

**Univerzita Pardubice
Fakulta ekonomicko-správní**

**Interaktivní aplikace pro prezentaci kontejnerů
na recyklovaný odpad v Pardubicích**

Bc. Miroslav Pásler

**Diplomová práce
2014**

Univerzita Pardubice
Fakulta ekonomicko-správní
Akademický rok: 2013/2014

ZADÁNÍ DIPLOMOVÉ PRÁCE

(PROJEKTU, UMĚLECKÉHO DÍLA, UMĚLECKÉHO VÝKONU)

Jméno a příjmení: Bc. Miroslav Pásler
Osobní číslo: E11634
Studijní program: N6209 Systémové inženýrství a informatika
Studijní obor: Informatika ve veřejné správě
Název tématu: Interaktivní aplikace pro prezentaci kontejnerů na recyklovaný odpad v Pardubicích
Zadávající katedra: Ústav systémového inženýrství a informatiky

Z á s a d y p r o v y p r a c o v á n í :

Cílem práce je navrhnout a vhodným způsobem implementovat interaktivní webovou aplikaci, která bude prezentovat rozmístění kontejnerů na separovaný odpad ve městě Pardubice. V rámci práce bude porovnáno několik možných způsobů implementace.

Zásady:

- Interaktivní prezentace prostorových dat v prostředí Internetu.
- Návrh a implementace aplikace ve zvolených prostředích.
- Zhodnocení výsledků implementace a doporučení pro projekty obdobného charakteru.

Rozsah grafických prací:

Rozsah pracovní zprávy:

Forma zpracování diplomové práce: tištěná/elektronická

Seznam odborné literatury:

BRANDON, D. M. Software Engineering for Modern Web Applications : Methodologies and Technologies. Hershey : IGI Global, 2008. 402 s. ISBN 978-1-59904-492-7.

KOMÁRKOVÁ, J., Kvalita webových geografických informačních systémů. Vyd. 1. 2008, Pardubice: Univerzita Pardubice. ISBN 978-80-7395-056-9.

LOOGNEY, P. Geographic information systems and science. first. Chichester: John Wiley & Sons, 2001. ISBN 0-470-87001-X.

PENG, Z.-R., TSOU, M.-H., Internet GIS: distributed geographic information services for the internet and wireless networks. c2003, Hoboken: John Wiley & Sons. ISBN 0-471-35923-8.

Vedoucí diplomové práce:


doc. Ing. Jitka Komárková, Ph.D.


Ústav systémového inženýrství a informatiky

Datum zadání diplomové práce: 1. října 2013

Termín odevzdání diplomové práce: 30. dubna 2014


doc. Ing. Renáta Myšková, Ph.D.
děkanka

L.S.


prof. Ing. Jan Čapek, CSc.
vedoucí ústavu

V Pardubicích dne 1. října 2013

PROHLÁŠENÍ

Prohlašuji, že jsem tuto práci vypracoval samostatně. Veškeré literární prameny a informace, které jsem v práci využil, jsou uvedeny v seznamu použité literatury.

Byl jsem seznámen s tím, že se na moji práci vztahují práva a povinnosti vyplývající ze zákona č. 121/2000 Sb., autorský zákon, zejména se skutečností, že Univerzita Pardubice má právo na uzavření licenční smlouvy o užití této práce jako školního díla podle § 60 odst. 1 autorského zákona, a s tím, že pokud dojde k užití této práce mnou, nebo bude poskytnuta licence o užití jinému subjektu, je Univerzita Pardubice oprávněna ode mne požadovat přiměřený příspěvek na úhradu nákladů, které na vytvoření díla vynaložila, a to podle okolností až do jejich skutečné výše.

Souhlasím s prezenčním zpřístupněním své práce v Univerzitní knihovně.

V Pardubicích dne 28. 4. 2014

Bc. Miroslav Pásler

PODĚKOVÁNÍ

Rád bych poděkoval vedoucí práce doc. Ing. Jitce Komárkové Ph.D. za odborné vedení, pomoc a rady, kterými přispěla k vypracování této diplomové práce. Dále bych chtěl poděkovat vyučujícím z Ústavu systémového inženýrství a informatiky za předané znalosti použité v této práci.

V neposlední řadě bych chtěl poděkovat přítelkyni, rodině a blízkým za neustálou podporu a pomoc, bez které by tato práce nevznikla.

ANOTACE

Tato diplomová práce se věnuje návrhu a implementaci interaktivní webové aplikace. Tato aplikace slouží k vhodné prezentaci rozmístění kontejnerů na recyklovaný odpad v Pardubicích na základě dat dříve získaných v rámci jiné diplomové práce na Univerzitě Pardubice. V rámci práce je vytvořeno více aplikací v různých nástrojích za účelem výběru nejlepšího řešení pro prezentaci kontejnerů a jiné projekty obdobného charakteru.

KLÍČOVÁ SLOVA

aplikace, GIS, web GIS, vizualizace, prezentace, API, JavaScript, Google, ArcGIS, Mapy.cz, hodnocení, rozhodovací procesy

TITLE

An interactive application for publishing recycle bins in Pardubice

ANNOTATION

This thesis deals with the design and implementation of interactive web application. This application serves to the suitable publishing of the recycle bins placement in Pardubice on the base of data early obtained as part of other thesis on University of Pardubice. In this thesis, there are more applications created in different tools for selection the best solution for the bins publishing and other project of similar kind.

KEYWORDS

application, GIS, web GIS, visualisation, publishing, API, JavaScript, Google, ArcGIS, Mapy.cz, evaluation, decision processes

OBSAH

Úvod	11
1. Prezentace prostorových dat	12
1.1 Prostorová data a jejich specifika	12
1.2 Možnosti prezentace prostorových dat	13
1.3 Interaktivní prezentace	16
2. Web GIS a služby mapových serverů	18
2.1 Mapy přístupné z webu.....	18
2.2 Web GIS a mapové servery	18
2.2.1 Funkce a použití	20
2.2.2 Architektura a technologie.....	21
2.2.3 Popis vybraných technologií	22
2.2.4 Nejběžnější služby	23
2.3 Mashup	24
3. Výběr nástrojů pro tvorbu webové GIS aplikace	26
3.1 Vymezení problému	26
3.2 Charakteristika cílové skupiny uživatelů	26
3.3 Popis použitých dat	27
3.4 Výběr nástrojů pro tvorbu aplikace	27
3.5 Stanovení požadavků na podobu a funkce aplikace	29
4. Návrh a tvorba aplikace pomocí vybraných nástrojů	32
4.1 Řešení pomocí nástroje Google Maps API	32
4.1.1 Popis nástroje Google Maps API	32
4.1.2 Vlastní tvorba aplikace pomocí Google Maps API	33
4.2 Řešení pomocí ArcGIS API	39
4.2.1 Popis nástroje ArcGIS API.....	39
4.2.2 Vlastní tvorba aplikace pomocí nástroje ArcGIS API	41
4.3 Řešení pomocí Mapy API.....	46
4.3.1 Popis nástroje Mapy API	46
4.3.2 Vlastní tvorba aplikace pomocí Mapy API.....	48
5. Výběr nejlepšího řešení.....	56
5.1 Stanovení a popis hodnotících kritérií	56
5.1.1 Základní požadavky	57

5.1.2	Přepínání vrstev	58
5.1.3	Vyhledání nejbližšího kontejneru.....	58
5.1.4	Vyhledání trasy k nejbližšímu kontejneru	58
5.1.5	Vyhledání lokality	59
5.1.6	Čas potřebný pro tvorbu aplikace	59
5.1.7	Rychlost	59
5.1.8	Podkladové mapy	60
5.1.9	Bohatost nástrojů API	61
5.1.10	Zpracování a podpora API	62
5.1.11	Licenční podmínky.....	63
5.2	Ohodnocení řešení dle kritérií.....	63
5.3	Stanovení vah kritérií	64
5.4	Výběr nejlepší varianty.....	65
5.5	Závěrečné zhodnocení a možnosti dalšího využití.....	66
Závěr.....		67
Použitá literatura.....		68
Přílohy		78

SEZNAM ILUSTRACÍ A TABULEK

Obrázek 1 - Reprezentace území pomocí skládání vrstev	13
Obrázek 2 - Internet v rámci typů programových řešení GIS.....	17
Obrázek 3 - Postavení Web GIS v rámci GIS	19
Obrázek 4 - Web GIS v rámci architektury klient/server	21
Obrázek 5 - Use-case diagram funkcí aplikace	30
Obrázek 6 - Struktura použitých KML dokumentů	34
Obrázek 7 - Zobrazení atributů vybraného kontejneru	34
Obrázek 8 - Využití jQuery pro přístup k datům KML dokumentu	37
Obrázek 9 - Možná podoba inicializační funkce v ArcGIS API	41
Obrázek 10 – Způsob experimentálního zjišťování převodní konstanty k	53
Obrázek 11 - Funkce pro zjištění vzdálenosti dvou bodů v Mapy API.....	54
Obrázek 12 – Výsledek výběru nejvhodnějšího nástroje.....	65
Tabulka 1 - Vlastnosti vybraných mapových serverů	24
Tabulka 2 - Vlastnosti vybraných softwarových nástrojů	28
Tabulka 3 - Hodnocení nástrojů - základní požadavky	57
Tabulka 4 - Hodnocení nástrojů – vyhledání trasy k nejbližšímu kontejneru	58
Tabulka 5 - Hodnocení nástrojů - vyhledání lokality	59
Tabulka 6 - Hodnocení nástrojů - čas potřebný pro tvorbu aplikace.....	59
Tabulka 7 - Hodnocení nástrojů - rychlost	60
Tabulka 8 - Hodnocení nástrojů - podkladové mapy.....	61
Tabulka 9 - Hodnocení nástrojů - bohatost nástrojů API	62
Tabulka 10 - Hodnocení nástrojů - zpracování a podpora API	62
Tabulka 11 - Hodnocení nástrojů - licenční podmínky	63
Tabulka 12 - Rozhodovací tabulka	63
Tabulka 13 - Saatyho matice a váhy kritérií	64
Tabulka 14 - Výsledné hodnocení	65

SEZNAM ZKRATEK A ZNAČEK

AJAX – Asynchronous JavaScript And XML

AMD – Asynchronous Module Definition

ASP – Active Server Pages

CORS – Cross-Origin Resource Sharing

CSS – Cascading Style Sheets

CSV – Comma-Separated Values

ČHMÚ – Český Hydro-Meteorologický Ústav

ČÚZK – Český Úřad Zeměměřičský a Katastrální

GIS – Geografické Informační Systémy

GPS – Globální Polohovací Systém

GPX – GPS eXchange format

HTML – HyperText Markup Language

HTTP – HyperText Transfer Protocol

IMS – Internetový Mapový Server

IT – Informační Technologie

JSON – JavaScript Object Notation

JSP – JavaServer Pages

KML – Keyhole Markup Language

OGC – Open Geospatial Consortium

PHP – Hypertext Preprocessor

S-JTSK – Souřadnicový systém Jednotné Trigonometrické Sítě Katastrální

SOAP – Simple Object Access Protocol

UGC – User-Generated Content

URL – Uniform Resource Locator

UTM33 – Univerzální Transverzální Mercatorův systém souřadnic

VGI – Volunteered Geographic Information

WFS – Web Feature Service

WGS84 – World Geodetic System 1984

WKID – Well-Known ID

WMS – Web Map Service

XML – Extensible Markup Language

ÚVOD

Od první mapy vyryté do kamene přibližně před šesti tisíci lety až po současnost ovládanou mapami od Googlu, existuje potřeba lidí nějakým způsobem vizualizovat prostředí, ve kterém žijí. Během dlouhé historie zobrazování geografických dat se více či méně měnil účel této vizualizace. Dnes se však více než kdy dříve do popředí dostávají dva pojmy, které vzájemnou synergií mění podobu a účel zobrazení prostorové informace. Jsou jimi *internet* a *komercializace*.

