

MASARYKOVA UNIVERZITA
FAKULTA INFORMATIKY



Implementace republikové databáze zařízení pro nakládání s odpady

BAKALÁŘSKÁ PRÁCE

David Kabáth

Brno, podzim 2007

Prohlášení

Prohlašuji, že tato bakalářská práce je mým původním autorským dílem, které jsem vypracoval samostatně. Všechny zdroje, prameny a literaturu, které jsem při vypracování používal nebo z nich čerpal, v práci řádně cituji s uvedením úplného odkazu na příslušný zdroj.

Vedoucí práce: Mgr. Michal Hejč

Shrnutí

Náplní této práce bylo analyzovat dostupné zdroje mapující zařízení, která nakládají s odpady v ČR a pokusit se nalézt obdobné zdroje i v zahraničí. Druhou částí byla implementace databáze daty z nalezených zdrojů (pouze ČR). Dále pak vytvořit internetové rozhraní pro komunikaci s databází, které bude poskytovat vyhledávání záznamů dle vybraných atributů. Internetové rozhraní by mělo splňovat pravidla pro Informační systémy ve státní správě (ISVS). Poslední částí práce byl návrh na optimalizaci systému v ČR evidujícího zařízení nakládající s odpady.

Poděkování

Na tomto místě bych rád poděkoval svému vedoucímu práce, Mgr. Michalu Hejčovi, za jeho pomoc a cenné rady.

Klíčová slova

odpadové hospodářství, odpady, Ministerstvo životního prostředí (MŽP), databáze, HTML, MySQL, CakePHP

Obsah

1	Úvod	2
1.1	Účel a cíl práce	2
1.2	Plánovaný postup práce	3
2	Analýza zdrojů	4
2.1	Analýza zdrojů v České republice	4
2.2	Analýza zdrojů v Evropě	9
3	Použité nástroje	10
3.1	Case Studio	10
3.2	EasyPHP	10
3.3	CakePHP	11
3.4	PSPad	11
4	Datový model	12
4.1	Datový model	12
4.2	Výstavba databáze	13
5	Naplnění databáze	17
5.1	Rozdělení podle formátů	17
5.1.1	HTML	17
5.1.2	XLS	22
5.1.3	DOC	22
5.1.4	PDF	23
5.2	Rozdělení podle krajů	23
5.3	Vkládání získaných dat	26
6	Tvorba webového rozhraní	28
6.1	Standardy ISVS	28
6.2	Tvorba webového rozhraní	29
6.2.1	Vyhledávání	29
6.2.2	Nápověda	31
6.2.3	O webu	31
6.2.4	Podrobnosti	31
6.2.5	Povolené odpady	31
6.2.6	Katastrální území	31
7	Další využití a možnosti	32
7.1	Aktivní zapojení MŽP ČR	32
7.2	Pasivní zapojení MŽP ČR	33
7.3	Další návrhy	34
8	Závěr	35
A	struktura webu při použití frameworku CakePHP	38

Kapitola 1

Úvod

V dnešní době, kdy na nás ze všech stran doráží mnoho zpráv o tom, jak je důležité třídit odpad, již nelze zůstat nedotčen tímto ekologickým tématem.

Nicméně se nabízí otázka: proč třídit odpad? Důvodů je opravdu mnoho. Tím nejhlasitějším je určitě globální oteplování [1]. Jde o termín popisující nárůst průměrné teploty zemské atmosféry a oceánů pozorované v posledních dekádách. Vědecké výzkumy konstatují, že tento nárůst může být od konce 19. století mezi 0,4 – 0,8°C a je pravděpodobné, že tomu napomohly různé lidské aktivity. Společným jmenovatelem těchto aktivit je stále se zvyšující množství oxidu uhličitého, který se uvolňuje při spalování fosilních paliv.

Avšak globální oteplování není tím jediným důvodem proč třídit odpady. Jistě, jde o zajímavý a velmi populární pojem, jenže čemu vlastně pomůžeme, když budeme třídit odpady [2]? Všichni víme, že Země má pouze omezené množství nerostných surovin, které může v budoucnu postupně mizet. A proto je třeba zavádět recyklační technologie, kdy již použité materiály můžeme dále využívat. Pokud například při výrobě papíru použijeme starý papír namísto dřeva, snižuje se energetická spotřeba až o čtvrtinu. Znečištění vzduchu je pak dokonce až o tři čtvrtiny nižší [3]. A právě na tomto příkladu je vidět idea třídění odpadu. Pokud se bude velké množství odpadů opětovně využívat, určitě například ubyde skládek a tím se sníží i negativní dopad na životní prostředí. Zcela jistě také ubyde množství škodlivých látek v ovzduší.

A právě proto je třeba třídit odpady. Jde totiž pouze o změnu životního stylu, která zabere minimum času. V dnešní době je prvotní odpovědnost za recyklaci kladena na domácnosti, protože pokud se odpad smíchá, tak je jen velmi obtížné a nákladné jej roztřídit a dále zpracovat. Proto je nutné třídit odpady již v domácnostech. Pro nejzákladnější tříděný odpad jako je sklo či papír lze použít kontejnery umístěné ve všech městech a většinou i na vesnicích. Pro ostatní, ne tolik standardní materiály jako je železo či jiný materiál, jsou zřízena speciální sběrná místa. Ta jsou společně se zpracovateli těchto materiálů evidována na úrovni jednotlivých krajů [4]. A právě analýzou těchto seznamů a vytvořením celorepublikové databáze se pokusí tato práce přispět k zlepšení efektivity svozu a následného využívání odpadů.

1.1 Účel a cíl práce

Cílem této práce bylo vytvoření celorepublikové databáze zařízení pro nakládání s odpady tak, aby mohla být využívána zejména menšími podnikateli a obcemi, ale také širokou veřejností. Účelem tedy bylo, aby byla jednoduchá, přehledná s propojením na další vyhledávání informací o jednotlivých firmách zabývajících se zpracováním odpadů na portálech jako je Živnostenský rejstřík [5], Administrativní registr ekonomických subjektů [6] nebo obchodní rejstřík [7].

1.2 Plánovaný postup práce

- analýza zdrojů o odpadovém hospodářství a jejich dostupnosti na Internetu. Nejprve v České republice po jednotlivých krajích, a poté se pokusit nalézt obdobné portály v ostatních zemích
- popis výsledků analýzy po jednotlivých krajích a možností získávání z nich informace. Popis dalších možných portálů, na které by mohla být databáze připojena.
- popis použitelných nástrojů pro vytvoření datového modelu i tvorby a naplnění databáze samotné
- návrh datového modelu budoucí databáze
- vkládání dat do databáze - popis postupu formátování a získávání dat z jednotlivých krajských stránek
- vytvoření internetových stránek s možností vyhledávání v databázi
- zhodnocení výsledného stavu, popis možných vylepšení, možnosti využití do budoucna

Kapitola 2

Analýza zdrojů

2.1 Analýza zdrojů v České republice

Jako zdroj dat (seznam zařízení zpracovávající odpady) jsem používal internetové stránky jednotlivých krajů, na kterých jsem se vždy snažil najít sekci s provozovateli sběrných dvorů, které dostaly povolení od krajského úřadu. Ta jednotliví provozovatelé dostávají v souladu se zákonem § 14 odst. 1 zákona č. 185/2001 Sb. [4]:

§ 14 Souhlas k provozování zařízení k využívání, odstraňování, sběru nebo výkupu odpadů (1) Zařízení k využívání, odstraňování, sběru nebo výkupu odpadů lze provozovat pouze na základě rozhodnutí krajského úřadu, kterým je udělen souhlas k provozování tohoto zařízení a s jeho provozním řádem (dále jen "souhlas k provozování zařízení"). V řízení předcházejícím vydání tohoto rozhodnutí musí krajský úřad posoudit všechna zařízení, která s těmito činnostmi souvisejí. Souhlas k provozování skládek nebezpečných odpadů se uděluje na dobu určitou, nejvýše na 4 roky. Doby platnosti souhlasu krajský úřad prodlouží na základě žádosti provozovatele skládky nebezpečných odpadů vždy nejvýše na další 4 roky, pokud jsou splněny podmínky a plněny povinnosti při provozování skládky stanovené tímto zákonem a prováděcím právním předpisem.

Jelikož se jedná o analýzu dostupnosti seznamů zařízení na jednotlivých krajských internetových portálech z pohledu běžného uživatele, myslím tedy, že není třeba analyzovat strukturu či datový model. Úkolem analýzy tedy bylo projít internetové portály všech 14 krajů České republiky a porovnat jednotlivá poskytovaná data a formáty. Pro srovnávání jsem se rozhodl uvádět několik atributů:

cesta – způsob, jakým se dá k seznamu provozovatelů v daném kraji "doklikat". V jaké hloubce se seznam vyskytuje.

adresa – URL, přes kterou se dá přímo k seznamu přistoupit

hodnocení – ohodnocení přístupnosti pro uživatele, který se snaží najít seznamy provozovatelů z krajské adresy (např. <http://www.kraj-jihocesky.cz/>), ohodnocení funkčnosti samotného ISOHu (Informační systém odpadového hospodářství), jeho kladů a záporů.

formát – formát, v jakém je seznam uložen (HTML, XLS, DOC, PDF). Jedná se zejména o hodnocení možnosti dalšího zpracování daného formátu. Více v kapitole 5 (Naplnění databáze).

atributy – atributy poskytovaných záznamů.

Jihočeský kraj

cesta – <http://www.kraj-jihocesky.cz/> - Krajský úřad – Odbor životního prostředí, zemědělství a lesnictví – seznam oprávněných osob k nakládání s odpady na území Jihočeského kraje – seznam oprávněných osob k nakládání s odpady na území Jihočeského kraje včetně jejich povolených odpadů

adresa – <http://www.kraj-jihocesky.cz/websouhlasy/index.php?akce=all&page=1>

hodnocení – špatná přístupnost, dobrý ISOH, různé druhy vyhledávání

formát – HTML

atributy – Provozovatel - oprávněná osoba, IČ, ulice, PSČ, obec, Provozovna - provozovna, ulice, PSČ, obec, katastrální území (název, katastrálního území, katastr, parcela), kódy zařízení, povolené odpady (katalogové číslo, kategorie odpadu, název odpadu, kód nakládání), platnost

Jihomoravský kraj

cesta – <http://www.kr-jihomoravsky.cz/> - informace - Krajský informační systém odpadového hospodářství

adresa – http://archiv.kr-jihomoravsky.cz/php/jmk/form.php?function=search&sql=&page=0&table_name=odpady&order=NAZEV

hodnocení – dobrá přístupnost, přehledný systém ISOH

formát – HTML

atributy – typ zařízení, IČ, název, obec, poznámka

Karlovarský kraj

cesta – <http://www.kr-karlovarsky.cz> – Životní prostředí – Odbor životního prostředí a zemědělství

adresa – <http://www.kr-karlovarsky.cz/kraj/.cz/krajskyurad/odbory/zivotni/>

hodnocení – poměrně složité doklikání k cíli a zbytečné rozdělení jednotlivých odpadů do několika souborů

formát – XLS, HTML

atributy – název firmy, IČ, adresa firmy, adresa provozovny, kontaktní osoba, telefon, e-mail, typy přijímaných vozidel, povolení provozu zařízení vydáno kým, platnost od, platnost do, rozsah povolení, upřesnění firmy, činnost

Královéhradecký kraj

cesta – <http://www.kr-kralovehradecky.cz> – životní prostředí a zemědělství – Aktuální informace z životního prostředí a zemědělství – Odpadové hospodářství – Přehled zařízení k nakládání s odpady v Královéhradeckém kraji – Přehled zařízení k nakládání s odpady v Královéhradeckém kraji

adresa – http://www.kr-kralovehradecky.cz/assets/files/4911/seznam-zařízení_podle_14_z_kona_o_odp.pdf

hodnocení – dobrá přístupnost, žádný ISOH, pouze seznam míst bez možnosti vyhledávání

formát – PDF

atributy – okres, typ zařízení, umístění, provozovatel, telefon, povoleno do

Liberecký kraj

cesta – <http://kraj-lbc.cz/> - informační systém životního prostředí – Odpadové hospodářství Libereckého kraje – Přehled zařízení k nakládání s odpady - vyhledávání

adresa – <http://www.kraj-lbc.cz/zivotniprostredi/souhlasy/index.php?akce=all&page=1>

hodnocení – složitější dostupnost, nicméně po dosažení cíle velmi přehledný ISOH s různými druhy vyhledávání a zobrazením na mapě

formát – HTML

atributy – Provozovatel - oprávněná osoba, IČ, ulice, PSČ, obec, Provozovna - provozovna, ulice, PSČ, obec, katastrální území (název, katastrálního území, katastr, parcela), kódy zařízení, povolené odpady (katalogové číslo, kategorie odpadu, název odpadu, kód nakládání), platnost

