování. Jako funkci vzdálenosti dvou objektů jsem zvolil nejjednodušší možnou, tedy vzdálenost Manhattanovskou, která se dá stanovit prostým sečtením rozdílů hodnot všech navzájem si odpovídajících atributů obou objektů.

Výběr metody shlukování pak také není až tak komplikovaný, jak by se mohlo zdát. Metody založené na rozdělování jsem vyloučil, neboť vyžadují od uživatele zadání požadovaného počtu shluků (který on ovšem nemůže znát) a navíc mohou dávat při každém běhu jiné výsledky a doba jejich běhu je nepředvídatelná (což je u webové aplikace, která musí počítat například s omezením maximální možné délky provádění skriptu na serveru, problém). Metody hierarchické zase poskytují výsledek ve tvaru hierarchie rozdělení do shluků, a to si vynucuje od uživatele vybrání některého z mnoha konkrétních rozdělení – a těch je k, kde k je rovno počtu shlukovaných objektů, což je v případě uzlů v EE velmi vysoké číslo – a není jasné, jak by uživatel Encyklopedie takový výběr měl provádět (pokud nemá předem stanoven počet požadovaných shluků, což nemá). Vhodnou se tedy ukazuje být metoda DENCLUE, která má jediný problém, a to je nutnost zvolení parametru σ. Jeho řešením může být opakované provedení shlukování s různou hodnotou parametru σ a nalezení nejdelšího intervalu, na němž se výsledný počet shluků nemění, neboť odpovídající výsledek shlukování lze považovat za optimální[9]. Protože nejnáročnější částí algoritmu shlukování je výpočet vzdáleností mezi všemi objekty navzájem, a ten stačí provést pouze jednou, bez ohledu na to, kolikrát se pak – s různou hodnotou σ – samotné shlukování provádí, výpočetní náročnost by tímto postupem neměla být zvýšena natolik dramaticky, aby to zabránilo realizaci této metody.

6.2.4 Implementace shlukování v Elektronické encyklopedii historie

Obě úlohy – hledání podobných uzlů i shlukování uzlů – vyžadují možnost stanovit vzdálenost dvou uzlů encyklopedie na základě jejich struktury. Jak už bylo popsáno výše, jde o převedení grafu na množinu objektů s atributy odpovídajícími počtu hran jednotlivých typů. Obě úlohy jsou prováděny nad množinou všech objektů stejného typu (nemá smysl hledat např. vojenskou jednotku která je „strukturou podobná“ jisté události) a jejich prvním krokem je právě toto převedení uzlů (uložených jako záznamy v tabulkách databáze) na objekty s atributy. Protože bylo experimentálně zjištěno, že implementace vyžadující víc dotazů do databáze a méně práce s pamětí v rámci php skriptu je na použitém hostitelském serveru rychlejší než implementace snažící se dosáhnout co nejmenšího počtu dotazů do databáze a nahrazující je složitější manipulací s těmito daty v rámci php, je postup následující:

Pro každý uzel daného typu se projdou všechny databázové tabulky realizující všechny vazby uzlů tohoto typu a postupně se v každé z nich vyhledají záznamy aktuálního uzlu. Jejich počet (rozlišený navíc podle kategorie vazby) se pak postupně ukládá jako atributy aktuálního uzlu. Seznam uzlů s těmito atributy je pak výsledkem této podúlohy, jehož obě úlohy využívají ke svému běhu.

6.2.4.1 Hledání podobných uzlů

Implementace hledání podobných uzlů je přímočará – stačí projít seznam všech uzlů s jejich atributy a spočítat (Manhattanovskou) vzdálenost každého z nich od uživatelem vybraného. Ty, jejichž vzdálenost je nejmenší, jsou pak zobrazeny (počet zobrazených si uživatel zvolí). Kromě zvoleného uzlu je možno vybrat, zda se má podobnost počítat na základě přesného počtu vazeb jednotlivých kategorií, a nebo zda se mají u každé kategorie vazeb rozlišovat pouze tři hodnoty – žádná vazba dané kategorie, jedna vazba dané kategorie, více než jedna vazba dané kategorie (atributy objektů pak nabývají nikoliv hodnot odpovídajících přesnému počtu vazeb každé kategorie, ale 0 pro řádnou vazbu, 1 pro jednu vazbu a 2 pro více než jednu vazbu dané kategorie). Toto

zmenšení přesnosti výpočtu může vést k lepším výsledkům, neboť jsou zanedbány rozdíly v kardinalitě, které nemusí být pro potřeby určování podobnosti struktury uzlů tak významné.