Škála možností map dostupných z internetu je tak obrovská, že jediná cesta jak zprostředkovávat uživateli všechny žádané informace je integrovat dříve samostatné služby, nástroje a informační zdroje. Není proto s podivem, že nejvyužívanější dostupné internetové mapové zdroje, ať již tuzemské jako jsou Mapy.cz nebo zahraniční jako jsou Google Maps, dávno překročili hranici prostého interaktivního zobrazení mapy určitého území. Můžeme se setkat s převratnými službami, jako jsou panoramatické snímky rozsáhlého území poskytovaných prostřednictvím služeb *Google Street View* nebo *Mapy.cz Panorama*, 3D mapy, integrace jízdních řádů, široké možnosti vyhledávání a v neposlední řadě i propojení s mobilními zařízeními.

Ruku v ruce s výše uvedeným stoupá i počet webových dostupných řešení zabývajících se jedním konkrétním problémem nebo malým okruhem problémů. Tyto projekty bývají často realizovány lidmi z řad široké veřejnosti, provozovány neziskově a využívající možnosti, které přináší fenomény *cloud computing* a *crowdsourcing*. Jedním z v současnosti nejoblíbenějších řešení tohoto druhu projektů je využití služeb, které poskytují mapové servery různých společností, přičemž forma poskytnuté služby se může lišit.

Cílem této práce je řešení konkrétního problému prezentace (vizualizace) kontejnerů na recyklovaný odpad v rámci zájmového území, jímž je město Pardubice. V souvislosti s tím je řešen výběr vhodného nástroje pro tvorbu interaktivní webové aplikace, která by umožňovala prezentaci kontejnerů na recyklovaný odpad, tvorba této aplikace ve zvolených nástrojích a výsledné porovnání těchto nástrojů. Výstupem práce je tedy interaktivní webová aplikace zprostředkovávající uživateli přehled o rozmístění kontejnerů na recyklovaný odpad v Pardubicích a poskytující vhodné interaktivní služby a dále doporučení vhodného nástroje pro tvorbu stejného nebo podobného druhu aplikací.

1. PREZENTACE PROSTOROVÝCH DAT

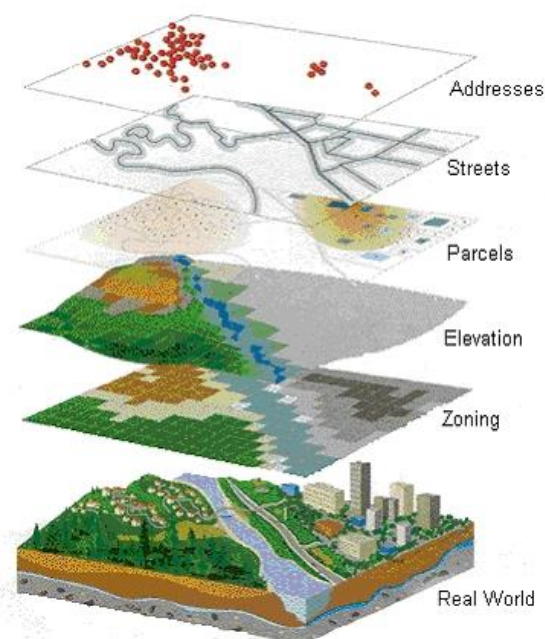
1.1 Prostorová data a jejich specifika

Prostorová data (někdy též geodata, geoprostorová data, angl. spatial data) jsou dle [1 s. 18] definována jako: „Data, která se vztahují k určitým místům v prostoru, a pro která jsou na potřebné úrovni rozlišení známa lokalizace těchto míst.“ Pojmy „prostorová data“ a „geodata“ mohou být vzájemně zaměnitelné, přestože u pojmu „geodata“ se za vztažený prostor předpokládá zemské těleso nebo jeho blízké okolí [2 s. 11]. Takto pojatý význam geodat vede až ke specifikaci prostorových dat jako informace, jež může být vyjádřena numericky pomocí geografických souřadnicových systémů [3]. Pojem prostorových dat bývá však nejčastěji definován právě v souvislosti s geografickými informačními systémy (GIS) a proto je obvykle rozdíl v definici obou těchto pojmů smazáván. [4] [5]

Důležitou vlastností prostorových dat je fakt, že nenesou pouze informaci o poloze objektu (prostorovou informaci), ale i informaci o vlastnostech objektu (atributová informace), časovou, vztahovou a funkční informaci. [4] [6 s. 21-22]

Další vlastností geografických prostorových dat, která je promítána do způsobu jejich vizualizace, je rozdělení dat dle datového modelu na data vektorová a rastrová. Tato problematika patří k základním otázkám řešeným v GIS a její bližší specifikaci lze nalézt například v [2 s. 211-220] [6 s. 35-41] [7] [8 s. 28].

Dále je pro vizualizaci prostorových dat třeba mít na zřeteli, že informace obsažená v takovýchto datech je modelem reality a že tedy při procesu modelování dochází k určité míře abstrakce reality [6 s. 35]. Z pohledu GIS je prostor na zemském povrchu, který je předmětem zkoumání, nazýván zájmové území. Nejběžnějším způsobem reprezentace reality (zájmového území) v GIS je použití mapových vrstev. Jedná se o způsob skládání výsledné vizualizované mapy, který umožňuje oddělený přístup k informacím, jež jednotlivá daná vrstva nese (za účelem specifické vizualizace, analýz, atp.). Princip reprezentace zájmového území pomocí vrstev a jejich skládání do výsledné podoby je zobrazen na obrázku níže (Obrázek 1). Z uvedeného vyplývá, že nejčastějšími kritérii, podle kterých dochází k dělení prostorových dat do vrstev, jsou hlediska datového modelu (vektorové vrstvy a rastrové vrstvy) a tematická hlediska (např. vodstva, výškopis, budovy,...). [9 s. 6] [8 s. 26]



Obrázek 1 - Repräsentace území pomocí skládání vrstev

Zdroj: [9 s. 6]

V neposlední řadě lze mezi specifika prostorových dat vzhledem k jejich vizualizaci zařadit jejich vztah ke kartografii a kartografickým vyjadřovacím prostředkům. Dle [10 s. 39] lze o kartografických vyjadřovacích prostředcích hovořit jako o základním elementu vizualizačních metod. Zejména povaha rozdělení vizualizovaných prostorových dat mezi plochy, line a body s sebou nese významná specifika vzhledem k jejich zobrazování. Pro srozumitelnost a správnost předávání informace v podobě vizualizace prostorových dat je klíčové dodržet zásady a zvyklosti použití kartografických vyjadřovacích prostředků [10 s. 39-50]

1.2 Možnosti prezentace prostorových dat

Přestože v běžných souvislostech mají pojmy „vizualizace“ a „prezentace“ podobný, ale nikoli stejný význam, pro účely této práce bude těmito pojmy rozuměno totéž. Z jiného pohledu lze pojem prezentace prostorových dat zařadit jako součást vizualizace spolu se sběrem údajů, jejich organizací a modelováním. [11 s. 7] [6 s. 52]

Dle [6 s. 52] můžeme vizualizaci (prezentaci) prostorových dat chápat jako: „Transformaci prostorových dat do viditelného obrazu k podpoře jejich vyšetření, zkoumání, poznání a vysvětlení.“ Jedná se tedy především o nástroj sloužící k interpretaci obrazových údajů. Z historického hlediska se vizualizací geodat zabýval téměř výhradně vědní obor zvaný kartografie (alespoň co se týká výstupů v podobě tištěných map). Prakticky veškerá mapová

díla tedy byla produktem odborníků při dodržení pravidel týkajících se vizualizace geodat, kterými se kartografie zabývá. S nástupem internetového věku byla však možnost vizualizace prostorových dat otevřena široké veřejnosti. Postupem času rostly možnosti interakce poskytovatele geografických datových služeb s jejich uživatelem, přičemž i hranice mezi těmito dvěma skupinami je stále obtížněji vymezená. Způsoby a možnosti vizualizace musely jít ruku v ruce s tím, jak se uživatelům interaktivně prezentovaných prostorových dat otvíraly širší možnosti vzájemné komunikace a množství prezentovaných informací stejně jako stále sílící trend zvyšování aktivity uživatele, který sám tvoří obsah se snižujícími se nároky na jeho odborné znalosti z oborů informatiky a kartografie. Dle [9 s. 9-10] v souvislosti s tímto fenoménem přechází na přelomu tisíciletí vizualizace geodat do věku, kdy hrají i v tomto oboru klíčovou roli fenomény jako jsou: *Cloud Computing*, *WEB 2.0*, *SaaS* a *Lightweight programming*. [9 s. 7-12] [12] [13]

Možností vizualizace prostorových dat je celá škála [14]. Níže jsou uvedena nejčastější dělení. Přestože by se mohlo zdát, že výsledkem vizualizace prostorových dat je mapa, tato problematika je složitější. Základní způsoby prezentace prostorových dat úzce souvisí s možnostmi výstupů z GIS a lze je dle [6 s. 52-53] rozdělit na:

- **analogové mapy** – způsob prezentace v podobě klasické tištěné mapy neumožňuje dynamickou práci s daty;
- **interaktivní zobrazení** – jedná se o grafickou komunikaci člověk – počítač (popř. jiné zařízení)[6 s. 52], která umožňuje v reálném čase interagovat s daty. Umožňuje uživateli ovlivnit podobu zobrazených informací [15 s. 4], tedy upravovat jejich zobrazení, zacházet s mapovými vrstvami, provádět nad daty prostorové analýzy apod. Interaktivní zobrazení lze dále dělit na:
 - webové – využívají prostředí internetu (mapové servery, webové mapové služby); [16]
 - ostatní – využívají moderních IT technologií, ale nepotřebují prostředí internetu (různé desktopové aplikace a prezentační nástroje);
- **další výstupy** – nedají se zařadit ani do jedné z předešlých skupin nebo jich lze použít jako součást obou. Jedná se o různé animace, grafy, diagramy a numerické výstupy.