Moravskoslezský kraj

cesta – <http://kr-moravskoslezsky.cz> – životní prostředí – Odpadové hospodářství – Seznam zařízení k využívání, odstraňování, sběru nebo výkupu odpadů

adresa – http://www.kr-moravskoslezsky.cz/zp_0401.html

hodnocení – v porovnání s ostatními kraji průměrná dostupnost, bez možnosti vyhledávání, s rozdělením různých druhů zařízení do různých souborů

formát – HTML

atributy – provozovatel, IČ, umístění, zařízení, platnost do

Olomoucký kraj

cesta – <http://www.kr-olomoucky.cz/> - Olomoucký kraj - Životní prostředí a zemědělství - Odpadové hospodářství - Seznam zařízení ke sběru-výkupu-využívání nebo odstraňování odpadů na území OK

adresa – <http://websouhlasy.kr-olomoucky.cz/index.php?akce=all&page=1>

hodnocení – přehledná mapa webu Olomouckého kraje, pomocí které se lze poměrně intuitivně dostat k požadovaným materiálům. Ty nabízí několik druhů vyhledávání a přehledný systém

formát – HTML

atributy – Provozovatel - oprávněná osoba, IČ, ulice, PSČ, obec, Provozovna - provozovna, ulice, PSČ, obec, katastrální území (název, katastrálního území, katastr, parcela), kódy zařízení, povolené odpady (katalogové číslo, kategorie odpadu, název odpadu, kód nakládání)

Pardubický kraj

cesta – <http://www.pardubickykraj.cz/> - Plán odpadového hospodářství PK – Tabulky – Ostatní provozovaná zařízení nebo Oprávněné osoby

adresa – <http://www.pardubickykraj.cz/externi/ozpz/poh/prezentace/prezent.htm>

hodnocení – velmi špatná dostupnost dat, možná neaktuálnost informací

formát – XLS

atributy – název firmy, adresa, IČ, kód obce, zařízení, souhlas, platnost povolení, kontakt, provozní hodiny, vybavení, kapacita

Plzeňský kraj

cesta – <http://www.kr-plzensky.cz/> - Domácí stránka - Životní prostředí - Odpady - Seznam zařízení v Plzeňském kraji k využívání, odstraňování, sběru nebo výkupu odpadů včetně autovraků

adresa – <http://www.kr-plzensky.cz/article.asp?itm=24345>

hodnocení – intuitivní cesta k požadovaným datům, evidovaná data jsou nepřehledná a nejednotného formátu

formát – DOC

atributy – provozovatel, zařízení, odpady, datum souhlasu, souhlas do

Hlavní město Praha

cesta – <http://www.praha-mesto.cz/> - Životní prostředí – Odpady – Témata – Seznam oprávněných osob - Zobrazit všechna provozovaná zařízení na území Hlavního města Prahy

adresa – <http://212.67.66.249/odpady/souhlasy/index.asp?akce=all>

hodnocení – intuitivní cesta k datům, místy lehce nepřehledná. V cíli pak přehledný systém s různými druhy vyhledávání

formát – HTML

atributy – Provozovatel - oprávněná osoba, IČ, ulice, PSČ, obec, Provozovna - provozovna, ulice, PSČ, obec, kódy zařízení, povolené odpady (katalogové číslo, kategorie odpadu, název odpadu, kód nakládání)

Středočeský kraj

cesta – <http://www.kr-stredocesky.cz/> - Životní prostředí – Odpadové hospodářství
- <http://stredoceske-odpady.cz>

adresa – <http://www.stredoceske-odpady.cz/?sid=19a4b5c6f279fbd6e1a2aa704224067e&lang=cz&uzel=88>

hodnocení – intuitivní cesta, zřízení vlastní stránky pro odpadové hospodářství, přehledný ISOH s různými druhy vyhledávání

formát – HTML

atributy – Provozovatel - oprávněná osoba, IČ, ulice, PSČ, obec, provozovna - provozovna, ulice, PSČ, obec, katastrální území (název, katastrálního území, katastr, parcela), kódy zařízení, povolené odpady (katalogové číslo, kategorie odpadu, název odpadu, kód nakládání), platnost

Ústecký kraj

cesta – <http://www.kr-ustecky.cz> – Životní prostředí – Ústecký kraj – Dokumenty odboru – Přehled zařízení k nakládání s autovraky/Přehled zařízení k nakládání s odpady

adresa – http://zivotniprostredi.kr-ustecky.cz/vismo5/dokumenty2.asp?u=450018&id_org=450018&id=529865&p1=86158

hodnocení – dobrá cesta k oblasti životního prostředí, potom zbytečné vyhledávání v dokumentech odboru

formát – XLS

atributy – IČ, provozovatel, sídlo provozovatele, typ zařízení, způsob nakládání, kód, kategorie odpadu, provoz/adresa provozovny, dotčené obce, příslušná ORP, platnost souhlasu, poznámky, číslo PZ

kraj Vysočina

cesta – <http://kr-vysocina.cz/> - Životní prostředí – Odpadové hospodářství – přehled povolených zařízení v kraji

adresa – http://www.kr-vysocina.cz/vismo/dokumenty2.asp?u=450008&id_org=450008&id=1186383&p1=0&p2=&p3=

hodnocení – dobrá dostupnost, ale necelistvý soubor. Data jsou rozdělena do několika souborů.

formát – DOC, XLS

atributy – kraj, název firmy, IČ, adresa firmy, adresa provozovny, kontaktní osoba, telefon, e-mail, typy přijímaných vozidel, povolení provozu zařízení vydané kým, platnost od, platnost do, Rozsah povolení, Upřesnění firmy, Činnost

Zlínský kraj

cesta – <http://www.kr-zlinsky.cz/> - Dokumenty – Dokumenty odborů krajského úřadu – Odbor životního prostředí a zemědělství – Zařízení ke sběru, výkupu, využití nebo odstranění odpadů

adresa – <http://www.kr-zlinsky.cz/docDetail.aspx?docid=29239&doctype=ART&nid=2753&cpi=1>

hodnocení – zbytečně zdlouhavé vyhledávání mezi všeobecnými dokumenty, mezi odkazy na životní prostředí je to bez zmínky

formát – XLS

atributy – název firmy, obec, adresa sídla, adresa provozovny (pokud se liší od sídla), osoba, telefon, typ zpracování, IČO, druh odpadu, platnost do

2.2 Analýza zdrojů v Evropě

V evropském měřítku se životním prostředím zabývá například Evropská agentura pro životní prostředí (European Environment Agency – EEA¹). Jedním z odvětví EEA je také sekce odpadového hospodářství. Ta vytvořila velmi zajímavou aplikaci² mapující firmy zabývající se odpadovým hospodářstvím. Je jí mapa vytvořená pomocí softwaru Macromedia Flash³ a technologie Google Earth⁴. Ke každé firmě na mapě je možno si nechat zobrazit podrobnosti jako je kategorie, do které spadá, klíčová slova, která se s ní vztahují a zdroj těchto dat. Nejvíce zmapovanými oblastmi jsou západní Evropa, zejména Německo, Itálie a Velká Británie.

Jako příklad systému konkrétní země můžeme vzít systém našeho jižního souseda, Rakouska. Na portálu Spolkového úřadu pro životní prostředí⁵ je umístěna databáze obsahující seznam provozovatelů zařízení zpracovávající odpady [8]. Lze vyhledávat podle oblasti nebo typu zařízení. Pro každý typ zařízení jsou evidovány specifické atributy (například pro skládku její plocha, pro šrotovací zařízení kapacita). Společnými atributy jsou pak například název provozovatele a kontakt.

Online-Abfrage zu abfallwirtschaftlichen Anlagen

Anlagentyp	Deponie
übernommene Abfallart	Baurestmassen
Standort	1210 Wien 21., Floridsdorf
	Wagramer Straße
Freies Deponievolumen	1.700.000 m3

Obrázek 2.1: Ukázka některých evidovaných dat v rakouském systému

1. <http://www.eea.europa.eu/>
2. http://technologies.ew.eea.europa.eu/atlas_map
3. <http://www.macromedia.com/software/flash/about/>
4. <http://earth.google.com/intl/cs/>
5. <http://www.umweltbundesamt.at/>

Kapitola 3

Použité nástroje

Pro jednotlivé fáze práce jsem potřeboval různý software. Při vytváření datového modelu jsem použil program CaseStudio¹. Pro samotnou realizaci pak balíček EasyPHP² (PHP³, MySQL⁴, Apache⁵), CakePHP⁶ a PSPad editor⁷.

3.1 Case Studio

Program od české firmy Charon Ware, které se podařilo prorazit s tímto kvalitním výrobkem do světa. Od roku 2007 se stala součástí Quest Software, která vývoj Case Studia ukončila a začala vydávat obdobný produkt pod jménem Toad Data Modeler. Case Studio se používá pro tvorbu diagramů a softwarové inženýrství všeobecně. Já jsem jej použil pro modelování tabulek databáze (verze 2.19.). Výhodou je, že po připravení diagramů lze vygenerovat SQL příkazy pro vytvoření tabulek v databázi.

3.2 EasyPHP

EasyPHP je balíček sdružující tři nejpoužívanější nástroje při tvorbě internetových stránek. Verze 1.8 obsahuje jazyk PHP (verze 4.3.10), databázový systém MySQL (4.1.9) a HTTP server Apache (1.3.33) a je volně ke stažení. PHP je skriptovací jazyk, jehož příkazy jsou prováděny na straně serveru. Je nezávislý na platformě a bez problémů pracuje na většině operačních systémů. Hlavní předností je přítomnost knihoven pro přístup k většině serverových databází (např. MySQL, MSSQL a další). Jak již bylo řečeno, MySQL je databázový systém od firmy MySQL AB a je nabízen ve dvou verzích, placené a volně stažitelné. Pro komunikaci s ním je používán jazyk SQL. Posledním z balíčku EasyPHP je http server Apache. Je to široce používaný, volně přístupný softwarový webový server. Dovoluje například konfiguraci zabezpečení, interní přepisování URL (přesměrování) či šifrovanou komunikaci.

Tento balíček jsem použil k testování na "localhostu", vlastním počítači. Jeho výhodou je především jednoduchá instalace, která spočívá v nainstalování všech tří částí najednou bez nutnosti nějaké další konfigurace.

1. <http://www.casestudio.com/csy/default.aspx>

2. <http://www.easyphp.org/>

3. <http://www.php.net/>

4. <http://www.mysql.com/>

5. <http://www.apache.org/>

6. <http://www.cakephp.org/>

7. <http://www.pspad.com/>

3.3 CakePHP

CakePHP není softwarem sám o sobě. Je to volně stažitelný framework⁸ zjednodušující práci s PHP svou architekturou postavenou na podobném principu jako Ruby on Rails⁹.

3.4 PSPad

PSPad je volně dostupný univerzální textový editor vyvinutý českým programátorem Janem Fialou. Jednou z jeho mnoha výhod je široká nabídka zvýrazňování syntaxe. Další například FTP klient nebo Lorem ipsum¹⁰ generátor.

8. Softwarová struktura sloužící jako podpora při programování. Může obsahovat například knihovny s funkcemi, které zjednoduší programování.