Dále je, podobně jako u nástrojů založených na analýze grafu, možno zvolit, které vazby se mají, resp. nemají pro porovnávání struktury uzlů používat, což opět umožňuje vynechat ty, které uživatel pro konkrétní účel nepovažuje za důležité.

6.2.4.2 Shlukování uzlů

Operace shlukování metodou DENCLUE je při větším množství objektů časově i prostorově značně náročná. Její časově nejnáročnější částí je spočtení vzdáleností mezi každou dvojicí objektů, které je potřeba provést na počátku. Samotné shlukování už pak není tak náročné, provádí se ovšem opakovaně (s měnící se hodnotou parametru σ), což celkový čas potřebný na tuto operaci protahuje. Jelikož vzájemná vzdálenost jednotlivých objektů se při změně parametru σ nemění, nabízí se jako první opatření oddělit tyto dvě fáze a vzdálenost vypočíst pouze jednou (na začátku) a opakovaně provádět pouze samotného shlukování.

Operace shlukování se nám tak rozpadá na dvě samostatné úlohy – výpočet vzdálenosti a samotné shlukování. Aby se časově náročný výpočet vzdálenosti jednotlivých objektů nemusel – byť „pouze“ jednou – provádět pokaždé, když jakýkoliv uživatel spustí úlohu shlukování, rozhodl jsem se oddělit ho od samotné úlohy shlukování úplně. Právo spustit tento proces má pouze šéfredaktor a jeho výsledek je uložen do databázové tabulky, odkud ho použije samotný proces shlukování kdykoliv je kýmkoliv spuštěn. Šéfredaktor by se měl dle vlastního uvážení starat o to, aby v databázi byl přiměřeně aktuální záznam, neboť v databázi uložené vzdálenosti všech objektů pochopitelně neodrážejí změny, ke kterým v encyklopedii došlo po jejich uložení.

Příprava dat potom využívá (stejně jako hledání podobných uzlů) operaci vytvoření seznamu všech uzlů s atributy, který prochází a počítá vzdálenost každého uzlu od každého dalšího. Operace se dá urychlit pouze tím, že není třeba počítat celou matici vzdáleností, ale jen její polovinu, neboť vzdálenost je symetrická podle diagonály (do databáze se však přesto ukládá kvůli jednoduššímu přístupu k hodnotám celá matice). Jelikož při snaze uložit celou matici vzdáleností do lokální paměti u sekcí s velkým počtem uzlů hrozí, že dojde k vyčerpání paměti, kterou server dovoluje využít jednomu procesu, ukládají se hodnoty do databáze po jednotlivých řádcích hned po jejich spočítání a ve skriptu je tato paměť okamžitě uvolněna.

Samotné shlukování pak využívá matice vzdáleností jednotlivých uzlů uložené v databázi. Protože možný rozsah hodnot parametru σ by neměl být závislý na (uživateli neznámém) rozsahu hodnot vzdáleností a nabývat hodnot mezi 0 a 1, je potřeba i vzdálenosti normalizovat do rozsahu 0 až 1. Téhož efektu se pro rychlejší zpracování dá dosáhnout úpravou použité hodnoty σ jejím vynásobením maximální hodnotou z matice vzdáleností. Samotný proces shlukování začíná výpočtem funkce hustoty v každém bodě prostoru dat, v němž se nachází nějaký objekt. Protože jsem pro něj použil obdélníkovou funkci hustoty, je funkce hustoty nad každým objektem rovna počtu sousedů daného objektu, tedy objektů, jejichž vzdálenost od počítaného bodu je menší než σ. Při jejich počítání si zároveň ke každému objektu vytvářím pro pozdější použití seznam všech sousedů.