Dle [6 s. 23] a [15] lze prezentaci prostorových dat dělit z jiného úhlu pohledu:

- **statická** – veškeré analogové mapy lze považovat za statické. Rozdíl od předchozího dělení spočívá především v tom, že i některá webová prezentace může být statická, stejně tak i mapové výstupy z GIS aplikací v různých formátech zobrazitelných v počítači nebo v prostředí webu mohou pouze nahrazovat analogové mapy a neposkytovat žádnou možnost interakce s uživatelem;
- **dynamická** – (někdy označována jako „klikací“ [16]) v dnešní době většina webových mapových služeb nabízí, alespoň minimální úroveň dynamiky v podobě posuvného mapového pole a změny měřítka. Dynamika vizualizace je možná také v podobě animovaných kartografických symbolů, integrace časového hlediska apod.

Přístup k vizualizaci je specifický také z hlediska prostoru [8 s. 25] [15] [17 s. 171]:

- **2D** – dvourozměrné (dvou-dimenzionální) zobrazení dat představuje zobrazení v ploše. Třetí rozměr bývá vyjádřen jako atribut. V tom případě se hovoří o 2,5 rozměrném prostoru. Třírozměrné objekty jsou transformovány a jsou reprezentovány příslušnými mapovými značkami;
- **3D** – přestože náročnost na velikost třídídimenzionálních dat znesnadňuje jejich vizualizaci v prostředí internetu, je možné ji vidět například při zobrazení některých budov v prostředí GoogleEarth. [18]

Dále lze dle [8 s. 68] [19 s. 95] [20 s. 2] [21] dělit způsoby vizualizace podle formy výstupu na:

- **tematické mapy** – tematickou mapu lze dle [8 s. 68] definovat takto: „*Tematická mapa je mapa, která na topografickém podkladu znázorňuje jedno nebo více zvláštních témat na úkor nepodstatných témat a je určena ke specifickému cíli.*“ Tematických map můžeme nalézt celou škálu (mapy zobrazující demografické údaje, environmentální údaje, ...);
- **komplexní mapy (též základní, všeobecné)** – do této kategorie lze zařadit i mapy topografické. Typicky se jedná o mapy, které zobrazují určité území bez konkrétního zaměření na jedno nebo malé množství témat. Jedná se

o různé autoatlasy, turistické mapy, podkladové mapy apod. (např. Základní mapa ČR 1 : 10 000).

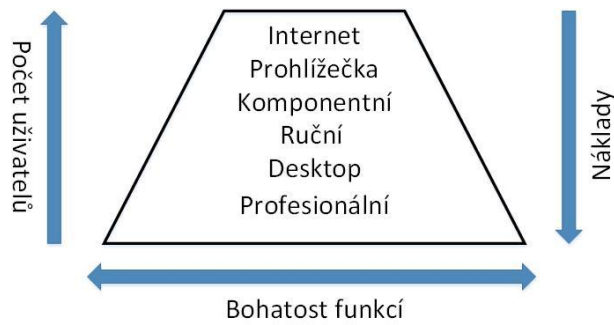
Rozdíl těchto dvou kategorií (tematické a komplexní) je patrný zejména u moderních webových mapových služeb. Zde však díky jejich dynamice a použití principu mapových vrstev i komplexní webové mapy mohou často poskytovat informace typické pro tematické mapy (zobrazení cyklotras na portálu Mapy.cz, zobrazení čerpacích stanic atp.).

Existují i další hlediska, podle kterých můžeme způsob vizualizace prostorových dat dělit do kategorií [20 s. 96,113]. Mezi nejvýznamnější patří:

- **měřítko** – rozlišujeme mapy malých (menší než 1:1 000 000), středních (1:200 000 až 1:1 000 000) a velkých měřítek (větší než 1:200 000). Pro způsob vizualizace je toto hledisko významné zejména v souvislosti s mírou generalizace zobrazovaných prvků, [8 s. 19-22,26]
- **účel prezentace** – zejména v souvislosti s předpokládanou charakteristikou uživatele hraje účel prezentace klíčovou roli. Jinou podobu budou mít výstupy komerční a propagační a jinou výstupy pro vojenské účely,
- **druh koncového uživatele** – významně ovlivňuje nejen způsob prezentace dat, ale také bohatost doprovodných funkcí u interaktivní prezentace. Přestože základní topografické zvyklosti jsou známy i laické veřejnosti, je třeba mít při vizualizaci na zřeteli i míru odbornosti uživatele,
- **souřadnicový systém** – zejména při přenosu dat mezi systémy je nutno pamatovat na dodržení souladu souřadnicových systémů. Elektronické zobrazovače prostorových dat (ať již webové nebo jiné) mohou pracovat s jiným souřadnicovým systémem, než v jakém jsou distribuována data [1 s. 25]. [22]

1.3 Interaktivní prezentace

Jak již bylo naznačeno v předchozí kapitole (1.2 Možnosti prezentace prostorových dat), v současnosti nejvyužívanějším způsobem prezentace prostorových dat je interaktivní přístup [20 s. 443]. Rozvoj informačních technologií (IT) technologií otevřel široké spektrum možností interaktivní práce s prostorovými daty [1 s. 11]. V oblasti prezentace prostorových dat v současné době jednoznačně dominuje využití prostředí internetu [23][24].



Obrázek 2 - Internet v rámci typů programových řešení GIS

Zdroj: vlastní dle [2 s. 193 - 200], [6 s. 13]

Jak si internetový GIS stojí v porovnání s ostatními GIS programovými balíky můžeme vidět na obrázku (Obrázek 1Obrázek 2). Internetový GIS je využíván největším počtem uživatelů a díky tomu s nejmenšími náklady na uživatele [6 s. 13]. Princip pyramidového rozložení zůstává zachován, oproti [6 s. 13] je useknutým vrškem pyramidy znázorněn fakt, že rozdílnost v bohatosti funkcí je postupem času zmenšována a internetové GIS disponují čím dál složitějšími funkcemi dříve běžnými pouze u desktopových aplikací. Tento fakt je dán dvěma faktory. První je faktor uživatele, tedy rostoucí vzdělanost („zběhlost“) laického uživatele při používání funkcí, které jí internetový GIS nabízí, rostoucí počet uživatelů internetu, využití GIS v terénu laickou veřejností. Druhý faktor souvisí s vývojem technologií. Oproti minulosti jsou stále lepší možnosti internetového spojení (i mobilního a bezdrátového), rychlost připojení a výpočetní možnosti na straně serveru.

2. WEB GIS A SLUŽBY MAPOVÝCH SERVERŮ

2.1 Mapy přístupné z webu

Přestože základním výstupem webových GIS je různá forma mapy, zdaleka ne všechny mapy přístupné pomocí internetu lze považovat za interaktivní nebo dokonce zařadit do kategorie Web GIS (viz definice dále). Statické publikování map v prostředí webu představuje zasazení mapy jako obrázku (GIF, JPG, ...) do webové stránky [25 s. 153,161]. Je možné nalézt řadu webových portálů sloužících k přístupu k mapám a jejich distribuci, nikoli však primárně k jejich interaktivnímu zobrazování v prostředí internetu. [9 s. 143]

Webová stránka, jejímž primárním cílem je zprostředkování geografických dat, se nazývá *geoportál* [26 s. 34]. Přestože lze nalézt mnoho případů, kdy je součástí geoportálu i možnost interaktivního zobrazování map, nejedná se o primární cíl, jako je tomu u mapových aplikací typu Mapy.cz (www.mapy.cz) nebo Google Maps (www.maps.google.com).

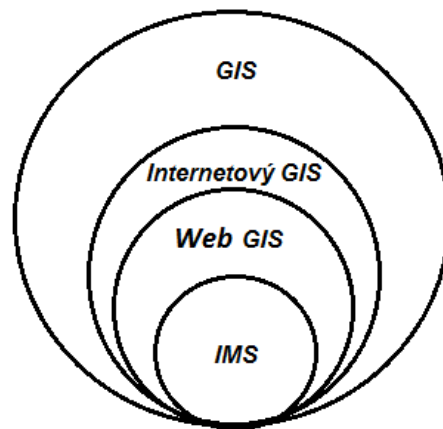
S nástupem fenoménů *Web 2.0* a *Cloud Computing* sílí trend vzájemné výměny informací mezi uživateli a decentralizované (distribuované) sdílení mapových zdrojů a prostorových informací (tematických map, kartografických map, modelů, analýz,...) [9 s. 9-10]. Vznikají tak hybridní portály (aplikace), jejichž primárním cílem je uspokojení široké škály uživatelů poptávajících různorodé mapové služby. Příkladem může být mapovací platforma ArcGIS Online od firmy Esri (<http://www.arcgis.com/features>).

2.2 Web GIS a mapové servery

Dle [9 s. 13] rozumíme pojmem *Web GIS*: „*jakýkoli GIS, který používá webové technologie ke komunikaci mezi komponentami.*“ Pojmy *Web GIS* a *webový GIS* jsou zaměnitelné. Jedná se o typ distribuovaného informačního systému. V nejjednodušší podobě se skládá ze serveru a klienta, kde klient nabývá podoby webového prohlížeče, ale také desktop aplikace nebo mobilní aplikace. Oblast Web GIS a mapových serverů můžeme chápat jako dynamicky se rozvíjející oblast GIS, v níž jsou využívány internetové a mobilní technologie. [9 s. 13]

V prostředí internetu je nevyužívanější technologií sloužící pro komunikaci mezi serverem a klientem protokol HTTP, stejně tak jedná-li se o Web GIS [25 s. 118-120]. Pojem Web GIS úzce souvisí s pojmy *Internetový GIS* a *mapový server*. Přestože na určité úrovni

jsou tyto pojmy zaměnitelné, existují zde některé rozdíly. Vztah těchto pojmů a jejich vztah ke GIS je ilustrován na obrázku (Obrázek 3). [9 s. 14]



Obrázek 3 - Postavení Web GIS v rámci GIS

Zdroj: vlastní dle [9 s. 14]

Web GIS může být považován za podmnožinu Internetového GIS, protože Web je pouze jedna ze služeb, kterou Internet zprostředkovává. Podobného charakteru je i rozdíl mezi pojmy Web GIS a mapový server. Zde je většinou zdrojů [27 s. 1120] pod pojmem internetový mapový server (IMS) rozuměna serverová část Web GIS, která zprostředkovává služby klientovi, na rozdíl od Web GIS chápaného jako informační systém využívající pro komunikaci mezi prvky (software, hardware, lidé, geodata, procedury) prostředí Webu [2 s. 25].