9. Framework pro programovací jazyk Ruby.

10. Označení pro standardní pseudolatinský text používaný jako demonstrativní výplň v grafickém designu.

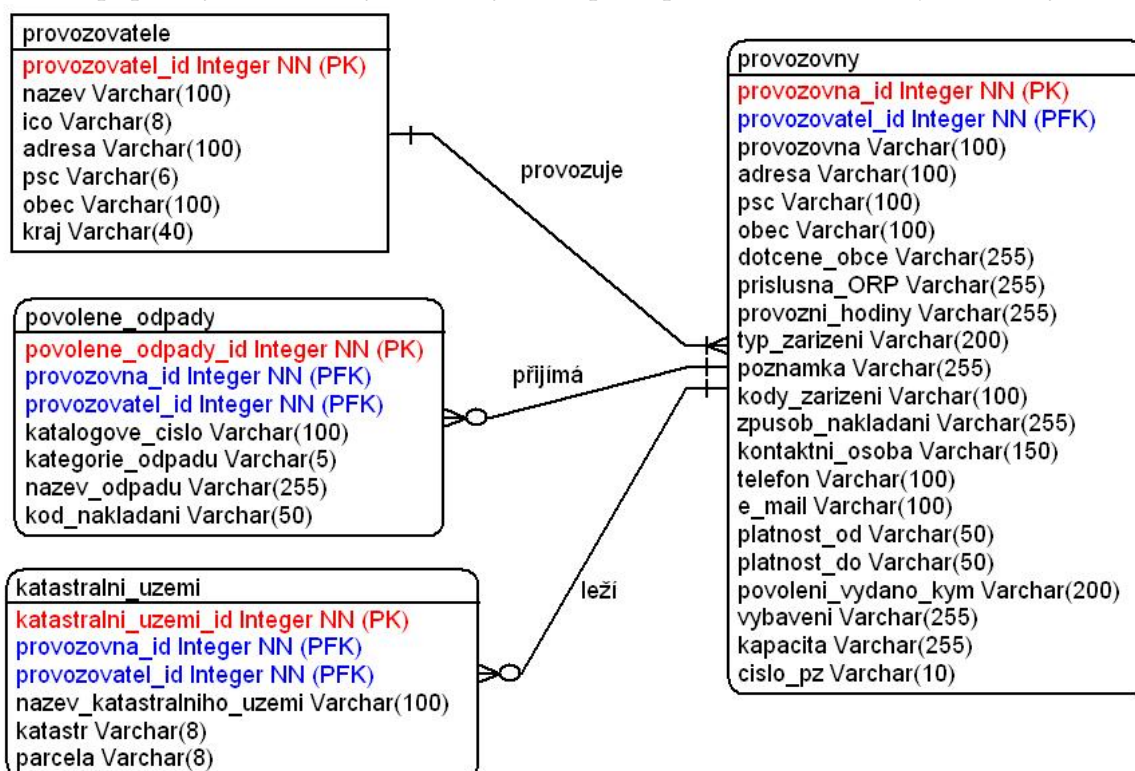
Kapitola 4

Datový model

4.1 Datový model

Nyní již můžeme přistoupit k návrhu a tvorbě databáze. Myslím, že v tomto případě lze vynechat postup podle některé z metod strukturované analýzy, jelikož se jedná z pohledu struktury o poměrně jednoduchý systém. Dalším důvodem je také fakt, že tento systém pouze poskytuje výpis z databáze, tudíž odpadá nějaká rozsáhlejší interakce ze strany uživatele. Ten pouze po jednoduchém nastavení hledaných atributů obdrží výsledek.

Pro popis výsledného systému by měl plně postačovat následující datový model:



Obrázek 4.1: Datový model systému

Ačkoliv správným postupem by bylo vytvořit datový model ještě dříve než vznikne výsledná databáze, v tomto případě se mi jevilo jako výhodnější tvoření výsledného datového modelu postupně podle jednotlivě zpracovávaných krajů. Každý kraj poskytuje jiná data, což znamená, že plní různé atributy.

Proto jsem postupně zpracovával jednotlivé kraje (více v kapitole 5) a vždy se snažil data právě zpracovávaného kraje "vtěsnat" do již vytvořených atributů v postupně vznikajícím datovém modelu. Prioritou tedy byla snaha dostat data všech krajů do co nejmenšího

množství atributů.

Prvním zpracovávaným krajem byl Jihočeský. Ten, podobně jako čtyři další (Liberecký, Olomoucký, Hlavní město Praha, Středočeský), má jednotný formát ISOHu, který poskytuje software od firmy INISOFT¹. A právě tento formát byl základním stavebním kamenem datového modelu na obrázku 4.1.

Oprávněná osoba	IČ	Ulice	PSČ	Obec	Provozovna	Ulice	PSČ	Obec	Katastrální území	Kódy zařízení	Povolené odpady	Platnost
"Recyklace odpadů a skládky a.s."	45021511	Písecká 1279	38601	Strakonice	"Recyklace odpadů a skládky a.s."	Písecká 1279	38601	Strakonice		R2	zobrazit odpady	neomezená
"Recyklace odpadů a skládky a.s."	45021511	Písecká 1279	38601	Strakonice	"Recyklace odpadů a skládky a.s."	Písecká 1279	38601	Strakonice		Sb,V	zobrazit odpady	neomezená
"Recyklace odpadů a skládky a.s."	45021511	Písecká 1279	38601	Strakonice	"Recyklace odpadů a skládky a.s."	Písecká 1279	38601	Strakonice		Z9	zobrazit odpady	neomezená
"Recyklace odpadů a skládky a.s."	45021511	Písecká 1279	38601	Strakonice	"Recyklace odpadů a skládky a.s."	Písecká 1279	38601	Strakonice		D9	zobrazit odpady	neomezená
"Recyklace odpadů a skládky a.s."	45021511	Písecká 1279	38601	Strakonice	"Recyklace odpadů a skládky a.s."	Písecká 1279	38601	Strakonice		Sb,V	zobrazit odpady	neomezená

Obrázek 4.2: Formát záznamu ISOHu od firmy Inisoft

Jeden záznam v tomto formátu má následující atributy:

Oprávněná osoba, IČ, Ulice, PSČ, Obec, Provozovna, Ulice, PSČ, Obec, Katastrální území, Kódy zařízení, Povolené odpady, Platnost

První čtyři atributy (*Oprávněná osoba, IČ, Ulice, PSČ*) se vztahují k provozovateli, tzn. ve většině případů právnické osobě, která provozuje jednu nebo více provozoven. Ty jsou charakterizovány zbylými atributy (*Ulice, PSČ, Obec, Katastrální území, Kódy zařízení, Povolené odpady, Platnost*). Proto jsem se rozhodl vytvořit dvě tabulky: *provozovatele* a *provozovny*. Které jsou vzájemně ve vztahu 1:n, kde provozovatelé mohou provozovat 1 až n provozoven (ne 0 provozoven, protože je zbytečné evidovat provozovatele, kteří neprovozují žádné provozovny). Zbylé dvě tabulky: *povolené odpady* a *katastrální území* jsem se rozhodl připojit k datovému modelu kvůli celkové stavbě ISOHu firmy INISOFT. Mezi atributy provozovny jsou sloupce *Katastrální území* a *Povolené odpady*. Tyto sloupce obsahují pouze odkazy na další stránky s tabulkami *Katastrální území* a *Povolené odpady*. Ty pak obsahují atributy uvedené v datovém modelu výše. Jedna provozovna může mít 0 nebo více údajů v tabulce *Katastrální území* a to samé platí pro tabulku *Povolené odpady*.

4.2 Výstavba databáze

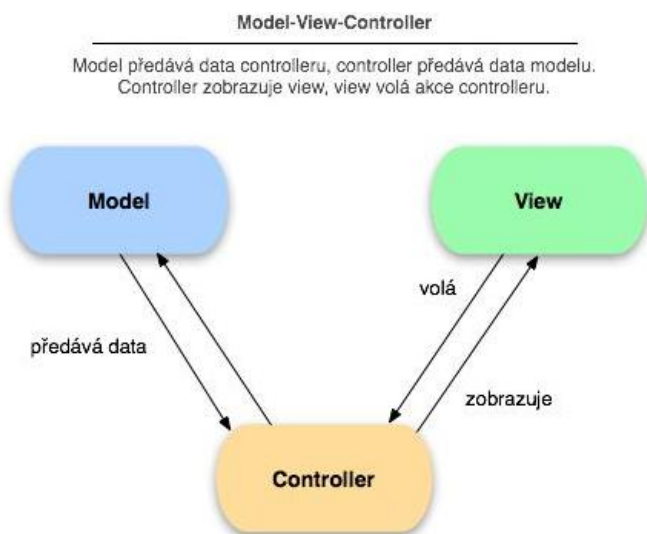
Dalším krokem bylo vytvoření databáze samotné pomocí nástrojů uvedených v kapitole 3 (Použité nástroje). Pro tvorbu datového modelu bylo použito CaseStudio, jehož pomocí jsem vygeneroval SQL příkazy pro vytvoření tabulek v databázi.

Tabulky jsou vytvořeny a nyní je třeba je napojit na webové rozhraní pomocí balíčku EasyPHP (viz. kapitola 3). Zde je velkým pomocníkem framework CakePHP. Pro tuto jednoduchou aplikaci by bylo využití takto robustní pomůcky zbytečné, nicméně za předpokladu, že bude tato databáze dále rozvíjena, tak lze říci, že CakePHP je investicí do budoucna.

1. <http://www.inisoft.cz/>

Tento framework využívá rozdělení na model, uživatelské rozhraní a řídicí logiku (tato objektová softwarová architektura se nazývá MVC - Model-view-controller). Model zajišťuje především komunikaci s databází, přesněji s jednotlivými tabulkami databáze (jeden model komunikuje s jednou tabulkou). Controller pak reaguje na události (přicházející z jednotlivých view) a volá model při potřebě komunikovat s databází. View zajišťuje rozhraní mezi systémovou logikou a uživatelem.

CakePHP dokáže usnadnit spoustu činností jako jsou jednodušší zápisy MySQL dotazů či nadefinování vztahů mezi jednotlivými tabulkami. To by ovšem, stejně jako spousta dalších pomůcek, nebylo možné, pokud by nebyly dodrženy určité konvence názvů.



Obrázek 4.3: Objektová struktura MVC

- Tabulky v databázi jsou pojmenovávány množným číslem za použití malých písmen. Pokud jsou názvy víceslovné, jsou odděleny podtržítkem.

provozovatele, provozovny, povolene_odpady, katastralni_uzemi

- Jméno objektu třídy model je jednotné číslo, pro víceslovný název jsou jednotlivá slova spojena dohromady a první písmeno každého z nich zvětšeno. Název souboru modelu je pojmenován jednotným číslem za použití malých písmen. Pokud jsou názvy víceslovné, jsou odděleny podtržítkem.

*Provozovatel (třída) – provozovatel.php (soubor), Provozovna – provozovna.php,
PovoleneOdpady – povolene_odpady.php, KatastralniUzemi – katastralni_uzemi.php*

- Jméno objektu třídy controller (ovladač) je množné číslo, pro víceslovný název jsou jednotlivá slova spojena dohromady a první písmeno každého z nich zvětšeno. Posledním slovem názvu je slovo Controller. Název souboru ovladače je pojmenován množným číslem, pokud jsou názvy víceslovné, jsou odděleny podtržítkem.

ProvozovateleController – provozovatele_controller.php

- View (pohledy) jsou pojmenovávány po jednotlivých událostech v ovladači. Jejich soubory jsou psány malými písmeny a mají příponu .html.

index.html, katastralni_uzemi.html, napoveda.html, o_webu.html, podrobnosti.html, povolene_odpady.html

Nyní se nabízí otázka, jak CakePHP pozná množné číslo například od slova provozovatel, když nejde o český framework. Odpověď je jednoduchá, vývojáři CakePHP mysleli i na toto a umožnili uživatelům z neanglicky mluvících zemí nadefinovat k jednotlivým slovům množná čísla.