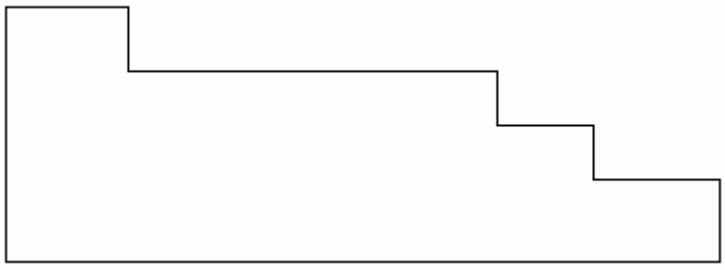
Hlavní etapa shlukování spočívá v hledání lokálních maxim funkce hustoty, které pak představují centra shluků. To probíhá na principu tzv. Hill Climbing metod – pro všechny objekty se hledá lokální maximum tím způsobem, že se v jejich okolí (to znamená mezi jejich sousedy) nalezne jiný objekt, jehož hodnota funkce hustoty je vyšší než u aktuálního objektu a k němuž zároveň nejstrměji stoupá, a ten pak považuje za nový aktuální objekt. Celý proces se opakuje dokud není

nalezen objekt, v jehož okolí se žádný další objekt s vyšší hodnotou funkce hustoty nevyskytuje – ten je pak středem shluku, do něhož patří i výchozí objekt (ten, z něhož prohledávání začalo). Pokud je tímto způsobem nalezen některý ze středů už existujících shluků, výchozí objekt je mu prostě přiřazen. Pokud je nalezeno nové lokální maximum, tedy nový střed, je nutno ho vytvořit, přiřadit mu výchozí objekt a všechny objekty, které se nacházejí ve stejném bodu prostoru (na rozdíl od standardní definice se v tomto pojetí může více objektů nacházet ve stejném bodě prostoru, tedy jejich vzdálenost být nulová, a to tehdy, pokud mají zcela shodnou strukturu vazeb).

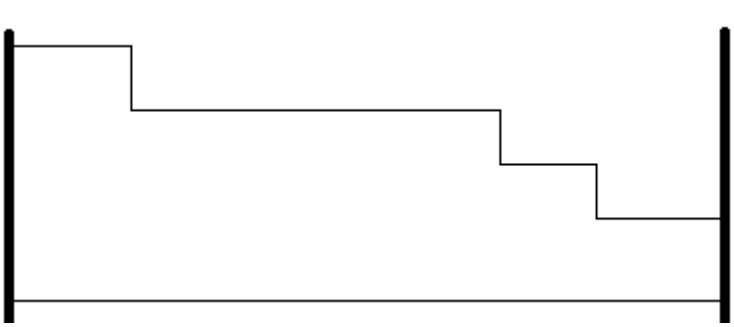
V praxi dále není nutno hledat od každého objektu odpovídající lokální maximum, ale stačí najít jakýkoliv objekt, který už je přiřazen některému shluku – od něj už bylo lokální maximum nalezeno dříve a aktuální objekt lze tedy rovnou přiřadit témuž shluku. Pro zrychlení procesu se mi dále osvědčilo přiřadit každému nově vytvářenému shluku všechny jeho sousedy, i když není jisté, že by při teoreticky správném postupu nebyly přiřazeny jinému shluku. Úspora času, která z tohoto mírného zmenšení přesnosti procesu vyplývá, je však velmi výrazná (až několikanásobná).

Konečně je třeba vyřešit problém nastavování hodnoty parametru σ. Jeho řešením je automatické hledání optimální hodnoty parametru σ, kterou lze definovat jako nejdelší interval hodnot σ, kdy se při použití jakékoliv hodnoty z tohoto intervalu nemění počet shluků, který je výsledkem procesu shlukování; tedy hledání nejstabilnějšího výsledku. V praxi to znamená opakované provádění shlukování s různými hodnotami parametru σ a hledání nejdelšího intervalu, na němž se počet shluků výsledku nemění. Výchozím intervalem hodnot parametru σ je interval <0;0,5>, neboť vyšší hodnota než 0,5 znamená téměř jistotu, že shlukování skončí vytvořením jednoho shluku obsahujícího všechny uzly (hodnota σ=0,5 znamená, že sousedem každého objektu budou všechny objekty, které se nacházejí až do poloviny maximální vzdálenosti mezi objekty od výchozího; tedy některý z objektů poblíž středu prostoru dat bude mít za sousedy všechny ostatní objekty a stane se středem jediného shluku) a úkolem je najít jeho nejdelší podinterval, na němž se výsledný počet shluků nemění.