Dle [9 s. 15-16] a [28] lze mezi nejvýznamnější výhody Web GIS (především oproti tradičním desktop GIS aplikacím) zařadit tyto charakteristiky:

- celosvětový dosah
- velký počet uživatelů
- multiplatformita
- nízké náklady na uživatele
- snadné používání
- jednotná aktualizace
- různorodost aplikací

Uvedená fakta a povaha Web GIS implikuje také řadu obtíží a nevýhod spojených s tímto druhem GIS. Logicky prvním úskalím Web GIS je závislost na internetovém připojení. Přestože většinou na straně serveru i klienta dochází k určité míře použití

vyrovnávací paměti pro přednačítání map [9 s. 83], obsluha většiny služeb je závislá na stálém připojení k serveru. Míra závislosti je odvozená především od rozložení úloh mezi klienta a server (viz dále). Vyjdeme-li z faktu, že nejpoužívanějším druhem klienta jsou webové prohlížeče, je nutno uvést jako nikoli bezvýznamnou nevýhodu jejich omezené schopnosti. Přestože pomocí nástrojů jako jsou JavaScript nebo AJAX lze na straně klienta přizpůsobit vzhled a chování aplikace zpravidla dostatečně, jsou oproti desktop aplikacím možnosti omezené. [9 s. 39]

2.2.1 Funkce a použití

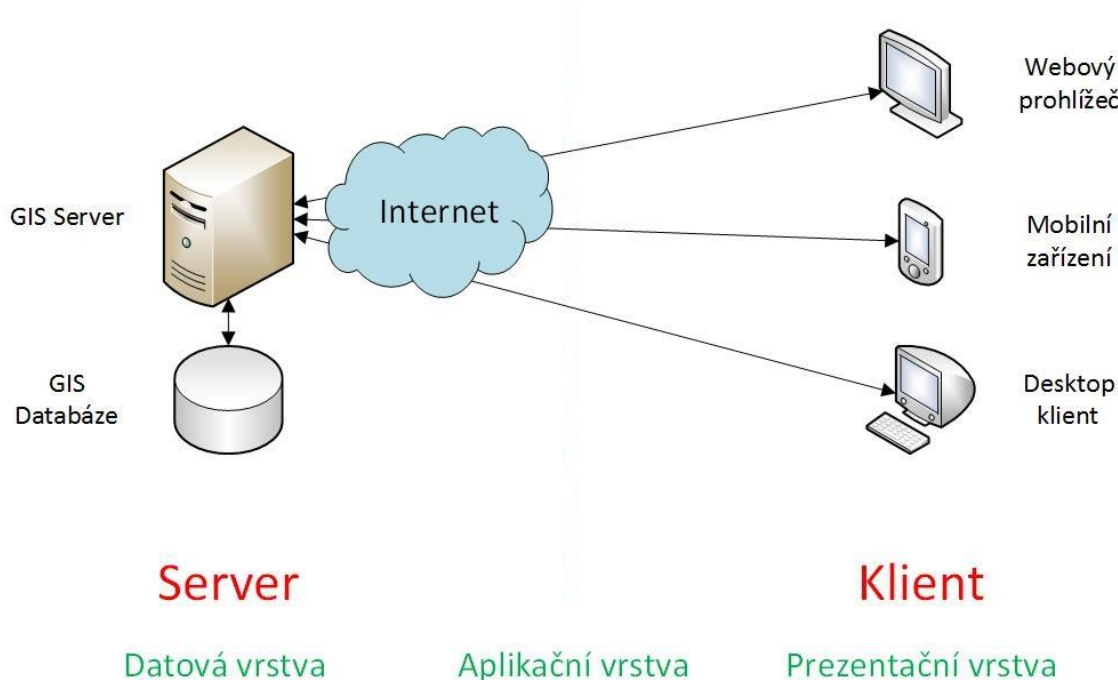
Přestože nejběžněji používanou funkcí Web GIS aplikací je vizualizace map [9 s. 17], možnosti Web GIS aplikací jsou větší a příliš nezaostávají za možnostmi klasických desktop aplikací (spíše je jejich rozsah uzpůsoben rozdílným požadavkům koncového uživatele a technickými omezeními). Mezi další běžné funkce dle [9 s. 17-18] patří:

- **sběr prostorových informací** – obor dříve vyhrazený pouze profesionálům (za použití speciální techniky, DPZ, a investice nemalých finančních prostředků) se za použití webu a zejména přenosných GPS zařízení dostává do rukou široké veřejnosti. Známým příkladem může být projekt OpenStreetMap (<http://www.openstreetmap.org>). Tento fenomén úzce souvisí s pojmy *user-generated content* (UGC) [29] a *volunteered geographic information* (VGI) [30 s. 2];
- **šíření prostorových informací** – pro distribuci prostorových informací je Web GIS ideální platforma. Existuje velká řada geoportálů umožňujících hledat, sdílet a stahovat prostorová data (ČÚZK,...);
- **prostorové analýzy** – mezi nejběžnější patří prosté měření vzdálenosti nebo vyhledávání trasy. Bohatost prostorových analýz je dána druhem koncového uživatele a zpravidla nebývá větší, než je třeba vzhledem k výpočetním nárokům a nárokům na podobu dat.

V oblasti každodenního života mají Web GIS nezastupitelnou a stále sílící roli zejména v souvislosti se stále významnější skupinou klientů (K-S architektura) v podobě mobilních telefonů. Potřeba znát informace jako: kde je nejbližší autobusová zastávka, jak se dostat z bodu A do bodu B nejrychleji, kudy vedou cyklotrasy v okolí nebo i kde se nachází nejbližší kontejner na recyklovaný odpad, dává Web GIS aplikacím nepřebornou škálu možností využití. [9 s. 18-21]

2.2.2 Architektura a technologie

Jak již bylo zmíněno, většina Web GIS aplikací pracuje na architektuře klient/server (C/S). Obvyklé Web GIS aplikace také využívají n-vrstvou architekturu skládající se z datové vrstvy, aplikační vrstvy (někdy logická) a prezentační vrstvy. Základní C/S architekturu Web GIS aplikace jako jednu z možných architektur webového GIS ukazuje obrázek (Obrázek 4).



Obrázek 4 - Web GIS v rámci architektury klient/server

Zdroj: vlastní dle [9] [25]

Jak je vidět na obrázku (Obrázek 4), data, aplikace a prezentace je rozložena mezi server a klienta tak, že data jsou umístěna na serveru, prezentace probíhá na klientu a aplikace běží z části na serveru (zpravidla analýzy vyžadující častý přístup k databázi) a z části na klientovi. S tím úzce souvisí pojmy tenký (slabý) klient a tlustý (silný, těžký) klient. Tenký klient představuje přístup, kdy je většina nebo celá aplikační vrstva přesunuta na server (v extrémním případě i část prezentační vrstvy), naopak v případě těžkého klienta na serveru zůstávají jen data popřípadě menší část aplikace (i část datové vrstvy se může nacházet na straně klienta). V kontextu Web GIS aplikace by u těžkého klienta probíhaly veškeré analýzy, dotazy a úkony na straně klienta, i část dat by se nacházela na straně klienta (například podkladové mapy apod.) a na straně serveru by byla pouze GIS databáze (například pouze poskytující data ve formě WMS služby). U tenkého klienta by na jeho straně probíhala pouze prezentace mapy se základní obsluhou webové aplikace. [9 s. 40-41] [25 s. 97-112] [31]

Technologie používané pro vývoj a správu web aplikací můžeme dle [9 s. 29,31,61-76] rozdělit na:

- **technologie na straně serveru**
 - webové servery – Apache, Tomcat,...
 - programovací jazyky – JSP, ASP, PHP,...
- **technologie na straně klienta**
 - webové prohlížeče – Internet Explorer, Firefox, Chrome,...
 - programovací jazyky – JavaScript, AJAX, HTML/CSS,...
- **formáty (standards) pro výměnu dat**
 - webové – XML, JSON,...
 - geografické – KML, KMZ,...
- **datové služby (standards, protokoly)**
 - webové – SOAP,...
 - geografické (mapové) – WMS, WFS,...

Technologií spojených s Web GIS aplikacemi je velké množství. Protože služby jako WMS jsou zároveň webové služby, je rozdělení mezi webové a geografické trochu zavádějící. Uvedené technologie souvisejí především pro web aplikace s webovým prohlížečem jako klientem. Pro mobilní a desktop aplikace se používají některé další technologie (Java, .NET, Python,...), i když technologie na straně serveru a standardy pro přenos dat zůstávají společné. [9 s. 36-38]

2.2.3 Popis vybraných technologií

Některé vybrané technologie nejčastěji využívané pro přenos dat a standardy pro výměnu dat (formáty datových souborů) – obecně specifikace, u kterých je vysoký předpoklad využití při tvorbě této práce, jsou popsány v následujícím výčtu:

- **XML** – eXtensible Markup Language – je značkovací jazyk vyvinutý konsorciem W3C pro standardizovaný přenos a ukládání dat ve strukturované formě. Je možné ho použít jako základ dalších jazyků, proto se v souvislosti s XML někdy hovoří o tzv. *metajazyku* [32]. Syntaxe XML využívá párových elementů tzv. tagů, jež obalují textovou informaci – hodnotu daného tagu [33] [34];

- **JSON** - JavaScript Object Notation – je (podobně jako XML) formát pro zápis a výměnu dat. Přebírá na JavaScriptu založenou syntaxi rodiny jazyků C. Na rozdíl od XML je textově úspornější, protože nepoužívá názvy tagů pro každý řetězec ve struktuře [35] [36];
- **KML** - Keyhole Markup Language – jedná se o jazyk (formát dat) založený na XML a určený primárně pro distribuci geografických dat. Je tedy standardně daná struktura a povolené tagy tohoto formátu. Pro zachycení souřadnic používá systém WGS84 ve tvaru desetinných stupňů. Od roku 2008 je KML 2.2 standardem podporovaným Open Geospatial Consortium (OGC). Často využívanou modifikací KML je KMZ, což je KML soubor volitelně spojený s dalšími podporovanými formáty a zabalený pomocí formátu ZIP. KMZ využívá především Nástroj Google Earth [37] [38] [39];
- **WMS** – Web Map Service – je standard vyvinutý OGC a poskytující HTTP rozhraní pro dotazování geodatabáze mapového serveru za účelem získání mapových vrstev od poskytovatele (server) směrem ke klientovi. Obvyklá odpověď je ve formě obrazových dat (JPEG, PNG,...) [40].