To byly ve stručnosti konvence, které je třeba dodržet, aby byla využita co největší síla tohoto frameworku. Jak již bylo zmíněno výše, jednou z výhod CakePHP je možnost nadefinovat vztahy mezi tabulkami databáze na úrovni modelů. V našem případě je na místě

nadefinování vztahů mezi tabulkami *provozovatele* a *provozovny*. V modelu *provozovatel* využijeme proměnnou *\$hasMany*, která je typu pole a nese v sobě informace o definici vztahu, v našem případě s modelem *provozovny*. To samé nadefinujeme i v modelu *provozovny*, pouze s tím rozdílem, že použijeme proměnnou *\$belongsTo*. Nyní ještě upravíme MySQL kód (vygenerovaný pomocí programu CaseStudio), tak aby odpovídal konvencím CakePHP. Jedná se pouze o pojmenování klíčů. Primární klíče jsou pojmenovány pouze *id*. Cizí klíč v tabulce *provozovny* je pojmenován *provozovatel_id*. V tuto chvíli již spolu tabulky *provozovatele* a *provozovny* začínají spolupracovat a odměnou je například podstatné zjednodušení MySQL příkazu *SELECT*. Při výběru záznamů z tabulky *provozovatele* získáme ihned i záznamy z tabulky *provozovny* napojené na vybrané provozovatele. Výsledek příkazu *SELECT* je pak uložen do pole:

```
Array
(
    [Provozovatele] => Array
        (
            [id] =>1
            [nazev] => název provozovatele
            [ico] => 12345678
            [adresa] => adresa provozovatele
            ...
        )
    [Provozovny] => Array
        (
            [0] => Array
                (
                    [id] =>1
                    [provozovatele_id] => 1
                    [provozovna] => název provozovny č. 1
                    [adresa] => adresa provozovny č. 1
                    [psc] => 616 00
                    ...
                )
            [1] => Array
                (
                    [id] =>2
                    [provozovatele_id] => 1
                    [provozovna] => název provozovny č. 2
                    [adresa] => adresa provozovny č. 2
                    [psc] => 620 00
                    ...
                )
            [2] => Array
                (
                    [id] =>3
                    [provozovatele_id] => 1
                    [provozovna] => název provozovny č. 3
                    [adresa] => adresa provozovny č. 3
                    [psc] => 616 00
                    ...
                )
        )
)
```

Podle datového modelu by nyní bylo na místě definovat vztahy mezi tabulkami *provozovny* a *povolene_odpady* (resp. *katastralni_uzemi*). To by opět bylo možné na úrovni modelů, avšak považuji to za zbytečné, jelikož výpis záznamů z tabulek *povolene_odpady* a *katastralni_uzemi* probíhá v externím okně mimo od záznamů v tabulce *provozovny* (tzn. není třeba použít SQL dotaz obsahující přirozené či jiné spojení).

Posledním krokem před založením tabulek v databázi bude vytvoření indexů na všech cizích klíčích (na primárních je index vytvořen automaticky). Tím dosáhneme vyšší rychlosti zejména při zpracovávání příkazů `SELECT` při použití spojení (`JOIN`). Nabízí se otázka, proč tedy neoznačíme indexem všechny atributy. Při využití typu uložiště MyISAM se data separují od indexů (jsou uložena fyzicky jinde). Pokud bychom tedy označili indexem všechny atributy, docházelo by opět při každém dotazu k výběru dat ze všech sloupců, čímž by byla ztracena výhoda rychlého přístupu k některým sloupcům.

Nyní tabulky *provozovatele*, *provozovny*, *povolene_odpady* a *katastralni_uzemi* upravíme do výsledné podoby (zápis je zkrácen, vynechané atributy, označené pouze pomocí "...", zůstaly nezměněny od podoby datového modelu):

```
CREATE TABLE `provozovatele` (
  `id` int(10) unsigned NOT NULL auto_increment,
  `nazev` varchar(100) default NULL,
  ...
  PRIMARY KEY (`id`)
) ENGINE=MyISAM DEFAULT CHARSET=utf8;

CREATE TABLE `provozovny` (
  `id` int(10) unsigned NOT NULL auto_increment,
  `provozovna` varchar(100) default NULL,
  ...
  `provozovatel_id` int(10) unsigned NOT NULL default '0',
  PRIMARY KEY (`id`),
  INDEX (`provozovatel_id`)
) ENGINE=MyISAM DEFAULT CHARSET=utf8;

CREATE TABLE `povolene_odpady` (
  `id` int(10) unsigned NOT NULL auto_increment,
  `katalogove_cislo` varchar(100) default NULL,
  ...
  `provozovna_id` int(10) unsigned NOT NULL default '0',
  PRIMARY KEY (`id`),
  INDEX (`provozovna_id`)
) ENGINE=MyISAM DEFAULT CHARSET=utf8;

CREATE TABLE `katastralni_uzemi` (
  `id` int(10) unsigned NOT NULL auto_increment,
  `nazev_katastralniho_uzemi` varchar(100) default NULL,
  ...
  `provozovna_id` int(10) unsigned NOT NULL default '0',
  PRIMARY KEY (`id`),
  INDEX (`provozovna_id`)
) ENGINE=MyISAM DEFAULT CHARSET=utf8;
```

Kapitola 5

Naplnění databáze

Po vytvoření databáze podle datového modelu přišla na řadu druhá fáze, kterou bylo její naplnění. Při analýze zdrojů v kapitole 2 (Analýza zdrojů) je u každého kraje uvedeno, v jakém formátu je seznam provozovatelů zařízení zpracovávající odpady. Získávání dat se pak podle jednotlivých formátů lišilo. Na přiloženém CD je 14 souborů s příponou .php v hlavní složce Bakalářská práce, pro každý kraj jedna. Hlavním souborem pro zpracování dat je *phpparser.php*, ve kterém jsou pomocí funkce `include("název_souboru")`, která vloží do hlavního souboru svůj zdrojový kód, umístěny "zpracovávající algoritmy" pro jednotlivé kraje. Adresáře s dávkovými soubory krajů jsou pak v adresáři *kraje_data*.

5.1 Rozdělení podle formátů

5.1.1 HTML

HTML je zkratka anglického názvu HyperText Markup Language. Umožňuje vytváření a publikaci stránek na Internetu. Jde o značkovací jazyk, který pomocí speciálních značek umožňuje formátování textu.

Pro získávání dat z formátu HTML bylo třeba vytvořit "dávkové soubory". Nejdříve jsem uvažoval nad použitím některého z programů jako například WGET¹, pomocí kterého lze stahovat soubory nebo i celé stránky z internetu zadáním příkazů do konzole (v případě MS Windows, příkazové řádky). Nicméně jsem po prozkoumání zdrojů a zjištění, že se jedná pouze o desítky internetových stránek, přistoupil k manuálnímu uložení dat a WGET jsem nepoužil. Při manuálním ukládání dat jsem také měl lepší příležitost projít si strukturu dat, což pro mě v další práci, kterou bylo parsování textu, bylo nezbytné.

V tuto chvíli mám vytvořeny dávkové soubory dat krajů, které používají HTML formát. Jako další krok tedy je jejich rozparsování. Přesněji řečeno projít jejich zdrojové kódy a pomocí příkazů jazyka PHP jednotlivé atributy "uchopit". Všechny kraje využívající formát HTML měly podobnou strukturu. Data byla srovnána v tabulce. Sloupce určovaly atributy a řádky jednotlivé provozovatele či provozovny. Tím pádem tedy stačí pokaždé jen "rozpoznat" jednotlivé buňky tabulky a jejich obsah uložit.

Pro začátek by bylo na místě nastínit základy HTML [9]. Zejména pak značky používané pro strukturu tabulky:

Jednotlivé značky mohou být buď povinně párové, nepovinně párové nebo nepárové. Povinně a nepovinně párové značky mají ukončující značku stejnou jako je úvodní, pouze s uzavíracím lomítkem. Nepovinně párové ukončující značku přímo nevyžadují, ale používá se pro zpřehlednění zdrojového kódu. Pro nepárové značky se pak používá ukončovací lomítko, aby bylo zřejmé, že někde v budoucnu nenásleduje ukončovací značka (toto je použito v novějších specifikacích HTML, zejména pak ve specifikaci XHTML).

1. <http://www.gnu.org/software/wget/>

<table>, </table> – povinně párová značka. Určuje začátek a konec tabulky, ohraničuje možný výskyt dalších tabulkových značek

<tr>, </tr> – nepovinně párová značka. Určuje nový řádek tabulky. Koncová značka není vyžadována, nový řádek lze označit další značkou **<tr>**

<td>, </td> – nepovinně párová značka. Určuje novou buňku tabulky. Opět není koncová značka vyžadována, novou buňku lze označit další značkou **<td>**

<th>, </th> – nepovinně párová značka. Určuje novou buňku v záhlaví tabulky a svůj text zvýrazní a vycentruje. Ani zde není koncová značka vyžadována, novou buňku lze označit další značkou **<th>**

**
** – nepárová značka. Označuje zalomení řádku.

** ** – v HTML je každá sekvence bílých znaků chápána pouze jako jedna mezera. Proto se používá sekvence ** **; pro vyjádření takzvané pevné mezery. Je to nezalomitelná mezera, takže se dá použít mezi předložkou a slovem, aby nedošlo k nevhodnému zalomení řádku mezi nimi. Dalším použitím je také zobrazení několika mezer za sebou. Toho docílíme pouze zápisem několika značek ** **; za sebou.

Nyní již mohu přistoupit k samotnému získávání obsahu jednotlivých buněk v tabulkách zdrojových souborů. Měl jsem jasno, že použiji programovací jazyka PHP, nebyl jsem však ještě rozhodnut jaké funkce využiji. Jako první jsem přistoupil ke zpracování dat Jihomoravského kraje. Jednalo se o 14 souborů stejné struktury (*JMkraj_1-14davka.txt*). O některých funkcích jazyka PHP jsem neměl zpočátku tušení, a proto první pokus o parsování textu dopadl sice v pořádku, nicméně neefektivně.

```
1 for ($k = 1; $k <= 14; $k++)
2 {
3     $i = 0;
4     $filename = "../kraje_data/JMkraj/JMkraj_". $k. "davka.txt";
5     $file = fopen($filename, "r") or die("Nepodarilo se otevrit soubor pro cteni");
6     fseek($file, "3", SEEK_CUR);
7     while ((filesize($filename)-ftell($file))>2)
8     {
9         $retez = fread($file, 3) or die("Nepodarilo se nacist 3 znaky.");
10        if ($retez != "<td") fseek($file, "-2", SEEK_CUR); dalsi 3 znaky
11    else
12    {
13        while (fread($file, 1) != ">") {}
14        $odkud = ftell($file);
15        while (fread($file, 5) != "</td>") fseek($file, "-4", SEEK_CUR);
16        fseek($file, "-5", SEEK_CUR);
17        $kam = ftell($file);
18        fseek($file, $odkud-$kam, SEEK_CUR);
19        $nacti = fread($file, $kam-$odkud);
20        $nacti = str_replace("<br />", "", $nacti);
21        $nacti = str_replace("&nbsp;", "", $nacti);
22        $arrayofresults[$k][$i]=$nacti;
23        fseek($file, 5, SEEK_CUR);
24        $bezmezer = uriznimezery($nacti);
25        $arrayofresults[$k][$i] = $bezmezer;
```

```

26 $i++;
27     }
28     }
29     fclose ($file);
30     }

```

řádek 1-30 – cyklus, který prochází všech 14 souborů Jihomoravského kraje

řádek 4-5 – zadání a otevření souboru, který se bude zpracovávat

řádek 6 – v souboru se na začátku vyskytovaly 3 blíže nespecifikované znaky

řádek 7-28 – průchod přes všechny znaky souboru dokud není dosažen konec (resp. dokud zbývají k přečtení ještě alespoň 3 znaky)

řádek 10 – jestliže načtené 3 znaky nejsou `<td>`, pak se vrat' o 2 znaky zpět

řádek 12-27 – dokud není načteným znakem `>`, načítej další

řádek 14 – zjištění pozice ukazatele v otevřeném souboru odkud se bude načítat obsah buňky tabulky

řádek 15 – nalezení uzavírací značky `</td>`

řádek 16-17 – po načtení značky `</td>` je třeba se posunout na pozici posledního znaku, který je uvnitř buňky tabulky a uloží se jeho pozice

řádek 18-19 – návrat ukazatele v souboru na začátek buňky a načtení obsahu buňky

řádek 20-21 – odstranění všech značek `<br />` a ` ` v textu

řádek 22-26 – odstranění přebytečných mezer a uložení výsledku

řádek 29 – zavření souboru

Tento algoritmus tedy projde text a každý znak načte minimálně jednou. To je z hlediska úspory zdrojového kódu i efektivity velmi nepraktické. Proto bylo potřeba najít jiné řešení. To jsem aplikoval na data všech ostatních krajů. Řešením bylo použití funkcí využívajících regulární výrazy standardu POSIX (Portable Operating System Interface – přenositelné rozhraní pro operační systémy). Tyto funkce nejsou binárně bezpečné, což ale není problém, protože na souborech probíhá pouze čtení². Nyní je na místě přiblížit, co to vlastně regulární výrazy jsou, protože jsou podstatou funkcí, které jsem využíval.

Regulárním výrazem(RE) [10] je řetězec popisující množinu řetězců, které danému regulárnímu výrazu vyhovují. Skládá se z literálů a speciálních znaků pro počty výskytů, přepínače, množiny či různé alternativy. Regulární výrazy se například používají při vyhledávání textu, který může mít více variant nebo neznáme jeho přesné znění.