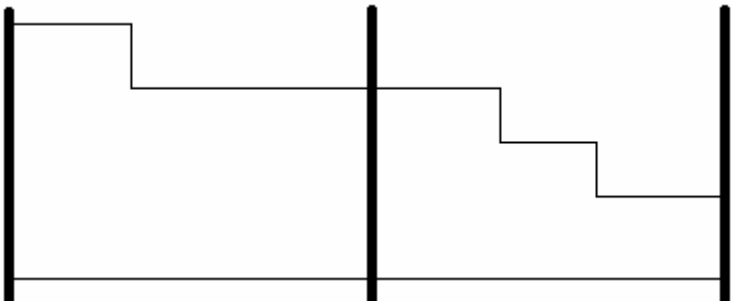
Ve své implementaci postupuji tak, že provedu shlukovou analýzu nejdříve pro minimální hodnotu σ (tj. 0) a pak pro maximální (tj. 0,5). Následně rekurzivně provádím shlukovou analýzu pro hodnotu σ rovnu vždy polovině intervalu mezi už spočítanými výsledky (na polovinu zmenšuji krok, se kterým prohledávám interval mezi minimální hodnotou σ a maximální hodnotou σ). Po každém cyklu zjišťuji, jaký je nejdelší nalezený interval, a proceduru je možno ukončit, když je dvojnásobek kroku menší, než rozdíl nejdelšího a druhého nejdelšího nalezeného intervalu. Postup znázorňuje následující ilustrace:



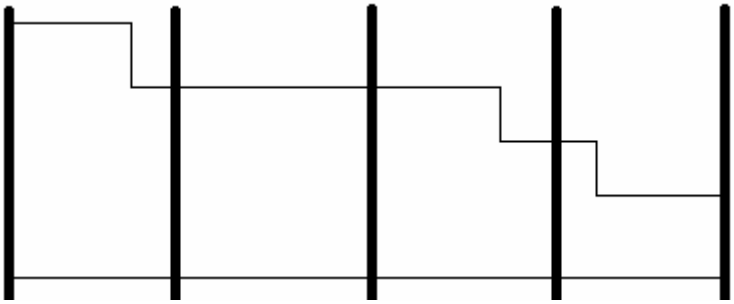
(1) zjednodušený příklad znázornění závislosti počtu shluků na hodnotě parametru σ



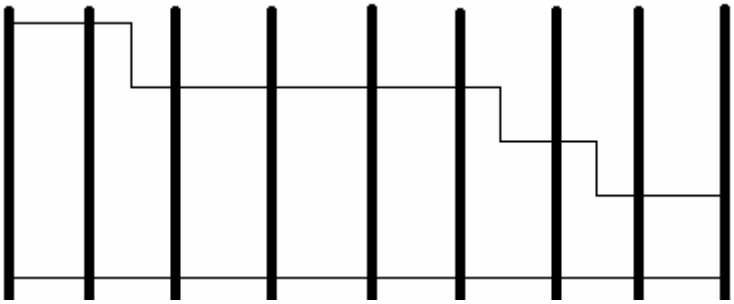
(2) v prvním kroku jsou zjištěny hodnoty na obou koncích intervalu



(2) v každém dalším kroku jsou zjištěny hodnoty vždy v polovině intervalu mezi už známými; nejdříve pouze jedna



(3) ve třetím kroku je nalezena dvojice stejných hodnot počtu shluků, protože však rozdíl nejdelšího a druhého nejdelšího intervalu (kterým je v tomto případě kterýkoliv z ostatních nalezených) není větší než dvojnásobek kroku (tedy počet nalezených bodů z nejdelšího intervalu není o dvě větší než počet bodů z druhého nejdelšího), hledání pokračuje



(4) ve čtvrtém kroku jsou nalezeny další dva body nejdelšího intervalu, z něhož tak nyní máme už čtyři body, zatímco do druhého nejdelšího intervalu (ať už je jím kterýkoliv z obou krajních) padly pouze body dva; je tedy zaručeno, že nemůžou být delší než první kandidát a algoritmus končí

Ještě je vhodné doplnit, že výsledek vedoucí k vytvoření jediného shluku obsahujícího všechny objekty, nebo k vytvoření tolika shluků, kolik je objektů (tedy každý objekt tvoří samostatný „shluk“) jistě nesplňuje představy uživatele o rozumném výsledku shlukování a proto, i kdyby byly podle výše uvedené definice teoreticky optimální, takovéto výsledky neberu v úvahu a hledám nejdelší interval s vynecháním intervalů, které k těmto výsledkům vedou.