2.2.4 Nejběžnější služby

V této kapitole budou popsány nejběžnější služby poskytované mapovými servery. Oproti funkcím uvedeným v kapitole výše (2.2.1 Funkce a použití) se bude jednat především o podrobnější popis uživateli poskytovaných služeb (analýzy, práce s daty apod.).

Významnou roli při volbě služeb poskytovaných mapovým serverem, a podobě jejich zpřístupnění uživateli a ovládání, hraje důraz na zkušenost uživatele [9 s. 45]. Zkušeností se zde nemyslí vzdělanost uživatele, ale dle [9 s. 45] spíše: „úroveň uspokojení uživatele při používání produktu nebo služby.“ Jedná se tedy především o rozpor mezi tím, na co je zvyklý běžný uživatel webových aplikací a odborně vzdělaný a zkušený GIS pracovník. Klíčové je tedy poskytnout uživateli webové GIS aplikace požadované služby (nástroje), které by očekával i odborník, ale formou, na kterou je zvyklý ze svých zkušeností s jinými aplikacemi.

Následující tabulka (Tabulka 1) ukazuje vybrané mapové servery a jimi poskytované služby, funkce a jejich vlastnosti. Mapové servery jsou dostupné na adresách:

(www.maps.google.com), (www.mapy.cz), (www.maps.yahoo.com),
 (www.openstreetmap.org), (www.cykloserver.cz/cykloatlas), (www.kr-olomoucky.cz/mapy-cl-95.html)

Tabulka 1 - Vlastnosti vybraných mapových serverů

	Google Maps	Mapy.cz	Yahoo Maps	Open StreetMap	Cykloserver.cz	Mapy – Olomoucký kraj*
Podkladové mapy (klasické/satelitní)	1/1	3/3	1/1	5/0	1/0	1/0
Posouvání mapy	ANO	ANO	ANO	ANO	ANO	ANO (klikáním)
Způsob zoomování**	M/K/P	M/K/P	M/K/P	M/K/P	M/K/P	Tlačítko (klikáním)
Možnost vyhledávání	ANO	ANO	ANO	ANO	ANO	NE
Operační vrstvy	5	9	1	0	5	5
Plánovač trasy	ANO	ANO	ANO	NE	NE	NE
Přehledka	NE	NE	NE	NE	ANO	ANO
Měřítko	ANO	ANO	ANO	ANO	ANO	NE
Legenda	NE	NE	NE	ANO	ANO	NE

Zdroj: vlastní na základě uvedených adres

* jedná se o 4 oddělené tematické mapy ** M = double-click myši, K = kolečko myši, P = posuvník

Shodný způsob posouvání mapy, přibližování a některé funkce, jež se opakují u všech vybraných mapových serverů, ukazují, že jsou zvyklosti v zobrazování a ovládání dodržovány tak, aby uživatel mohl aplikaci ovládat intuitivně. Významným faktem je, že často není součástí prezentace mapy jeden (nebo více) ze základních kompozičních prvků mapy. Absence legendy poukazuje především na to, že tvůrci považují uživatele za natolik zvyklého pracovat se základní mapou, že běžně používané kartografické prvky jsou mu známé.

2.3 Mashup

Výraz *mashup* se obvykle do češtiny nepřekládá a v kontextu webových aplikací jej dle [41] můžeme chápat jako: „*Webovou aplikaci, která využívá obsah z více než jednoho zdroje k vytvoření samostatné služby zobrazené v jednom samostatném grafickém rozhraní.*“ Podstatou mashup není pouze „smíchání“ obsahu několika stránek, ale tato integrovaná kompozice by měla mít i nějakou svou vlastní přidanou hodnotu.

Mezi nejpopulárnější typy mashupů lze zařadit mapový mashup. Dle [41] a [42] mapové „mashupy“ tvoří kolem třiceti procent ze všech oblastí. Lze se setkat také s výrazem *geomashup*, který dle [9 s. 92] představuje: „*mashup, kde alespoň jedna ze součástí výsledného obsahu nebo funkcí je georeferencována.*“ Integrace mnoha mapových vrstev,

obvykle z několika zdrojů, je jednou z nejběžnějších funkcí webových GIS aplikací. Příkladem může být portál RadarBourky (www.radar.bourky.cz), který kombinuje aktuální informace o srážkách na území ČR poskytované Českým hydro-meteorologickým ústavem (ČHMÚ) a mapové podklady OpenStreet a MapQuest (dříve Google Maps).

2.4 Web GIS a vizualizace

Vizualizace geoprostorových dat v podobě interaktivního mapového pole je tváří prakticky všech Web GIS aplikací. Přestože geoprostorová data a výsledky prostorových a statistických analýz se dají prezentovat různým způsobem (viz. kapitola 1.2 Možnosti prezentace prostorových dat), je to právě interaktivní mapa, která s využitím možností webových technologií zajišťuje prezentaci, na kterou je koncový uživatel zvyklý a ví jak s ní intuitivně zacházet. [9 s. 17]

Jak bylo nastíněno v předchozích kapitolách, rozvoj Web GIS aplikací s sebou přinesl jistá specifika související s poskytovanými službami, obsahem, nástroji a způsobem vizualizace. Zavedené zvyklosti ustupují požadavkům uživatele a specifickým prostředí webových stránek. Proto není výjimkou, že se setkáme s Web GIS aplikacemi, u kterých součástí prezentace nejsou všechny základní kompoziční prvky mapy, nejsou dodrženy standardní symboly pro vyjádření kartografických prvků apod. Stejně jako je zacházeno s pojmem *mashup* jako s jistou formou hybridu vzniknuvšího spojením několika zdrojů, je možné i samotný Web GIS z pohledu vizualizace chápat jako hybridní spojení vizualizačních standardů, návyků a zvyklostí daných historicky pro mapy a moderního přístupu k interaktivní vizualizaci související s web-designem.

3. VÝBĚR NÁSTROJŮ PRO TVORBU WEBOVÉ GIS APLIKACE

3.1 Vymezení problému

Tato diplomová práce navazuje na poznatky a data získaná v rámci diplomové práce *Dostupnost kontejnerů pro separovaný odpad v Pardubicích* Lenky Skopalíkové. V souvislosti s názvem, zadáním a zásadami této práce je stěžejním řešeným problémem posoudit možnosti pro interaktivní webovou prezentaci dat získaných ve zmíněné diplomové práci. Hlavním výstupem této práce má tedy být nejenom výsledná webová GIS aplikace (pro každý zvolený nástroj jedna), která bude vhodným způsobem prezentovat získaná data, ale také zhodnocení, porovnání a doporučení, který z vybraných nástrojů je vhodný pro tento nebo podobný typ řešeného problému.

Předmětem prezentace pomocí zmíněné Web GIS aplikace jsou data v podobě kontejnerů na recyklovaný odpad. Tato data jsou specifická, co se týče jejich atributů, nicméně se jedná o vzorová data a problematiku vzájemného porovnání vybraných nástrojů neovlivní. Účelem práce je tedy vytvořit samostatné aplikace, u kterých není žádoucí, aby byly totožné, ale byly právě tak podobné (dle stanovených kritérií, implementovaných funkcí apod.), aby mohly být vybrané nástroje nejefektivněji porovnatelné.

3.2 Charakteristika cílové skupiny uživatelů

Jak bylo zmíněno dříve, charakteristika koncového uživatele Web GIS aplikace je jedním z klíčových faktorů determinujících celkovou podobu aplikace. Proto je nutné specifikovat cílovou skupinu uživatelů.

Vzhledem k danému problému bude výsledná Web GIS aplikace určena především široké veřejnosti. Při členění uživatelů Web GIS aplikací do čtyř skupin dle [43 s. 95-96] na: příležitostné uživatele, pravidelné uživatele, „high-end“ uživatele a mobilní uživatele, lze předpokládat, že množina uživatelů potenciálně využívající vytvořenou aplikaci, bude většinou zasahovat do prvních dvou zmíněných skupin. V rámci cílové skupiny uživatelů těžko hledat další specifika. Je očekávatelné, že služby aplikace budou využívat především lidé, kteří se aktivně věnují třídění odpadu. Na podobu aplikace však vyčlenění takovéto skupiny uživatelů nemá vliv. Výsledná podoba web GIS aplikace by tedy vzhledem k výše zmíněnému měla dodržet zvyklosti a podobu webových GIS aplikací určených široké veřejnosti.

3.3 Popis použitých dat

Povaha a struktura použitých dat je podrobně popsána ve zmíněné diplomové práci *Dostupnost kontejnerů pro separovaný odpad v Pardubicích*. Rozsah této práce dovoluje pouze orientační a stručný popis dat. Data jsou uspořádána do čtyř vrstev dostupných ve formátu *ShapeFile* [45]. Oproti původním získaným datům nebyla v této práci použita vrstva kontejnerů na drobné elektrospotřebiče a baterie. Také některé atributy vypovídající o stavu kontejneru nebyly použity, zejména pro svou obtížnou aktualizovatelnost. Fakticky jejich použití však nic nebrání. Všechny kontejnery se nacházejí na zájmovém území, kterým je město Pardubice a přilehlé části. V práci tedy byly využity následující atributy: adresa, název, typ kontejneru, druh kontejneru, přístup pěšky, přístup autem, otevíratelný, městský obvod.