Stejně jako všechny pojmy matematiky či informatiky mají i regulární výrazy přesnou definici, tu však můžeme v tomto případě přeskočit a přistoupit k popisu praktického použití regulárních výrazů.

. – nahrazuje právě jeden znak

***** – kladně vyhodnotí 0 nebo více výskytů předcházejícího znaku. Příklad: `ab*c` vyhovuje pro `ac` i `abbbbc`

2. Binárně bezpečná funkce (anglicky binary-safe function) je taková funkce, která může použít binární soubor bez toho, aby změnila jeho obsah. To znamená, že binárně bezpečná funkce nemůže poškodit používaný soubor.

? – kladně vyhodnotí 0 nebo 1 výskyt předcházejícího znaku

+ – kladně vyhodnotí 1 nebo více výskytů předcházejícího znaku

[] – určuje množinu znaků, které budou kladně vyhodnoceny. Příklad: [abc] – kladně vyhodnotí a, b nebo c

[^] – doplněk předchozího, kladně jsou vyhodnoceny znaky, které neodpovídají obsahu závorek.

(a|b) – možnost výběru, kladně je vyhodnocen znak a i b

Existují také značky pro množinu znaků:

[:blank:] – označuje mezeru nebo tabulátor

[:space:] – označuje množinu znaků, které vytvářejí bílé místo. Tím může být například nový řádek nebo tabulátor.

Regulární výrazy mají samozřejmě mnohem širší množinu použitelných pravidel či označení. Výběr výše zmíněných pravidel zápisu jsem vytvořil s ohledem na jejich využití v této práci.

Jazyk PHP obsahuje množství funkcí využívajících regulární výrazy. Pro optimalizaci výše zmíněného algoritmu byly použitelné tyto:

Definice funkcí jsou psány stylem: typ výsledku-název funkce (parametry)

array spliti (string \$pattern, string \$string [, int \$limit])

Funkce vracející pole, v kterém jsou části textu \$string rozdělené na podřetězce podle vzoru \$pattern. Vzor \$pattern je regulárním výrazem. Třetí parametr je nepovinný a lze jím nastavit maximální velikost výsledného pole (resp. kolik částí textu \$string rozdělených pomocí \$pattern bude uloženo s tím, že v poslední části pole bude zbytek nerozděleného textu \$string). Nerozlišuje velikost písmen.

string eregi_replace (string \$pattern, string \$replacement, string \$string)

Argumenty této funkce jsou řetězec \$string ve kterém funkce hledá \$pattern, který je nahrazen parametrem \$replacement. Výsledek je nakonec vrácen jako řetězec. Nerozlišuje velikost písmen.

Pro převod dat z Olomouckého kraje jsem již použil optimalizovaný algoritmus:

```
1 $sloupce = array ('Oprávněná osoba', 'IČ', 'Ulice', 'PSČ', 'Obec',
2   'Provozovna', 'Ulice', 'PSČ', 'Obec', 'Katastrální území',
3   'Kódy zařízení', 'Povolené odpady');
4 for ($k = 1; $k <= 17; $k++)
5 {
6   $filename = "../kraje_data/Olomoucky_kraj/OK_". $k. "davka.txt";
7   $file = fopen($filename, "r")
8   or die ("Nepodarilo se otevrit soubor pro cteni");
9   $vse = fread ($file, filesize ($filename));
10  fclose ($file);
11  $vysledek[$k] =spliti('[:blank:]*(</td>[:space:])*?((</tr>)?
12   [:space:]*<tr[^\>]*>[:space:])*?<td[^\>]*>', $vse);
13 }
```

řádek 1 – pole s nadpisy jednotlivých sloupců – zadané manuálně

řádek 2-9 – cyklus, který prochází všech 17 souborů Olomouckého kraje

řádek 4-5 – zadání a otevření souboru, který se bude zpracovávat

řádek 6 – načtení celého, právě otevřeného souboru do proměnné `$vse`

řádek 7 – zavření souboru

řádek 8 – pomocí funkce `split` používající regulární výrazy se rozdělí řetězec v proměnné `$vse`

Celý tento algoritmus spočívá ve správném sestavení regulárního výrazu na řádce 8:

```
[[:blank:]]*(  
[[:space:]]*)?(  
[[:space:]]*  
<tr[^\>]*[[:space:]]*)?<td[^\>]*>
```

Základem bylo sestavení takového regulárního výrazu, který dokázal vyhledat tagy `<tr>` a `<td>` resp. `</tr>` a `</td>` a podle nich text v proměnné `$vse` rozložit.

Zdrojový kód pro data z Olomouckého kraje vypadají takto (ostatní kraje mají data uspořádána stejně nebo s menšími obměnami):

```
1 <tr class="even">
2 <td>" KOVOs Tovačov v.o.s. "</td>
3 <td>45197555</td>
4 <td>190</td>
5 <td>75102</td>
6 <td>Troubky</td>
7 <td class="dvojlinka">KOVOs Tovačov v.o.s.</td>
8 <td>Sladkovského</td>
9 <td>75102</td>
10 <td>Holice</td>
11 <td></td>
12 <td><span onmouseover="return overlib('S7-Sběrové místo',
    CAPTION, 'Kód zařízení');" onmouseout="return nd();">S7</span></td>
13 <td><span onmouseover="return overlib('zobrazit seznam povolených
    odpadů', CAPTION, 'zobrazit odpady');" onmouseout="return nd();">
    <a href='odpady.php?zuj=574988&pro=4&fir=45197555&kz=S7
    &page=1' onclick="return !katalog(this,600,600);">zobrazit odpady
    </a></span></td>
14 </tr>
```

V takovém zdrojovém kódu tedy vyhledáváme značku `<td>`, která uvozuje novou buňku tabulky. Jestliže nenačítáme první řádek tabulky, pak značku `<td>` mohou předcházet značky `</td>`, `</tr>`, `<tr>`. Regulární výraz akceptující tyto podmínky může vypadat takto:

```
(</td>)?(</tr>)?(<tr>)?<td>
```

Takový RE pak vyhledává značku `<td>`, která může být uvozena jednou nebo žádnou značkou `</td>`, `</tr>` nebo `<tr>`. Dále značky `<tr>` a `<td>` mohou obsahovat atributy před uzavírací závorkou např. `<tr class="even">`. Proto je potřeba náš RE doplnit:

```
(</td>)?(</tr>)?(<tr[^\>]*>)?<td[^\>]*>
```

Tento RE rozšiřuje předchozí, tím, že mezi `<tr>` resp. `<td>` se může vyskytovat 0 a více znaků různých než `>`. Lomítko je použito, aby znak `>` nebyl považován za systémový. Dalším krokem je ošetření mezer a bílých míst. Z ukázky zdrojového kódu je vidět, že se před značkou `</td>` může vyskytovat mezera a mezi jednotlivými dalšími značkami `</td></tr><tr>` i nový řádek. Proto opět doplníme náš RE o množiny `[[:blank:]]` a `[[:space:]]`:

```
[[:blank:]]*(</td>)?[[:space:]]*(</tr>)?[[:space:]]*
(<tr[^\>]*>)?[[:space:]]*<td[^\>]*>
```

Tento stav však není úplně konečný. Je třeba ještě upravit závorky, protože některá bílá místa se mohou vyskytovat pouze pokud se vyskytuje značka `</tr>` apod. . Proto upravíme RE do tohoto stavu:

```
[[:blank:]]*(</td>[[:space:]]*)?((</tr>)?[[:space:]]*
<tr[^\>]*>[[:space:]]*)?<td[^\>]*>
```

Nyní již máme regulární výraz použitelný pro funkci `split()`, Tento RE je v menších obměnách použit pro zpracování všech dat krajů, která byla ve formátu HTML.

5.1.2 XLS

Druhým nejpočetněji zastoupeným formátem dat byl XLS. Jedná se o tabulkový formát nejvíce proslavený aplikací Microsoft Excel, která je v dnešní době mezi uživateli pravděpodobně nejrozšířenější. Existují však i další možnosti, jakou je například kancelářský balík OpenOffice.org³, který je nabízen zdarma.

Zpracování tohoto formátu probíhalo ve dvou fázích. První bylo převedení XLS formátu do CSV, což je formát, který obsah jednotlivých buněk tabulky odděluje předem zvoleným znakem (tím může být čárka, mezera, apod.). V tomto případě jsem zvolil středník. Zde je příklad části CSV souboru pro Karlovarský kraj:

```
45809712;.A.S.A.;Chebská;Sokolov 1
45809712;.A.S.A., spol. s r.o.;Ďáblická 791/89;Praha 8
45809712;.A.S.A., spol. s r.o.;1.máje 752;Habartov
```

Dalším krokem tedy bylo zpracování formátu CSV obdobně jako HTML dat, tedy pomocí jazyka PHP. Opět jsem s úspěšností použil funkci `split()`, která tentokrát využívala jednodušší regulární výraz.

```
(;| /n)
```

Tento RE tedy jako oddělovač vyhodnotil středník nebo nový řádek.

5.1.3 DOC

Tato zkratka pod sebou skrývá název dokument, který zde jistě není třeba široce představovat. Jde o zkratku označující soubory textového editoru.

Získávání dat z toho formátu by se mohlo zdát na první pohled složitější než je. V dnešní době již textové editory mohou převádět dokumenty na soubory formátu HTML (opět například výše zmíněný kancelářský balík OpenOffice.org). Výsledný zdrojový HTML kód pak nevypadá příliš "učesaně", nicméně pro naše účely postačuje. Ten pak opět zpracujeme pomocí již známé PHP funkce `split()` a RE sestaveného v kapitole 5.1.1.

3. <http://www.openoffice.cz/>

5.1.4 PDF

Posledním zpracovávaným formátem je PDF⁴. Tento formát založený na jazyce PostScript zajišťuje, že PDF dokumenty jsou na všech zařízeních zobrazeny stejně. Stačí využít některý z mnoha PDF prohlížečů (např. Adobe Reader⁵).

Použití tohoto formátu se z hlediska dalšího zpracování jeví jako poněkud nešťastné. K získání dat je třeba využít některý OCR software⁶. Vyzkoušel jsem například TopOCR⁷ či Recognita⁸. Nicméně výsledek rozpoznávání byl tak špatný, že jej nebylo možno parsovat a data získat. Proto nejsou data Královéhradeckého kraje v databázi kompletní. Pravděpodobně jedinou možností k získání dat je kontaktovat Královéhradecký krajský úřad a data si vyžádat.

5.2 Rozdělení podle krajů

V této kapitole projdeme zpracování dat jednotlivých krajů.

Jihočeský kraj

formát – HTML

Data jsou v přehledném ISOHu vytvořeném firmou INISOFT. Použit je přesný postup z kapitoly 5.1.1. o zpracování HTML formátu. Jediným problémem bylo zpracování dat ze sloupců *Katastrální území* a *Povolené odpady*, kde jsou uloženy pouze odkazy na další podstránky. Proto jsem ukládal URL, která na výpis z katastru resp. seznam odpadů odkazovala a daný soubor následně rozparsoval a zpracoval.