Nakonec je nutno ještě zvolit, kterou hodnotu z nalezeného intervalu použít pro výsledné shlukování. Volím hodnotu odpovídající dolnímu okraji intervalu, protože shluky jsou pak kompaktnější (má to vliv na to, který objekt bude označen za střed shluku).

Po zobrazení výsledku shlukování je uživateli nabídnuta možnost provést shlukování znovu s tím, že si přeje rozdělení do většího, případně menšího, počtu shluků, než nalezené optimální. Toho se dosáhne jednoduše tak, že se výchozí interval nastaví na hodnotu <0; dolní hranice optimálního intervalu> pro více shluků, resp. <horní hranice optimálního intervalu; 0,5> pro méně shluků, a hledání optimální hodnoty σ se spustí znovu (nyní se tedy hledá optimální hodnota jen v rámci intervalu hodnot zajišťujícího, že všechny výsledky povedou k většímu, resp. menšímu počtu shluků, než jaký byl nalezen původně).

Po naimplementování tohoto algoritmu se ukázalo, že v případě shlukování většího počtu uzlů (řádově tisíce) je jeho časová složitost kvůli opakovanému provádění shlukování během hledání optimální hodnoty σ neúnosná. Navíc (jak už bylo uvedeno výše) není možno načíst celou matici vzdáleností do paměti, a proto je nutno při každém přístupu k údajům o vzdálenostech tato získávat dotazem přímo z databáze, což časovou náročnost dále zvyšuje. Proto jsem implementoval alternativní postup shlukování („ruční“), který rezignuje na snahu nalézt optimální hodnotu σ – namísto toho proběhne pouze jedno shlukování, a to s hodnotou σ zvolenou uživatelem. Uživatel má posléze samozřejmě možnost tuto hodnotu upravit tak, aby dostal větší/menší počet shluků, a shlukování spustit znovu.

7 Testování a zhodnocení

Samotná aplikace, tedy Elektronická encyklopedie historie, prošla poměrně intenzivním testováním, přičemž byla používána reálná data, simulující reálné nasazení takovéto aplikace. Část obsahu byla vkládána za účelem testování aplikace, druhá část už dokonce v rámci reálného projektu. Na základě těchto zkušeností lze prohlásit, že princip informačního systému pro historická data používající datový model založený na sémantické síti se osvědčil. Systém lze – pod podmínkou použití odpovídajícího kvalitního datového modelu – použít jak za účelem reprezentace modelu historické reality pro konkrétního badatele, tak za účelem sdílení tohoto modelu a informací mezi více badateli. Navíc je ho možno použít k přímé prezentaci dat v něm uložených. S využitím zkušeností z práce s tímto systémem bylo také postupně upraveno ovládání systému tak, aby bylo i pro redaktora co nejjednodušší a pokud možno intuitivní.

Bylo dále v praxi ověřeno, že vznikající sémantická síť splňuje charakteristiky sítě typu malý svět. Pouze co se týče poloměru sítě, je nutno brát zřetel na to, že díky nedostatečné zaplněnosti sítě daty se ve skutečnosti v praxi rozpadá na několik nesouvislých částí, neboť některé hrany, který by měly propojit tyto části, prostě nejsou zadány (což neznamená, že v reálném světě neexistují).