3.4 Výběr nástrojů pro tvorbu aplikace

Nástrojů pro tvorbu webových GIS aplikací existuje velká řada. Využívají rozličné přístupy, technologie a licence použití. Díky tomu, že je pro tvorbu Web GIS aplikací třeba zpravidla využít několika různých technologií, což je dáno různými technologiemi pro stranu serveru a stranu klienta, nabízí se velké množství kombinací. Principiálně je možné pouze použitím technologií pro tvorbu webových aplikací (PHP, JavaScript, HTML,...) vytvořit Web GIS aplikaci, ale takový postup vyžaduje profesionální znalost daných technologií a je ho vhodné použít pouze v případě, že by se jednalo o složitou, hybridní aplikaci nebo nebylo možné využít žádného z dalších řešení určených pro tvorbu Web GIS aplikací. Obvykle je poskytované řešení zaměřeno na technologie na straně serveru, tedy je poskytován mapový server, nebo na technologie na straně klienta, které jsou pak dány především poskytovaným aplikačním programovacím rozhraním (API) daného řešení. Toto rozdělení znamená, že se technologie používané pro tvorbu Web GIS aplikací dělí na dvě skupiny:

- **komplexní mapové servery** – jedná se o programové řešení na straně serveru, které zprostředkovává komplexní služby mapového serveru. Nevýhodou je potřeba umístění webových stránek na server s nainstalovaným mapovým serverem. Pokud uživatel produktu využívá služby hostingové společnosti, musí se zpravidla spokojit s programovým vybavením již nainstalovaným; [49]
- **API mapových portálů (API mapových serverů)** – při využití tohoto řešení není třeba instalovat produkt na server a je tedy možné umístění stránek na libovolný webový server. Mapové portály jako *Google Maps* nebo *Mapy.cz* poskytují API pro webové technologie (nejčastěji JavaScript) a umožňují tak

umístit mapové pole do obsahu stránek. Někdy je však API poskytováno bez existence veřejného mapového portálu (například *OpenLayers*). Jedná se tedy o poskytovanou webovou mapovou službu umožňující tvorbu *mashup*. Díky poskytnutému API mapových portálů se v kombinaci s webovými technologiemi dá obecně dosáhnout srovnatelných výsledků jako při použití mapových serverů. Na rozdíl od poskytovaných komplexních mapových serverů je API mapových portálů poskytována zpravidla zdarma a to často i pro komerční účely. [46]

Nástrojů v obou kategoriích existuje velké množství. Vlastnosti vybraných softwarových nástrojů jsou zobrazeny v tabulce (Tabulka 2).

Tabulka 2 - Vlastnosti vybraných softwarových nástrojů

Mapové servery				
Název	Licence/Otevřenost	Programovací jazyk	Podporované OS	Podporované technologie
ArcGIS Server	Placená/Proprietární	.NET/Java	Windows, Linux	WMS, WFS, WCS, SOAP, ...
GeoServer	Freeware/OpenSource	Java	Windows, Linux, Mac OS	WMS, WFS, WCS,...
MapServer	Freeware/OpenSource	C	Windows, Linux, Mac OS	WMS, WFS, WCS,...
T-MapServer	Placená/Proprietární	-	Windows, Linux	WMS, SOAP,...
API mapových portálů				
Název	Licence	Programovací jazyk	Podporované formáty dat a technologie	
Google Maps API	Nutná registrace (zdarma)	JavaScript	KML (GeoRSS), WMS,...	
Mapy API	Zdarma i pro komerční použití	JavaScript	KML, GPX, WMS,...	
OpenLayer	BSD (zdarma)	JavaScript	KML, WMS,...	
ArcGIS API	Zdarma pro nekomerční využití	JavaScript	KML, CSV, WMS,...	

Zdroj: vlastní dle [47][48][49][50][51][52][53][54]

Vzhledem k tomu, že u API mapových portálů se jedná v podstatě o poskytovanou službu, je tomu přizpůsoben i způsob licencování. Některé mapové portály (např. Google Maps) poskytují tuto službu omezenou určitým počtem návštěv apod.

Pro snadnost přístupu k technologii resp. nepotřebu instalace doplňkových prostředků a především pro lepší možnosti závěrečného srovnání bylo rozhodnuto, že všechny nástroje vybrané pro tvorbu web GIS aplikace budou z kategorie API mapových portálů (API mapových serverů). Po prozkoumání možností technologií uvedených výše, vzhledem k povaze řešeného problému, definování cílové skupiny uživatelů a po konzultaci s vedoucí práce byly vybrány pro tvorbu aplikace následující tři nástroje z důvodů:

- **Mapy API** – zástupce nástroje umožňujícího tvorbu aplikací i pro komerční využití. Je nejpoužívanějším českým poskytovatelem API pro vkládání map do obsahu webových stránek (vzhledem k [24]).
- **ArcGIS API** – V rámci ústavu *Systémového inženýrství a informatiky* jsou produkty firmy ESRI dlouhodobě využívány a ArcGIS API s nimi dovoluje vzájemnou interakci.
- **Google Maps API** – Jedná se o široce využívanou dynamickou a robustní službu v současnosti nejvyužívanějšího mapového portálu [55].

3.5 Stanovení požadavků na podobu a funkce aplikace

Pro možnost výsledného porovnání jednotlivých použitých nástrojů je nutno předem stanovit základní požadavky na podobu a funkce aplikace, které je třeba dodržet při tvorbě pomocí všech třech nástrojů. Pokud z nějakého důvodu nebude dodržen nějaký z požadavků, bude tento fakt uveden v závěrečném porovnání jednotlivých nástrojů a bude zohledněn v hodnocení nástrojů.

Základní požadavky na podobu a funkce aplikace byly vzhledem k výše zmíněnému, struktuře dat a po konzultaci s vedoucí práce stanoveny takto:

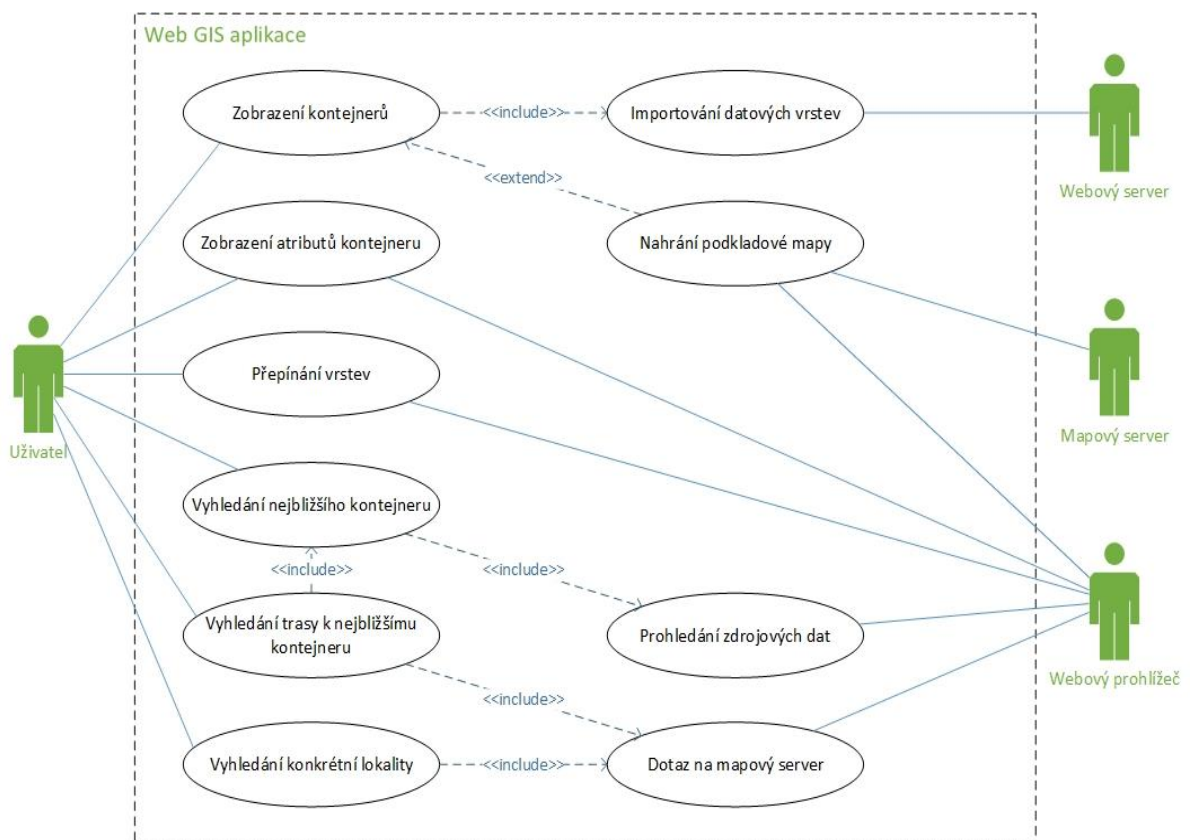
- webová GIS aplikace musí umožnit uživateli zobrazení umístění kontejnerů na recyklovaný odpad v Pardubicích,
- podoba aplikace musí dodržovat zásady a zvyklosti pro podobu Web GIS aplikací uvedených v kapitole 2.4 Web GIS a vizualizace,
- vizualizovaná data budou graficky rozlišena dle typu kontejneru,
- aplikace umožní uživateli formou dotazu (při kliknutí na daný objekt kontejneru na mapě) zjistit vhodné atributové informace o daném objektu.

Mimo tyto základní požadavky jsou stanoveny rozšířené požadavky na funkce aplikace. Implementace těchto požadavků bude odvislá od možností jednotlivých nástrojů.

Pro možnost efektivního závěrečného porovnání jednotlivých nástrojů, mají tyto funkce stanovené pořadí, v němž by měli být při tvorbě aplikace implementovány. Rozdělení požadavků na základní a rozšířené bude zohledněno ve způsobu závěrečného porovnání jednotlivých řešení. Byly stanoveny tyto rozšířené požadavky:

1. možnost přepínání vrstev dle typu kontejneru,
2. možnost vyhledání nejbližšího kontejneru dle typu (z daného místa na mapě),
3. možnost vyhledání trasy k nejbližšímu kontejneru,
4. možnost vyhledání konkrétní lokality (ulice, čtvrti) na mapě.

Požadované funkce jsou dostatečně obecného charakteru, aby stejné nebo podobné řešení bylo aplikovatelné i na jiná tematická prostorová data (nejen na kontejnery na separovaný odpad). Funkce aplikace stanovené jako součást základních i rozšířených požadavků jsou znázorněny pomocí Use-case diagramu (Obrázek 5) sestaveného dle [56] [57] [58].