Zpracování proběhlo obdobně jako pro všechna data v HTML formátu, tzn. za pomoci funkce `split` i a regulárních výrazů. Nejprve však bylo třeba uložit data do kódování UTF-8 (stránky s výpisy z katastru a seznamu odpadů využívaly kódování ISO-8859-2). Pro změnu kódování jsem úspěšně využil funkci `iconv`.

string iconv (string \$in_charset, string \$out_charset, string \$str)

Funkce vrací řetězec v kódování `$out_charset`, který vznikne z řetězce uloženého v proměnné `$str` v kódování `$in_charset`. Příklad: `iconv("ISO-8859-2", "UTF-8", 'řetězec')`

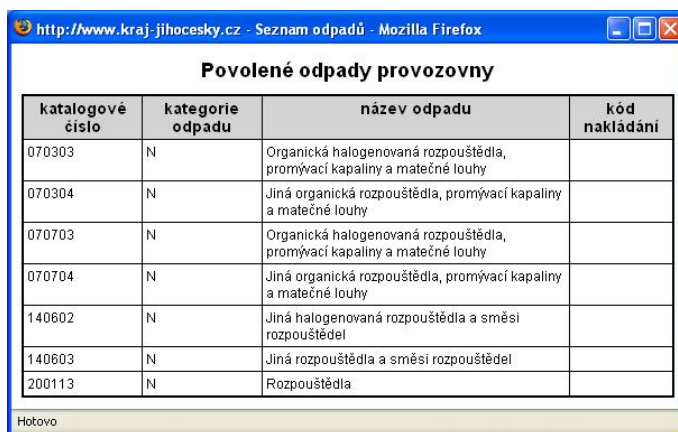
4. Portable document format = Formát pro přenositelné dokumenty

5. <http://www.adobe.com/cz/>

6. OCR - Optical character recognition - rozpoznávání znaků

7. <http://www.topocr.com/>

8. <http://www.recognita.com/>



katalogové číslo	kategorie odpadu	název odpadu	kód nakládání
070303	N	Organická halogenovaná rozpouštědla, promývací kapaliny a matečné louhy	
070304	N	Jiná organická rozpouštědla, promývací kapaliny a matečné louhy	
070703	N	Organická halogenovaná rozpouštědla, promývací kapaliny a matečné louhy	
070704	N	Jiná organická rozpouštědla, promývací kapaliny a matečné louhy	
140602	N	Jiná halogenovaná rozpouštědla a směsi rozpouštědel	
140603	N	Jiná rozpouštědla a směsi rozpouštědel	
200113	N	Rozpouštědla	

Obrázek 5.1: Zobrazení povolených odpadů

Po úpravě kódování již stačilo opět pomocí funkce `spliti` rozparsovat získaná data. Použitý regulární výraz byl tentokrát jednodušší, protože data byla uložena v tabulce bez dalších atributů u značek `<td>` resp. `<tr>`:

```
((</td></tr>)?<tr><td>|</td><td>|</td></tr>[[:blank:]]*
```

Jihomoravský kraj

formát – HTML

Přehledný ISOH jsem rozparsoval "neefektivním" způsobem popsaným v kapitole 5.1.1.

Karlovarský kraj

formát – XLS, HTML

Data byla uložena v několika souborech. Zpracovatelé autovraků v HTML formátu, který jsem zpracoval téměř obvyklým způsobem. Pouze jsem navíc využil funkci `eregi_replace` pro nahrazení nadbytečných formátovacích HTML značek za prázdný řetězec. XLS formát byl zpracován podle postupu v kapitole 5.1.2.

Královéhradecký kraj

formát – PDF

Jediný kraj, který využíval formát PDF. Data se mi pomocí OCR softwaru nepodařilo získat.

Liberecký kraj

formát – HTML

Liberecký kraj využíval již výše zmíněný ISOH firmy Inisoft, proto byl postup zpracování stejný jako u Jihočeského kraje. Stránky s tabulkami povolených odpadů a katastrálních území opět využívala kódování ISO-8859-2, a proto byla nutná konverze do UTF-8.

Moravskoslezský kraj

formát – HTML

Zpracování proběhlo využitím postupu v kapitole 5.1.1. pomocí PHP funkce `spliti`.

Olomoucký kraj

formát – HTML

Další z krajů využívajících služby ISOHu firmy Inisoft. Postup zpracování popsán u Jihočeského kraje. Tabulky povolených odpadů a katastrálních území bylo třeba převést z kódování ISO-8859-2 do UTF-8.

Pardubický kraj

formát – XLS

Zpracováno podle postupu 5.1.2. Jednalo se o 3 dávky.

Plzeňský kraj

formát – DOC

Jeden ze dvou krajů mající data ve formát DOC. Zpracoval jsem je podle postupu v kapitole 5.1.3. Menším, avšak řešitelným problémem, bylo několik oddělených sekcí v datových souborech.

Hlavní město Praha

formát – HTML

Opět využití systému firmy Inisoft. Postup zpracování obdobný jako u Jihočeského kraje. Zde byla třeba konverze z kódování Windows-1250 do UTF-8.

Středočeský kraj

formát – HTML

Středočeský kraj také využívá ISOH od Inisoftu. Viz. Jihočeský kraj včetně konverze kódování.

Ústecký kraj

formát – XLS

Data jsem rozparsoval podle postupu popsaného v kapitole 5.1.2.

Kraj Vysočina

formát – DOC, XLS roztroušené do více souborů

Kombinace postupů v kapitolách 5.1.2. a 5.1.3.

Zlínský kraj

formát – XLS

Jeden soubor formátu XLS rozdělený do několika sekcí. Využití postupu v kapitole 5.1.2. s menšími úpravami kvůli výše zmíněnému rozdělení.

5.3 Vkládání získaných dat

V této chvíli jsou připraveny jak tabulky databáze, tak i data, která se do tabulek budou vkládat. Proto je třeba doposud napsané PHP skripty doplnit o vkládání do databáze. Pro jednotlivé kraje se může postup mírně lišit, nicméně hlavní kostra zůstává stejná.

Jelikož potřebujeme provázat záznamy v tabulkách *provozovatele* a *provozovny* a také *provozovny* a *povolene_odpady* resp. *katastralni_uzemi*, musíme dbát na správné vkládání dat do sloupců označených jako cizí klíče. Také není třeba vkládat provozovatele, které provozuje více provozoven do tabulky *provozovatele* vícekrát, ale stačí jednou.

Proto zápis do databáze pro Jihočeský kraj vypadá takto:

```

1  if ($i+12<=sizeof($vysledek[$k]))
2  {
3      $result = mysql_query("SELECT id FROM provozovatele WHERE
        ico='".$vysledek[$k][$i+1]."' AND kraj = 'Jihočeský'")
        or die('Spatny dotaz - 1');
4      if (mysql_num_rows($result)==0)
5      {
6          mysql_query("INSERT INTO provozovatele(nazev,ico,adresa,psc,obec,
            kraj) VALUES('".$vysledek[$k][$i]."', '".$vysledek[$k][$i+1]."',
            '".$vysledek[$k][$i+2]."', '".$vysledek[$k][$i+3]."',
            '".$vysledek[$k][$i+4]."', 'Jihočeský'")
            or die('Spatny dotaz - 2');
7          $idecko = mysql_insert_id();
8      }
9      else //pokud jiz je ulozen dany provozovatel
10     {
11         $provozovatel = mysql_fetch_array($result, MYSQL_ASSOC);
12         $idecko = $provozovatel['id'];
13     }
14     mysql_free_result($result);
15     $vysledek[$k][$i+5] = ereg_replace('""', '', $vysledek[$k][$i+5]);
16     $vysledek[$k][$i+10] = ereg_replace('(<span[^\>]*>|</span>)', '',
        $vysledek[$k][$i+10]);
17     mysql_query("INSERT INTO provozovny(provozovna,adresa,psc,obec,
        kody_zarizeni,platnost_do,provozovatel_id) VALUES(
            '".$vysledek[$k][$i+5]."', '".$vysledek[$k][$i+6]."',
            '".$vysledek[$k][$i+7]."', '".$vysledek[$k][$i+8]."',
            '".$vysledek[$k][$i+10]."', '".$vysledek[$k][$i+12]."',
            '".$idecko."'") or die('Spatny dotaz - 3');
18     $provozovna_idecko = mysql_insert_id();
19     echo 'Záznam byl uložen. <br />';
        //posun v poli na dalsi ctenou polozku
20     $i=$i+9;$j=9;
21 }

```

pozn.: v poli `$vysledek[$k]` jsou uložena data z Jihočeského kraje zpracovaná postupem v sekci 5.1.1., v 9. sloupci se nachází odkaz na katastrální území a v 11. na povolené odpady.

řádek 1-21 – záznamy v tabulce Jihočeského kraje mají 13 sloupců proto, pokud existuje ještě 13 buněk v poli s výsledkem, tak jej zpracuj

řádek 3 – SQL dotazem zjistíme, zda-li je již provozovatel s daným *id* vložen

řádek 4-8 – pokud provozovatel s daným ičo ještě není vložen do tabulky *provozovatele*, tak jej vytvoř a zapamatuj si jeho *id*

řádek 9-13 – pokud již byl provozovatel s daným ičo vytvořen, tak zjisti jeho *id*

řádek 15-16 – úprava formátu vkládaných dat

řádek 17 – vložení dat do tabulky *provozovny*, posledním vkládaným atributem je cizí klíč *provozovatel_id* zjištěný na řádcích 4-8 nebo 9-13

řádek 18 – zjištění *id* posledního vloženého záznamu

řádek 19 – výpis na obrazovku o úspěšném vložení

řádek 20 – posun v načítaném poli výsledků přes již načtené buňky

```

1  if ($j==9) //pokud ukládáme data do tabulky katastralni_uzemi
2  {
3      for ($q = 0; $q <= count($zaznamy)-3; $q++)
4      {
5          mysql_query("INSERT INTO katastralni_uzemi (
            nazev_katastralniho_uzemi,katastr,parcela,provozovna_id) VALUES
            ('".$zaznamy[$q]."', '".$zaznamy[$q+1]."', '".$zaznamy[$q+2]."',
            '".$provozovna_idecko."')") or die('Spatny dotaz -
            insert - katastralni_uzemi');
6          $q = $q+2;
7      }
8  }
9  if ($j==11) //pokud ukládáme data do tabulky povolene_odpady
10 {
11     for ($q = 0; $q <= count($zaznamy)-4; $q++)
12     {
13         mysql_query("INSERT INTO povolene_odpady(katalogove_cislo,
            kategorie_odpadu,nazev_odpadu,kod_nakladani,provozovna_id) VALUES (
            '".$zaznamy[$q]."', '".$zaznamy[$q+1]."', '".$zaznamy[$q+2]."',
            '".$zaznamy[$q+3]."', '".$provozovna_idecko."')")
            or die('Spatny dotaz - insert - povolene_odpady');
14     $q = $q+3;
15     }
16 }
```

pozn.: v poli *\$zaznamy* jsou uložena data z tabulek katastrální území nebo povolené odpady řádek 1-8: pokud máme načtená data z 9. sloupce (katastrální území), tak je ulož

řádek 3-7 – pokud ještě existují 3 buňky v získaných záznamech (*\$zaznamy*), tak je ulož

řádek 5 – do tabulky *katastralni_uzemi* vlož záznam, posledním vkládaným atributem je cizí klíč *provozovna_id* zjištěný v předchozím fragmentu kódu (řádek 18)

řádek 9-16 – pokud máme načtená data z 11. sloupce (povolené odpady), tak je ulož

řádek 11-15 – pokud ještě existují 4 buňky v získaných záznamech (*\$zaznamy*), tak je ulož

řádek 13 – do tabulky *povolene_odpady* vlož záznam, posledním vkládaným atributem je cizí klíč *provozovna_id* zjištěný v předchozím fragmentu kódu (řádek 18)

Obdobným způsobem došlo k vložení dat i u ostatních krajů.

Kapitola 6

Tvorba webového rozhraní

6.1 Standardy ISVS

Jelikož zveřejňování seznamů zařízení pro zpracování odpadů je dáno zákonem (viz. kapitola 2), tak i webové rozhraní státních informačních systémů podléhá některým standardům. Nejprve bych zmínil 2 poměrně důležité pojmy:

- **Informační systém** – funkční celek nebo jeho část zabezpečující cílevědomou a systematickou informační činnost. Každý informační systém zahrnuje data, která jsou uspořádána tak, aby bylo možné jejich zpracování a zpřístupnění, a dále nástroje umožňující výkon informačních činností [11].
- **Informační systémy veřejné správy (ISVS)** – jsou souborem informačních systémů, které slouží pro výkon veřejné správy. Jsou jimi i informační systémy zajišťující činnosti podle zvláštních zákonů (o státní statistické službě, živnostenský zákon, o veřejném zdravotním pojištění, obchodní zákoník, o správě daní a poplatků) [11].

Provoz ISVS je upravován zákonem č. 365/2000 Sb., o informačních systémech veřejné správy. Toto je ve stručnosti jeho význam:

Zákon o informačních systémech veřejné správy stanoví práva a povinnosti správců informačních systémů veřejné správy (ISVS) a dalších subjektů, jež souvisejí s vytvářením, užíváním, provozem a rozvojem informačních systémů veřejné správy. V návaznosti na to upravuje působnost Ministerstva informatiky (po zániku ministerstva informatiky přechází správa o ISVS na ministerstvo vnitra) jako ústředního správního úřadu pro tvorbu a rozvoj informačních systémů veřejné správy. Zákon vytváří podmínky, aby kvalitní informační systémy byly dobrým nástrojem pro výkon veřejné správy.