Implementované analytické nástroje zhodnotíme o něco podrobněji, každý zvlášť:

Nástroj Hledání dosažitelných uzlů přináší především možnost, jak aspoň trochu uživatelsky přívětivým způsobem zobrazit větší část sémantické sítě, než jen jeden uzel a jeho vazby, což je obsahem karty záznamu. Při testování se ukázalo, že užitečnost výsledků je výrazně vyšší, pokud je zatrhnuta volba „Respektovat data existence vazeb/uzlů“, což odráží skutečnost, že časové údaje jsou pro informační systém tohoto typu klíčové (s trochou nadsázky jsou pro něj totéž co zeměpisné souřadnice pro GIS systém). Přehlednost a použitelnost výsledků také klesá s množstvím uzlů, které jsou takto zobrazeny – v praxi se ukazuje, že nemá smysl vyhledávat dosažitelné uzly do vzdálenosti větší než zhruba 4-5, neboť to jednak vede k rozsáhlým a proto nepřehledným výsledkům, a jednak už na takovéto vzdálenosti začíná být souvislost dosažitelných uzlů s výchozím uzlem nevýznamná (od určité vzdálenosti už v sémantické síti typu malý svět začíná takříkajíc souviset všechno se vším a z tohoto faktu nelze získat žádnou užitečnou informaci pro konkrétní uzel).

Nástroj Hledání cest mezi uzly umožňuje ověření či zjištění, jaký je vzájemný vztah mezi dvěma zvolenými uzly. To může v praxi znamenat například odpověď na otázku, zda se osoba X nějakým způsobem znala s osobou Y, co měla osoba X společného s městem Z apod. Platí totéž co u Hledání dosažitelných uzlů: Jednak nemá smysl uvažovat příliš dlouhé cesty, neboť jakmile se délka cesty začne blížit poloměru sítě, je možno se jí dostat k jakémukoliv uzlu; navíc při zadaných delších cestách se začínají ve výsledku objevovat nesmyslné cesty mezi uzly, které kopírují některou z už nalezených kratších cest a přidávají do ní několik – sémanticky zbytečných – uzlů navíc. A jednak užitečnost výsledků výrazně zvyšuje zatrhnutí volby „Respektovat data existence vazeb/uzlů“. Užitečným se také ukázala možnost vybrat jen některé typy vazeb, které se mají při hledání cest brát v potaz, resp. možnost některé typy vazeb vynechat. Například pobyt dvou osob v tomtéž velkoměstě (byť ve stejnou dobu) ještě nezakládá jakoukoliv reálnou souvislost těchto osob, kdežto například dislokace dvou vojenských jednotek na stejném místě už jistou souvislost mezi nimi vyvolává a je tudíž vždy na uživateli, aby se rozhodl, které vazby jsou pro něj v konkrétním případě relevantní.

Nástroj Hledání souvisejících uzlů je asi nejužitečnějším z implementovaných analytických nástrojů. Dovoluje totiž uživateli snadno najít další uzly v encyklopedii, s nimiž zvolený uzel souvisí, a tím mu nabízí rychlé zjištění kontextu, ke kterému by se jinak musel dopracovávat sám

procházením sítě v okolí uzlu. I zde platí že je vhodné respektovat při hledání souvisejících uzlů data existence vazeb a uzlů, a že může být užitečné mít možnost z výsledku odfiltrovat některé typy vazeb. Zásadní je však nastavení hloubky, tedy maximální délky cesty, která se má ještě do výsledku započítávat. Jako nejvhodnější se opět ukazuje hodnota okolo 4-5. Pro menší hloubku se některé reálné souvislosti ztrácejí a při větší hloubce se ve výsledku už naopak objevují i uzly, jejichž souvislost se zvoleným uzlem je natolik nepřímá, že v praxi nemá žádný efekt.

Užitečnost zbývajících dvou nástrojů, založených na principech používaných při shlukové analýze, je poněkud diskutabilní. Základním problémem shlukování, který se týká jak nástroje umožňujícího shlukování samotné, tak nástroje pro hledání podobných uzlů na základě struktury, je nutnost, aby byly kompletně vyplněny všechny (relevantní) údaje u uzlů sekce, jejíž shlukování chceme provádět. Dá se ovšem s úspěchem pochybovat o tom, zda takového stavu bude v průběhu používání encyklopedie kdy dosaženo, neboť z principu její funkce plyne, že velká část jejího obsahu pravděpodobně vždy budou „okrajové“ uzly, tedy takové, které nejsou samotným předmětem bádání, ale nějak s ním souvisejí, a do encyklopedie se tudíž dostaly jen díky této své vazbě; nebudou však kompletně vyplněny. Jak bylo uvedeno v kapitole 5, velké množství uzlů v sémantické síti typu malý svět má pouze malý počet vazeb na ostatní uzly – v praxi to jsou právě tyto uzly. Je proto nutno smířit se s tím, že struktura uzlů odráží mnohem více stav naplněnosti encyklopedie daty než stav reálných entit, které tato data reprezentují. Proto i nástroj Hledání podobných uzlů může spíše než k hledání skutečně podobných entit sloužit k nalezení podobností v rámci stylu zadávání dat do encyklopedie. Testováním bylo ověřeno, že tímto způsobem je možno ho používat.