Obrázek 5 - Use-case diagram funkcí aplikace

Zdroj: vlastní

Rozdíl mezi webovým serverem a mapovým serverem je v tom, že na webovém serveru je umístěna vlastní aplikace, datové vrstvy apod., mapový server poskytuje

podkladovou mapu, určuje její chování, poskytuje API a zpracovává případné dotazy. K use-case diagramu byly sestaveny následující scénáře (popis) jednotlivých činností:

- **Zobrazení kontejnerů** – proběhne automaticky, když uživatel vstoupí na webovou stránku. Případ nahrání samotné webové stránky nebyl do diagramu případů užití zanesen. Klient (C/S architektura) ve formě webového prohlížeče zadá požadavek na webový server, který vrátí obsah stránky a datové vrstvy kontejnerů (případ užití *Importování datových vrstev*). Prohlížeč se pak postará o jejich vykreslení dle pravidel získaných z mapového serveru. Mapový server také poskytuje podkladovou mapu.
- **Zobrazení atributů kontejneru** – uživatel zobrazí atributy kontejneru kliknutím na značku daného kontejneru v mapě. O zobrazení se starají programové prostředky na straně klienta – webového prohlížeče.
- **Přepínání vrstev** – uživatel může vybrat jaké vrstvy (dle typu kontejneru) budou zobrazeny. O skrytí a zobrazení vybrané vrstvy se opět starají programové prostředky na straně klienta.
- **Vyhledání nejbližšího kontejneru** – uživatel vybere místo odkud chce hledat nejbližší kontejner (například místo bydliště) kliknutím do mapy. Programové prostředky na straně prohlížeče prohledají zdrojová data a naleznou nejbližší kontejner. Výsledek je sdělen uživateli.
- **Vyhledání trasy k nejbližšímu kontejneru** – pomocí nástrojů poskytnutých mapovým serverem proběhne vyhledání trasy k nejbližšímu kontejneru. Tento případ užití v sobě zahrnuje *Vyhledání nejbližšího kontejneru*. Požadavek (dotaz) provede webový prohlížeč.
- **Vyhledání konkrétní lokality** – uživatel zadá název ulice nebo čtvrti do příslušného textového pole. Dále potvrdí, že chce vyhledat danou lokalitu. Webový prohlížeč předá požadavek mapovému serveru. Ten vrátí souřadnice umístění hledané lokality. Prohlížeč zobrazí uživateli hledanou lokalitu na mapě.

4. NÁVRH A TVORBA APLIKACE POMOCÍ VYBRANÝCH NÁSTROJŮ

V této kapitole budou postupně popsány vždy nejdříve vlastnosti jednoho z nástrojů a po té vlastní tvorba aplikace v daném nástroji.

4.1 Řešení pomocí nástroje Google Maps API

4.1.1 Popis nástroje Google Maps API

Produkt Google Maps existuje v dnešní podobě od poloviny prvního desetiletí nového milénia [59] [60]. V červnu roku 2005 byla spuštěna služba Google Maps API [61], která poskytuje vývojářům možnost integrace Google map do jejich webových stránek. Přestože je tato služba primárně zdarma, během let docházelo k drobným změnám licence zpravidla s příchodem nové verze Google Maps API pro JavaScript [62]. V současné době při verzi *Google Maps JavaScript API v3* je nekomerční využití zdarma, ovšem je nutné integrovat do JavaScript kódu stránek speciální API klíč a tedy vlastnit účet na Google (zcela zdarma) [63]. Pomocí přiděleného klíče svázaného s účtem může Google monitorovat počet přístupů k mapě. Pro komerční využití je k dispozici verze *Google Maps API for Business* [64].

Využití Google Maps API pracuje na principu *mashup* (viz. kapitola 2.3 Mashup), kdy je do běžné webové stránky integrována mapa pocházející z externího zdroje (není umístěna na stejném serveru jako samotné stránky). Pomocí API je pak možné upravovat chování a podobu mapy vždy s ohledem na možnosti poskytované daným API. Vlastní mapu lze do kódu stránky vložit jednoduše pomocí JavaScriptu, kdy je nutné pouze integrovat externí skript [63][65], přičemž k odkazu na zdroj skriptu je připojen vygenerovaný API klíč, a inicializovat mapu pomocí inicializační funkce. V tom nejjednodušším provedení musí inicializační funkce obsahovat pouze vytvoření objektu mapy a přidělení daných vlastností [63]. Mezi hlavní funkcionality, které Google Maps API nabízí pro úpravu podoby mapy dle [66] a [63] lze zařadit:

- **základní nastavení mapy** – provádí se zpravidla v těle inicializační funkce a to v rámci proměnné *MapOptions* [66]. Zde je možné nastavit celou řadu základních vlastností mapy, přičemž mezi nejvýznamnější patří: počáteční měřítko, počáteční souřadnice (vycentrování mapy), minimální a maximální přiblížení, nastavení posouvání, nastavení vizuálních vlastností (barva podkladu), nastavení podkladové mapy a další;

- **události** – Google Maps API nabízí vlastní podmnožinu podporovaných událostí pro API objekty (mapy, řídicí prvky, značky atp.). Události jakého typu daný prvek podporuje je podrobně popsáno v dokumentaci k API; [66]
- **řídicí prvky** – zobrazená mapa defaultně obsahuje několik prvků, které slouží k řízení chování mapy (posun, přibližování, rotace, změna podkladu,...). Tyto prvky je možné z mapy odstranit, měnit jejich podobu a umístění. Také je možno vytvořit vlastní kontrolní prvek a přiřadit mu vlastní obsah i chování; [63]
- **překryvy (overlays)** – jedná se o veškeré prvky vložené do mapy, které ji překrývají (značky, linie, polygony, ikony, symboly, obrázky,...). Obvykle pro všechny tyto objekty v Google Maps API existuje předdefinovaná třída (*Marker, Polyline, Polygon,...*) s bohatou škálou možností nastavení vzhledu a chování; [63]
- **mapové vrstvy** – Google Maps API primárně podporuje formáty KML/KMZ. Dokáže importovat tyto vrstvy do mapy včetně přednastaveného chování a symbologie, nedokáže však k prvkům vrstev přistupovat jako k objektům (primárně není možné získat nebo měnit obsah vrstev pomocí metod poskytovaných Google Maps API); [63]
- **služby** – do této kategorie spadají služby, které poskytuje server Googlu a tudíž zpravidla existují denní limity na jejich poskytování. Patří sem geokódování (vyhledání místa na mapě dle zadané adresy), vyhledávání trasy, výškopis a služba Street view (ta není omezena limitem); [63]
- **knihovny** – jak bylo zmíněno výše, Google Maps API pro JavaScript se do obsahu stránky integruje jako externí skript. Existuje však řada knihoven, které je třeba nahrát zvlášť (pokud je jejich obsah pro práci žádoucí). [63]

4.1.2 Vlastní tvorba aplikace pomocí Google Maps API

Zdrojová data bylo pro použití v řešení pomocí Google Maps API nutno převést do podporovaného formátu KML. Kromě volně dostupných specializovaných konvertorů, je možné data ve formátu *shapeFile* převést na formát KML/KMZ pomocí nástroje *Layer to KML* v *ArcGIS ArcMap*. Konverze pomocí tohoto nástroje zajišťuje zachování vlastností vrstvy uchovaných v atributové tabulce a automaticky je převádí do HTML formátu (uvnitř KML) takovým způsobem, že po importování vrstvy do mapy je tato vrstva dotazovatelná kliknutím a tyto údaje jsou zobrazeny. Formát zobrazovaných dat (všeobecně i obsah

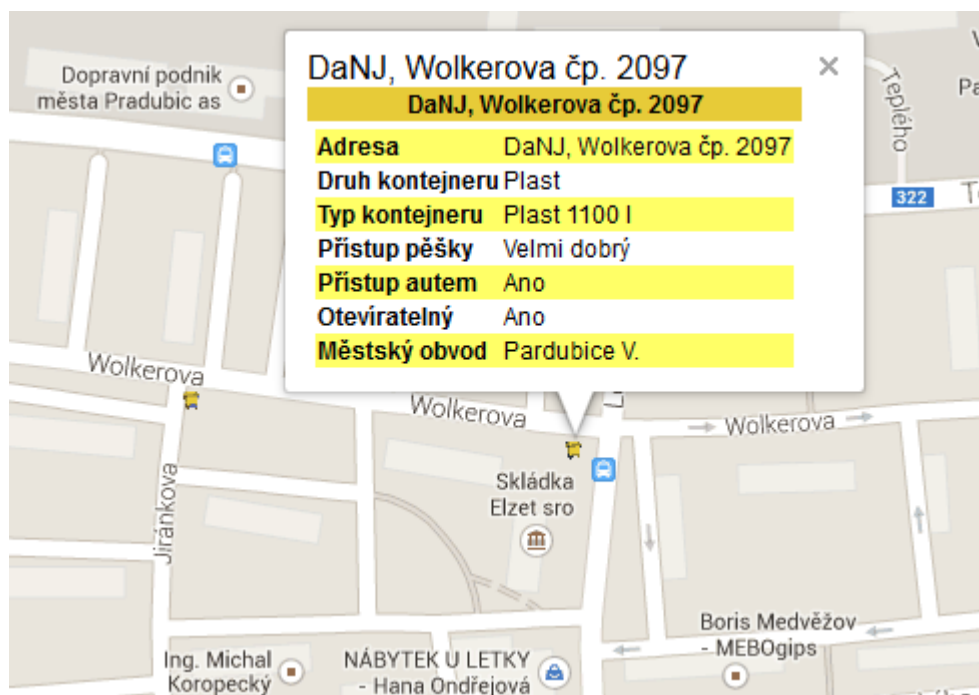
reprezentovaný KML souborem) lze měnit přímo v KML souboru. Strukturu použitých KML souborů ilustruje obrázek (Obrázek 6).

```
<xml>
<kml>
  <Document>
    <name></name>
    <Folder>
      <name></name>
      <Placemark>
        <name></name>
        <description> html kód </description>
        <styleURL></styleURL>
        <Point>
          <altitudeMode></altitudeMode>
          <coordinates></coordinates>
        </Point>
      </Placemark>
      .placemarks
    </Folder>
    <style>...</style>
  </Document>
</kml>
```

Obrázek 6 - Struktura použitých KML dokumentů

Zdroj: vlastní

Výslednou podobu zobrazených údajů z atributové tabulky pro vybraný kontejner ukazuje obrázek (Obrázek 7). Zobrazované údaje pro jednotlivé druhy kontejnerů jsou barevně odlišeny dle jejich typu (modře, žlutě a zeleně).