Zákon dále mj. upravuje atestace ISVS a postavení atestačních středisek, doručování zpráv orgánům veřejné moci prostřednictvím portálu veřejné správy a poskytování ověřených výstupů z ISVS [11].

V roce 2006 byla vydána novela tohoto zákona. Její stručný význam by se dal charakterizovat takto:

Hlavním principem novely zákona je zpřesnit práva a povinnosti osob a dalších subjektů související s řízením informačních systémů veřejné správy a upravit v návaznosti na to působnost Ministerstva informatiky (dnes Ministerstva vnitra) jako ústředního správního úřadu pro tvorbu a rozvoj informačních systémů veřejné správy. Cílem novely zákona je vytvoření podmínek pro zajištění kvalitních dat veřejné správy a bezpečné technologické výměny informací za předem stanovených podmínek. Pravidla řízení informačních systémů veřejné správy, která byla původně pokrytá vydáváním standardů, jsou nově zabezpečena vydáním prováděcích právních předpisů (vyhlášek) a metodických pokynů.

Mezi hlavní zásady uplatňované v jednotlivých fázích životního cyklu informačních systémů patří vedle funkčnosti zejména ekonomická efektivita a zásady řízení bezpečnosti

těchto systémů. Novela zavádí nově akreditaci atestačních středisek. Akreditace bude zaručovat vysokou kvalifikaci a garantovanou způsobilost atestačních středisek pro výkon atestací. Povinnost atestací informačních systémů veřejné správy se zužuje pouze na dlouhodobé řízení informačních systémů a na způsobilost realizace vazeb mezi informačními systémy. Zákon nově zavádí povinnost přizpůsobit webové stránky institucí státní správy a samosprávy tak, aby byly přístupné i pro osoby zdravotně postižené. Další novinkou je možnost dodání datové zprávy orgánu veřejné moci prostřednictvím portálu veřejné správy, jakožto centrálního místa elektronické komunikace veřejné správy s občanem. Zákon zavádí institut vydávání ověřených výstupů z informačních systémů veřejné správy. Zavedení tohoto institutu si vyžádal rozvoj informačních a komunikačních technologií umožňujících přístup k datům elektronickými prostředky na dálku. Jedná se o ověřený listinný výstup z elektronicky vedené evidence, rejstříku nebo seznamu. Ověřené výstupy budou moci vydávat notáři, stanovené obce, Hospodářská komora ČR a držitel poštovní licence (Česká pošta, s.p.).

Novelizovaný zákon má rozdělenou účinnost. Obecně nabývá účinnosti dnem 1. ledna 2007, kromě několika výjimek. Vydávání ověřených výstupů z elektronicky vedené evidence, rejstříku nebo seznamu ověřujícími osobami nabude účinnosti dnem vyhlášení zákona ve Sbírce zákonů. Od stejného data nabývá účinnosti možnost dodání datové zprávy orgánu veřejné moci prostřednictvím portálu veřejné správy. Povinnost přizpůsobit webové stránky institucí státní správy a samosprávy tak, aby byly přístupné i pro osoby zdravotně postižené nabývá účinnosti dne 1. ledna 2008 [11].

Jelikož mnou vytvořený ISVS se nebude snažit získat atestaci, tak jsem vybral pouze některá pravidla¹, kterých by se měl držet²:

- webové rozhraní by mělo odpovídat pravidlům "blind-friendly webu"³, tzn. webu, čitelnému i lidem s určitým zrakovým postižením
- pracovat podle pravidel konsorcia W3C⁴, tak aby stránky byly W3C validní
- aby ovládání webového rozhraní bylo co nejjednodušší

6.2 Tvorba webového rozhraní

Jelikož jsem využil framework CakePHP (viz. kapitola 4.2.), bylo pro webové rozhraní třeba vytvořit tzv. pohledy (view). Ty jsou jinak řečeno obrazovkami, které od ovladače (controlleru) získají data (logiku) a zobrazí uživateli to, co si přeje vidět. Webové rozhraní se skládá z 6 pohledů – vyhledávání (*index.thtml*), nápověda (*napoveda.thtml*), o webu (*o_webu.thtml*), podrobnosti (*podrobnosti.thtml*), povolené odpady (*povolene_odpady.thtml*) a katastrální území (*katastralni_uzemi.thtml*).

6.2.1 Vyhledávání

Pro všechny hledané atributy (vyjma kraje) platí následující pravidla:

1. <http://www.isvs.cz/isvs-teorie/legislativa-a-standardy.html>
2. <http://www.pravidla-pristupnosti.cz/>
3. <http://www.blindfriendly.cz/>
4. <http://www.w3.org/>

- Jednotlivé vyhledávané řetězce je třeba oddělovat čárkou, jelikož je obsah každého vyhledávaného atributu rozložen na řetězce pomocí funkce `explode` s využitím čárky jako separátoru (funguje obdobně jako výše zmíněná funkce `split`, až na to, že nepoužívá jako separátory regulární výrazy). Pro každý takto získaný podřetězec je prověřováno, jestli není součástí některého záznamu v databázi v daném atributu (pro SQL dotaz to znamená, že v klauzuli `WHERE` se dotazujeme na řetězec `%abc%` místo `abc`).
- Mezery na začátku a na konci zadaných řetězců jsou zanedbávány. Odstranění těchto přebytečných mezer probíhá opět pomocí funkce `explode` tentokrát však s využitím mezery jako separátoru, a pokud je první či poslední získaný podřetězec délky 0, odstraníme první či poslední znak.
- Prázdné řetězce nebo řetězce skládající se pouze z mezer jsou také zanedbávány.

Databáze zařízení pro nakládání s odpady

Vyhledávání: vyhledávání | [nápověda](#) | [o webu](#)

Kraj: (lze vybrat i více krajů zároveň při stisknutí tlačítka CTRL)

- Jihočeský
- Jihomoravský
- Karlovarský
- Královéhradecký
- Liberecký
- Moravskoslezský
- Olomoucký
- Pardubický
- Plzeňský
- Hlavní město Praha
- Středočeský
- Ústecký
- Vysočina
- Zlínský

U všech atributů odděluje jednotlivé hledané výrazy čárkou (např. Brno,Kounicova).

Název provozovatele: (lze zadat i pouze část názvu)

Adresa provozovny: (lze zadat i pouze část adresy - např. ulici nebo město)

Kódy zařízení: (pro nápovědu použijte [katalog způsobu využívání odpadů](#))

Povolené odpady: (pro nápovědu použijte [katalog druhů odpadů](#))

kód:

kategorie:

název:

Obrázek 6.1: Vyhledávací formulář na hlavní stránce

Kraj

Výběr kraje v kterém se hledaná zařízení mají nacházet. Hledání proběhne v sloupci *kraj* tabulky *provozovatele*. Pokud není vybrán žádný kraj, vyhledávání neprobíhá. Lze vybrat několik krajů zároveň tažením myši při stisknutí levém tlačítku. Pokud výběr krajů není souvislý, lze jednotlivé kraje označit myší při stisknutí klávese CTRL.

Název provozovatele

Zadaná slova (lze zadat i části slov) budou vyhledávána v tabulce *provozovatele* v sloupci *název*.

Adresa provozovny

Zadaná slova (lze zadat i části slov) budou vyhledávána v tabulce *provozovny* a jejich sloupcích *obec* a *adresa*.

Kódy zařízení

Budou vypsány záznamy s odpovídajícími kódy zařízení. Hledání proběhne v sloupci *kody_zarizeni* tabulky *provozovny*. Kódy musejí být zadány přesně, jinak nebude hledání úspěšné. Jako pomůcka může sloužit odkaz na *katalog způsobu využívání odpadů*.

Povolené odpady - kód

Budou vypsány záznamy s odpovídajícími kódy povolených odpadů. Hledání proběhne ve sloupci *katalogove_cislo* tabulky *povolene_odpady*. Kódy musejí být zadány přesně, jinak nebude hledání úspěšné. Jako pomůcka může sloužit odkaz na *katalog druhů odpadů*.

Povolené odpady - kategorie

Budou vypsány záznamy s odpovídající kategorií povolených odpadů. Hledání proběhne ve sloupci *kategorie_odpadu* tabulky *povolene_odpady*. Kategorie musejí být zadány přesně, jinak nebude hledání úspěšné. Jako pomůcka může sloužit odkaz na *katalog druhů odpadů*.

Povolené odpady - název

Zadaná slova (lze zadat i části slov) budou vyhledávána v tabulce *povolene_odpady* ve sloupci *nazev_odpadu*. Pro nápovědu lze využít odkaz na *katalog druhů odpadů*.

6.2.2 Nápověda

Obrazovka Nápověda obsahuje detailní popis pro uživatele, jak správně zadat vyhledávané řetězce do pole daného atributu. Obsahuje také nefunkční vyhledávací formulář, aby si uživatel mohl jednotlivé vyhledávané atributy přiřadit k nápovědě.

6.2.3 O webu

Tato obrazovka obsahuje obecné informace o důvodu vzniku tohoto informačního systému a jeho využití.

6.2.4 Podrobnosti

Na obrazovku *Podrobnosti* se lze dostat odkazem z obrazovky *Vyhledávání*. Každý vyhledaný záznam obsahuje odkaz na výpis podrobností. Zobrazí uživateli o daném záznamu další informace.

6.2.5 Povolené odpady

Stejně jako na obrazovku *Podrobnosti*, tak i na tuto obrazovku se lze dostat odkazem z obrazovky *vyhledávání*. Každý vyhledaný záznam obsahuje odkaz na výpis z tabulky *povolene_odpady*. Povolené odpady pro daný záznam mohou, ale nemusí být uvedeny.

6.2.6 Katastrální území

Tato obrazovka funguje na stejném principu jako předchozí. Opět je u každého vyhledaného záznamu odkaz na výpis z tabulky *katastralni_uzemi*. Záznamy o katastru mohou, ale nemusí být uvedeny.

Kapitola 7

Další využití a možnosti

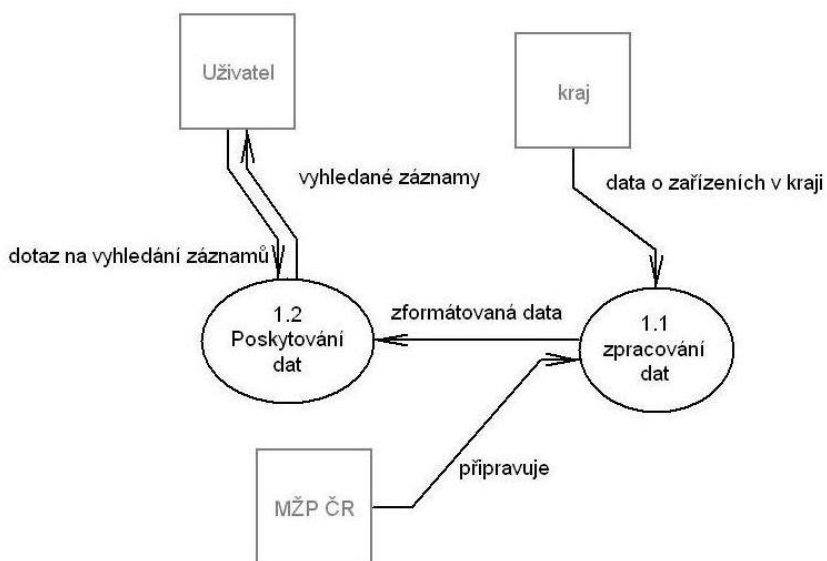
Výsledný stav této databáze by neměl být konečným stavem ISVS o zařízeních zpracovávajících odpady. V tuto chvíli systém eviduje data poskytovaná jednotlivými kraji a umožňuje v nich vyhledávání. Velkým problémem při tvorbě tohoto informačního systému byla nejednotnost dat poskytovaných jednotlivými kraji. Každý kraj poskytoval pouze atributy, které uznal za vhodné a v různých formátech. Některé kraje pouze vydaly jakýsi neformální seznam zařízení. Proto se může stát, že data obsažena v databázi jsou někdy uložena v nesprávném sloupci, ať už chybou mou či chybou vydavatele daného seznamu. Data by šla pravděpodobně lépe parsovat, nicméně to by již hraničilo s manuální prací bez využití nějakého algoritmu, protože jednotlivá poskytovaná data se od sebe v některých případech opravdu liší.