Samotné shlukování (nástroj Shlukování) je časově velice náročná operace (což je úděl všech úloh dolování dat) a v případě velkého množství dat trvá příliš dlouho na to, aby se mohlo stát standardní operací prováděnou běžně uživateli. Při testování bylo zjištěno, že aspoň v jistém smyslu použitelným výsledkem tohoto procesu jsou nalezené odlehlé hodnoty (tedy shluky o jednom uzlu, případně několika málo uzlech), protože ty upozorňují na neobvyklé uzly v encyklopedii (což v praxi svědčí nejpravděpodobněji o nějakém neobvykle kompletně vyplněném uzlu).

8 Závěr

Cílem této práce bylo navrhnout a vytvořit webovou aplikaci s názvem Elektronická encyklopedie historie, tedy informační systém pro historiky, a navrhnout a implementovat nástroje, které umožní analýzu dat v tomto systému.

Byly proto analyzovány požadavky na takovou aplikaci a stanoveny základní principy jejího fungování. V návaznosti na to byl navržen webový informační systém, který umožňuje splnit požadavky na snadnou dostupnost a jednoduchost ovládání. Dále bylo navrženo vytvořit datový model pro ukládání dat v systému na principu sémantické sítě, což umožňuje provádět analýzy dat postavené zčásti na grafových algoritmech a zčásti na dolování dat v grafech, resp. v sociálních sítích. Návrh konkrétního datového modelu pro EE jsem do této práce nezahrnul, neboť jeho podrobné zkoumání značně vybočuje z tématu této práce; zařadil jsem proto jen jeho náčrt, obsahující pouze základní a obecné informace o něm.

Navržený informační systém byl implementován a byla otestována jeho funkčnost. Systém splnil požadavky, které byly na počátku práce stanoveny, a ukázal se být i v praxi použitelným a užitečným nástrojem. Do budoucna lze jistě očekávat další změny datového modelu reflektující potřeby uživatelů. V případě širšího použití bude zcela jistě nutno vypracovat metodiku práce s takovýmto systémem, která zajistí, aby jakýkoliv redaktor (i takový, který ve skutečnosti nemá ponětí o tom, co je to sémantická síť) používal systém správným způsobem, to znamená aby systém mohl splnit to, co se od něj očekává (sdílení informací v definovaném tvaru, možnost analýzy vložených informací). Klíčové je definování a dodržení sémantiky jednotlivých uzlů (resp. jejich atributů) a především vazeb a jejich jednotné používání. To však musí být prováděno v úzké součinnosti s uživateli-redaktory, a především odborníky na danou oblast, tedy historiky. Dále by bylo možné sílu systému zvětšit poskytnutím nějakého prostředku pro zápis komplexních dotazů nad obsahem sémantické sítě, využívajících sémantiky jednotlivých vazeb a uzlů, který by umožňoval zodpovězení specifických dotazů uživatele (např. „Jak se jmenoval ten Napoleonův generál, který se zúčastnil ruského tažení a zemřel někdy ve třicátých letech 19. století?“).

Hlavním přínosem této práce je demonstrace možností analýz dat v informačním systému s datovým modelem založeným na sémantické síti. Tím se také implementovaný systém liší od všech podobných existujících systémů. Byla prokázána použitelnost a užitečnost všech tří navržených a implementovaných nástrojů vycházejících z postupů analýzy grafů. U nástrojů vycházejících z principů dolování dat byla ověřena funkčnost a teoretická použitelnost, praxe však ukázala, že užitečnost takovýchto nástrojů v rámci implementovaného informačního systému není veliká. Za očekáváním zůstal především nástroj na hledání podobných uzlů na základě struktury, od něhož jsem si sliboval možnost nalézt ostatní záznamy v Encyklopedii, které jsou stejného druhu jako zvolený uzel (např. další politické strany, ostatní vedoucí činitelé států apod.). Zde by se dalo uvažovat o zvýšení „rozlišovací schopnosti“ tohoto nástroje pomocí zahrnutí dalších parametrů mezi atributy, které definují onu strukturu uzlu, jejíž podobnost se vyhodnocuje. Kupříkladu časové údaje, resp. doby trvání jednotlivých vazeb, informace o typu uzlu, na něhož vazba vede, v extrémním případě víceúrovňové vyhodnocování atributů – tzn. za atributy uzlu považovat i atributy jeho sousedů (uzlů, na něž vedou vazby z vyhodnocovaného uzlu) a tím zohlednit i strukturu okolí uzlu v sémantické síti. Problémem, který však není asi žádným způsobem možno odstranit, neboť je zakotven přímo v principu fungování celé aplikace, je to, že tyto analýzy ve skutečnosti spíše než charakteristiky v systému uložených reprezentací entit reálného světa zkoumají charakteristiky zaplněnosti encyklopedie a kompletnosti zadaných informací u jednotlivých uzlů. Totéž platí samozřejmě i pro

nástroje vycházející z metod analýzy grafů, avšak u nich se to zdaleka neprojevuje tak významně – zkoumají totiž ve skutečnosti vždy omezenou, poměrně malou část sítě, takže když jsou spuštěny na aspoň částečně zaplněné oblasti (jinde nemá smysl od nich očekávat smysluplné výsledky), poskytují výsledky odrážející už skutečně spíše charakteristiku entit reálného světa v encyklopedii modelovaných. Naopak nástroje založené na metodách dolování dat zkoumají ve skutečnosti vždy celý prostor dat (resp. v tomto případě všechny uzly vybrané sekce) a proto se na nich nekompletnost zadaných dat projeví vždy.

Závěrem je tedy možno konstatovat, že práce ukázala možnosti analýzy dat v realizovaném informačním systému pro historiky, prokázala užitečnost nástrojů založených na metodách analýzy grafů a potíže, se kterými se setkala snaha použít za tímto účelem postupy shlukové analýzy.

Literatura

[1] Wikipedia. Dokument dostupný na URL:
http://en.wikipedia.org/wiki/Semantic_network (prosinec 2009)
http://en.wikipedia.org/wiki/Graph_isomorphism (prosinec 2009)

[2] ABZ slovník cizích slov. Dokument dostupný na URL:
http://slovnik-cizich-slov.abz.cz/ (prosinec 2009)

[3] Masopust, T.: Grafové algoritmy. (Materiál k předmětu GAL vyučovanému na FIT) Dokument dostupný na URL: https://www.fit.vutbr.cz/study/courses/GAL/public/gal-text.pdf

[4] Sowa, J.F.: Semantic Networks.
Dokument dostupný na URL: http://www.jfsowa.com/pubs/semnet.htm

[5] Han, J., Kamber, M.: Data Mining: Concepts and Techniques. Morgan Kaufmann Publishers. ISBN 1-55860-901-3.

[6] Boonstra, O., Breure, L., Doorn, P.: Past, present and future of historical information science. Amsterdam, 2004. ISBN 90-6984-413-3. Dokument dostupný na URL:
http://www.iono.noa.gr/hellinomnimon/Books%20and%20Papers/historical%20information%20science.pdf

[7] Juan, M.: Temporal GIS and Spatio-Temporal Modeling. Dokument dostupný na URL:
http://www.ncgia.ucsb.edu/conf/SANTA_FE_CD-ROM/sf_papers/yuan_may/may.html

[8] Steyvers, M., Tenenbaum, J.: The Large-Scale Structure of Semantic Networks
Dokument dostupný na URL: http://psiexp.ss.uci.edu/research//papers/smallworlds.pdf

[9] Kapavík, R.: Získávání znalostí z dat – shlukovací algoritmy. Brno, 2008, bakalářská práce, FIT VUT v Brně.

Přílohy

Přílohou této práce je CD obsahující zdrojový text práce, zdrojové kódy aplikace, uživatelskou příručku, redaktorskou příručku a programátorskou příručku. Aplikace je provozována na adrese http://ee.valka.cz.