To samozřejmě není vina krajských úředníků. Řešení tohoto problému by muselo přijít z vyšších než krajských míst. To znamená z Ministerstva životního prostředí (MŽP¹). Pokud by byl MŽP vydán předpis o povinnosti evidovat určitá data o zařízeních zpracovávajících odpady, tak by byl vytvořen jakýsi obsahový i formátový standard. Zpracování dat od jednotlivých krajů podléhajících tomuto standardu by pak poskytovalo mnohem větší možnosti vyhledávání a zobrazování informací.

7.1 Aktivní zapojení MŽP ČR

V tomto případě by MŽP vytvořilo a udržovalo informační systém. Kraje by data podle ministerstvem vytvořeného standardu posílaly MŽP, které by je kontrolovalo a poté umístilo do systému. Výhodou tohoto systému by byla možnost kontroly dat od jednotlivých krajů ministerskými úředníky. Na druhou stranu by nevýhodou bylo zbytečné posílání dat do systému přes MŽP (což znamená zbytečnou byrokracii navíc). Obrázek 7.1 představuje diagram datových toků na 0. úrovni dekompozice.

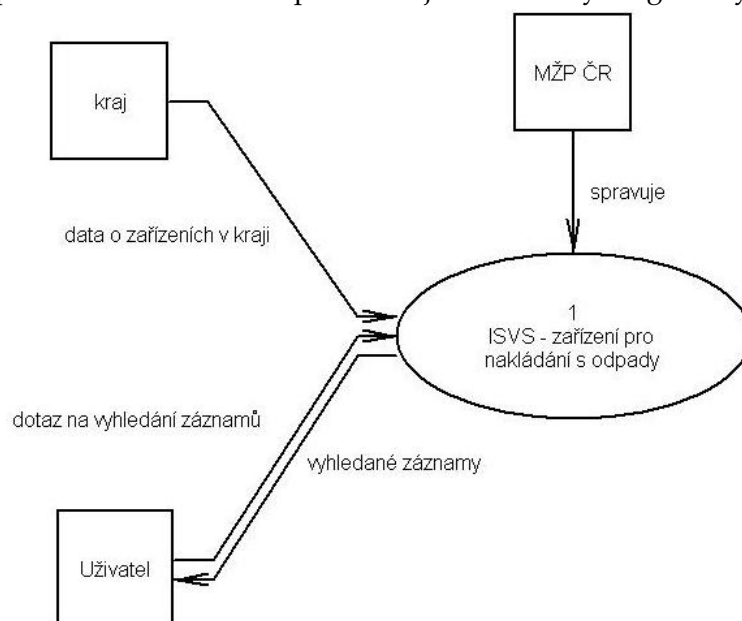
1. <http://www.env.cz/>



Obrázek 7.1: Diagram znázorňující aktivní zapojení MŽP

7.2 Pasivní zapojení MŽP ČR

Tento přístup spočívá v jednodušší účasti MŽP formou správy systému. Tím pádem by MŽP pouze zajistilo vznik informačního systému a údržbu jeho chodu. Data by pak jednotlivé kraje přes autentizovaný přístup posílala přímo do systému. Opět podle MŽP předem vytvořeného standardu. Výhodou by byl rychlejší a přímý přístup krajů do systému a MŽP by zde plnil pouze funkci supervizora. Obrázek 7.2 představuje kontextový diagram systému.



Obrázek 7.2: Diagram znázorňující pasivní zapojení MŽP

7.3 Další návrhy

Po standardizaci ukládaných dat již bude možné zapojit do systému další vylepšující funkce. Se standardizací by přišlo navázání jednotlivých atributů na číselníky. Těmi důležitými by byly určité ty, které identifikují polohu na mapě (tzn. číselníky obcí, zejména pak katastrálních území). Pomocí nichž by již nebylo technicky neřešitelné zobrazování jednotlivých firem a jejich provozoven na mapě například pomocí serveru mapy.cz².

Dalším vylepšením by mohlo být vyhledávání v Administrativním registru ekonomických subjektů (ARES). Kliknutím na IČO dané firmy by se zobrazil výpis o firmě z ARES či jiného obdobného registru.

Uživatelé by dále pravděpodobně ocenili výpis dat v různých formátech. Například CSV či XML.

2. <http://www.mapy.cz>

Kapitola 8

Závěr

Cílem práce bylo vytvořit systém mapující zařízení s povolením nakládat s odpady. Povolení provozovatelům vydávají jednotlivé kraje podle zákona § 14 odst. 1 zákona č. 185/2001 Sb. Proto samotné práci na systému předcházela analýza seznamů vydaných jednotlivými kraji.

Během práce jsem se blíže seznámil s legislativou a internetovými portály zabývajícími se odpadovým hospodářstvím. Při samotné tvorbě systému pak s možnostmi programovacího jazyka PHP, zejména využitím funkcí pro práci s řetězcí a regulárními výrazy. Dále také databází MySQL, značkovacím jazykem HTML a kaskádovými styly CSS. Velkou zkušeností a kladem bylo také využití frameworku CakePHP.

Jak jsem již zmínil v kapitole 7 (Další možnosti a využití), tak vytvořený systém v této podobě mapuje jednotlivá zařízení, avšak bez standardizace kraji poskytovaných údajů ze strany Ministerstva životního prostředí nelze tento systém dále vylepšovat.

Výsledný systém je vystaven na adrese <http://bakule.uk.fme.vutbr.cz/>.

Literatura

- [1] *Globální oteplování - Wikipedie, otevřená encyklopedie* [online].
Stránka byla naposledy editována v 19:15, 15. 12. 2007 [cit. 2007-12-20]. Dostupný z WWW: http://cs.wikipedia.org/wiki/Globální_oteplování.
- [2] *Jak Třídít — MÁ TO SMYSL. TŘÍDĚTE ODPAD* [online].
Stránka byla naposledy editována v roce 2006 [cit. 2007-12-20]. Dostupný z WWW: <http://www.jaktridit.cz/odpady/index.html>.
- [3] *Evropský parlament - Zpravodajství - Aktuality - Článek - Jak se nepohrřbít pod horami odpadu* [online].
Stránka byla naposledy editována v 12:02, 12. 2. 2007 [cit. 2007-12-20]. Dostupný z WWW: http://www.europarl.europa.eu/news/public/story_page/064-2977-043-02-07-911-20070209ST002961-2007-12-02-2007/default_cs.htm.
- [4] *Zákon č. 185/2001 Sb., o odpadech* [online].
Stránka byla naposledy editována 15. 5. 2001 [cit. 2007-12-20]. Dostupný z WWW: <http://www.env.cz/www/zakon.nsf/0/d639e197181a80c8c125653700310748?OpenDocument>.
- [5] *MSP ČR - Zadávání podmínek pro výběr subjektu* [online].
Stránka byla naposledy editována 4. 1. 2008 [cit. 2008-01-05]. Dostupný z WWW: <http://www.justice.cz/xqw/xervlet/insl/index?sysinf.@typ=or&sysinf.@strana=searchSubject>.
- [6] *ARES* [online].
Poslední editace neuvedena [cit. 2008-01-05]. Dostupný z WWW: http://www.info.mfcr.cz/ares/ares_fr.html.cz.
- [7] *RSW - Dotaz* [online].
Stránka byla naposledy editována 31. 12. 2007 [cit. 2008-01-05]. Dostupný z WWW: <http://dw.czso.cz/rswj/dotaz.jsp>.
- [8] *Umweltbundesamt: Online-Abfrage zu abfallwirtschaftlichen Anlagen* [online].
Poslední editace neuvedena [cit. 2008-01-05]. Dostupný z WWW: http://www.umweltbundesamt.at/umweltschutz/abfall/abfall_datenbanken/abfrage03/.
- [9] *Jak psát web, návod na html stránky* [online].
Stránka byla naposledy editována 7. 11. 2007 [cit. 2007-12-20]. Dostupný z WWW: <http://www.jakpsatweb.cz/>.

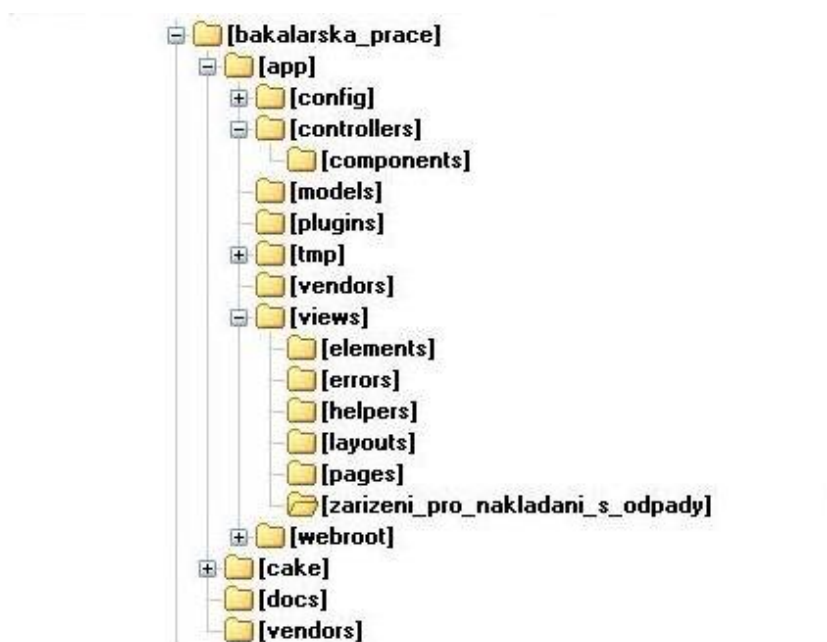
- [10] *Regular-Expressions.info - Regex Tutorial, Examples and Reference - Regexp Patterns* [online]. Stránka byla naposledy editována 17. 12. 2007 [cit. 2007-12-19]. Dostupný z WWW: <http://www.regular-expressions.info/>.
- [11] *Zákon č. 365/2000 Sb., o informačních systémech veřejné správy* [online]. Poslední aktualizace neuvedena [cit. 2007-12-19]. Dostupný z WWW: http://www.mvcr.cz/micr/scripts/detail.php_id_389.html.

Příloha A

struktura webu při použití frameworku CakePHP

Framework CakePHP zjednodušuje práci při tvorbě webových stránek, avšak je třeba dodržovat určitou strukturu. Zejména pak adresářovou strukturu (v našem případě byl domovským adresářem `bakalarska_prace`).

- Soubory třídy *Model* se ukládají do adresáře `/app/models`.
- Soubory třídy *Controller* se ukládají do adresáře `/app/controllers`.
- Soubory třídy *View* se ukládají do adresáře `/app/views/"jméno příslušné třídy controller"`. V našem případě šlo o `controller zarizeni_pro_nakladani_s_odpady`.
- Pomocné funkce lze ukládat jako instance třídy *Component* a následně používat v třídách *Controller*. Komponenty by měly obsahovat především logiku. Ukládají se do adresáře `/app/controllers/components/`.
- Opakující se části zdrojového kódu lze ukládat jako instance třídy *Helpers*. Mělo by jít především o formátovací úseky kódu (například značky HTML). Ukládají se do adresáře `/app/views/helpers/`.



Obrázek A.1: Adresářová struktura webu vystavěného pomocí frameworku CakePHP

↑ [..]		<DIR>	12/31/2007 17:06---
katastralni_uzemi	php	100	12/28/2007 21:08-a--
povoleny_odpad	php	90	12/28/2007 21:08-a--
provozovatel	php	771	12/14/2007 16:34-a--
provozovna	php	419	12/28/2007 21:08-a--

Obrázek A.2: adresář *bakalarska_prace/app/models*

↑ [..]		<DIR>	12/28/2007 22:37---
[components]		<DIR>	12/30/2007 09:43---
zarizeni_pro_nakladani_s_odpady_controller	php	2,041	12/30/2007 10:11-a--

Obrázek A.3: adresář *bakalarska_prace/app/controllers*

↑ [..]		<DIR>	12/29/2007 18:13---
index	thtml	2,208	12/30/2007 10:14-a--
katastralni_uzemi	thtml	943	12/29/2007 09:29-a--
napoveda	thtml	8,104	12/30/2007 20:14-a--
o_webu	thtml	459	12/30/2007 12:40-a--
podrobnosti	thtml	2,060	12/30/2007 10:34-a--
povolene_odpady	thtml	987	12/29/2007 09:35-a--

Obrázek A.4: adresář *bakalarska_prace/app/views/zarizeni_pro_nakladani_s_odpady*