Obrázek 7 - Zobrazení atributů vybraného kontejneru

Zdroj: vlastní

Jednotlivé vrstvy, dle druhu kontejneru reprezentované buď žlutou, modrou nebo zelenou ikonkou kontejneru (plast, papír a sklo) je možné také zapínat a vypínat pomocí zaškrťávacích tlačítek na panelu v pravém horním rohu mapy. Tento panel slouží zároveň i jako legenda pro mapové vrstvy. Vzhledem k tomu, že se jedná o účelovou tematickou mapu, legenda popisující prvky podkladové mapy (silnice, ulice, zeleň,...) nebyla do aplikace implementována. Panel s legendou je možné skrýt, přičemž zůstává zachováno, které vrstvy uživatel zvolil skrýt nebo zobrazit (primárně jsou všechny vrstvy zobrazené). Samotný prvek je vložen do mapového pole (resp. ho překrývá) jako vlastní (nepředdefinovaný) řídicí prvek. Použití řídicího prvku má výhodu v tom, že je k němu automaticky přistupováno jako k součásti mapy, je tedy možné, aby ji překrýval bez zásahu do HTML kódu stránky (například využitím z-souřadnice). Existují zde ovšem i určitá omezení. Google Maps API umožňuje pozicovat řídicí prvky pouze na předem dané pozici na mapě [63]. Pro každou pozici existuje kolekce prvků, na této pozici umístěných. Pokud je na stejnou pozici umístěno více prvků, dochází k zařazení prvku kolekce s vyšším pořadovým číslem (zpravidla přidán jako poslední) pod prvek s nižším pořadovým číslem. K této kolekci se dá jednoduše přistoupit: `map.controls[google.maps.ControlPosition.pozice]`, kde *pozice* představuje jednu z možných pozic prvku (například `RIGHT_TOP`). Pomocí metody `removeAt()` je tedy nejdříve třeba odstranit prvek z kolekce a pomocí metody `push()` na jeho místo umístit jiný. Prvek je vytvořen inicializační funkcí a je tedy chápán jako objekt, proto po vyjmutí z kolekce nezaniká a může být do kolekce vrácen. Toho se dá využít při skrývání panelu legendy, kdy je samotný panel z kolekce vyjmut a nahrazen tlačítkem umožňujícím opětovné zobrazení legendy. Díky výše popsanému přístupu, kdy je objekt reprezentující legendu zachován, je možné zachovat také volbu vrstev. Samotná podoba legendy je vytvářena pomocí JavaScriptu a to konkrétně za použití vlastnosti `innerHTML`, která rodičovskému prvku (například obalovému tagu `<DIV>`) přiřadí HTML kód. Díky tomu je možné formátovat legendu jako samostatnou HTML stránku pomocí HTML tagů a kaskádových stylů.

Způsob vizualizace bodů reprezentujících kontejnery (obrázek, velikost, chování) je součástí samotného KML dokumentu. Specifikace zajišťující konkrétní způsob vizualizace je součástí tagu *Style*. Jednotlivých stylů může KML dokument obsahovat libovolné množství, přičemž každému stylu je přiřazeno ID a pomocí tohoto ID je každé jednotlivé položce vrstvy (v tomto případě každému kontejneru) přiřazen požadovaný styl zobrazení. Protože stejná velikost ikon není vyhovující pro všechny úrovně přiblížení mapy (malé ikony při velkém přiblížení jsou špatně viditelné, velké ikony při malém přiblížení nepůsobí esteticky dobře),

bylo třeba zajistit různou velikost ikon vzhledem k úrovním přiblížení. Tohoto efektu bylo docíleno nahráním tří různých KML vrstev pro jednotlivý druh kontejnerů, které se od sebe liší rozdílnou velikostí ikon nastavených v tagu *Style*. Tyto vrstvy se poté střídavě zapínají a vypínají v reakci na událost vyvolanou změnou přiblížení mapy. Zároveň s tím bylo sérií podmínek zajištěno, aby na vypínání a zapínání vrstev v legendě reagovala pouze právě zobrazená vrstva a aby se tento údaj přenášel i při změně vrstvy dle přiblížení.

Google Maps API primárně neobsahuje třídu ani knihovnu, která by podporovala práci s kontextovým menu, které se zpravidla zobrazuje při kliknutí pravým tlačítkem na libovolné místo v mapě. Například u originálních Google Maps lze pomocí této volby vyhledat trasu z daného místa nebo dotázat se na dané místo na mapě. Pro vytvořenou aplikaci bylo kontextové menu vytvořeno za využití externího JavaScript souboru (volně dostupného z: <http://code.martinpearman.co.uk/googlemapsapi/contextmenu>). Tento soubor umožňuje vytvořit vlastní obsah kontextového menu a sám se stará o to, aby se menu zobrazovalo v místě kliknutí do mapy a zachovávalo si předdefinovaný formát (ten je možno upravit). Obsah kontextového menu je v aplikaci implementován v inicializační funkci mapy. Pro každý prvek menu se jako jedna z vlastností uchovává název události obsluhující daný prvek menu. Kontextovému menu je poté přiřazen posluchač událostí, který se ve své službě přiřazuje danému prvku požadovanou funkcí.

Funkce, které aplikace nabízí skrze kontextové menu, jsou: nalezení nejbližšího kontejneru z místa kliknutí (pro každý typ jedna položka menu), vyhledání trasy k nejbližšímu kontejneru a odstranění vyhledané trasy z mapy. Možnost nalezení nejbližšího kontejneru vyhledá nejbližší kontejner dle typu vzdušnou čarou od místa kliknutí na mapě. Vyhledávání trasy využívá poskytovaný vyhledávač tras, u něhož je nastavena možnost „pěšky“. Obě tyto funkce potřebují ke své činnosti projít souřadnice všech kontejnerů na mapě a nalézt vzdálenost k nejbližšímu získáním vzdálenosti mezi souřadnicemi kontejneru a souřadnicemi kliknutí. Souřadnice kliknutí se (pomocí výše zmíněného externího skriptu) odevzdávají funkci, která má na starost obsluhu kontextového menu a přes ni dále jednotlivým funkcím starajícím se o vyhledávání kontejneru. Algoritmus vyhledávání funguje tak, že projde všechny získané souřadnice všech kontejnerů, přičemž je do proměnné ukládána vždy nová nejmenší vzdálenost. Informace o vzdálenosti vyhledaného kontejneru je uživateli předána pomocí dialogového okna a v mapě je vykreslena spojnice mezi místem kliknutí a nalezeným kontejnerem.

Jak bylo zmíněno v předchozí podkapitole, Google Maps API umožňuje vkládání KML vrstev do mapy, ale nedisponuje metodami, které by umožňovaly přístup k strukturovanému obsahu KML souboru. Tento problém by řešil objektový formát dat JavaScript Object Notation (JSON), k jehož struktuře lze přistupovat pomocí JavaScriptu. Převádění KML do formátu JSON by však vedlo ke ztrátě podpory některých metod Google Maps API. Problém přístupu ke strukturovanému obsahu XML dokumentů však řeší jQuery, což je JavaScriptová knihovna usnadňující (kromě jiného) práci s HTML a XML [67]. Do obsahu stránek ji lze integrovat stejně jako jakýkoli Javascript [65] externím odkazem nebo knihovnu stáhnout přímo z domovských stránek projektu (<http://jquery.com/>) a umístit ji na vlastní server. Díky využití metod knihovny jQuery je možné projít všechny jednotlivé položky strukturovaného dokumentu dle zadané specifikace. V případě vytvořené aplikace je pro každou položku KML dokumentu, která představuje jeden konkrétní kontejner daného typu, vyhledáván obsah tagu *coordinates*. Ten obsahuje souřadnice daného dokumentu, jež je pro další práci třeba pomocí textových funkcí a metod Google Maps API převést do formátu *LatLng*, se kterým umí pracovat Google Maps API [63]. Ukázku části KML dokumentu (červeně orámováno) a obslužného JavaScript kódu (černě orámováno) s využitím jQuery ilustruje obrázek (Obrázek 8).

```
$(kml).find("Placemark").each(function () {  
  
    var x = $(this).find("coordinates").text();  
  
    var lng = x.substring(1,x.indexOf(",");  
    var lat = x.substring(x.indexOf(",")+1,x.lastIndexOf(",");  
    var kontLatLng = new google.maps.LatLng(lat, lng);  
    var distance = google.maps.geometry.spherical.computeDistanceBetween(latLng, kontLatLng);  
  
    </description>  
    <styleUrl>#IconStyle00</styleUrl>  
    <Point>  
        <altitudeMode>clampToGround</altitudeMode>  
        <coordinates> 15.86600270728887,50.00289448254867,0</coordinates>  
    </Point>  
    </Placemark>
```

Obrázek 8 - Využití jQuery pro přístup k datům KML dokumentu

Zdroj: vlastní

Pomocí funkcí *each* a *find*, které nabízí jQuery je možné selektivně procházet obsah dokumentu a dále s tímto obsahem pracovat jako s textem.

Dále byla do aplikace implementována funkce vyhledávání (geokódování). Uživatel tak může vyhledat konkrétní ulici nebo čtvrť v Pardubicích. Vyhledávání je řešeno intuitivně

stejnou cestou, jako je tomu prakticky u všech internetových mapových portálů, které tuto možnost nabízejí (Google Maps, Mapy.cz,...), tedy vstupním formulářovým prvkem pro vkládání textu a tlačítkem „Vyhledat“. Vyhledávání pozic na mapě je pomocí Google Maps API prováděno přes službu *Geocoding*, kdy se jedná o proces konvertování adresy (zadáva uživatel) na geografické souřadnice (v případě Google Maps API *LatLng* formát) [63]. Přístup ke službě probíhá prostřednictvím objektu `google.maps.Geocoder` a metody `Geocoder.geocode()`, která přenáší dotaz (objekt `GeocodeRequest`) v podobě adresy samotné službě. Vlastní adresa může nabývat podoby několika různých formátů od zadání samotného státu na jedné straně až po názvy parků, ulic, letišť apod. na straně druhé [63], přičemž obsluha služby dokáže sama rozpoznat, který z formátů adresy byl zadán. Při předávání dotazu pomocí metody `Geocoder.geocode()` je nutné implementovat funkci, která se bude starat o výsledek dotazu vrácený službou. Tato funkce musí obsahovat dva parametry připravené pro objekty, jež jsou navraceny jako výsledek dotazu. Jedná se o objekty *results* a *status*. Objekt *status* obsahuje metainformace výsledku dotazu. Jedná se o jednu z možných konstant: `OK`, `ZERO_RESULTS`, `OVER_QUERY_LIMIT`, `REQUEST_DENIED` a `INVALID_REQUEST`. Obsahem objektu *results* je mimo jiné i parametr *location* obsahující výsledné souřadnice ve formátu *LatLng*. V základní bezplatné verzi Google Maps API existuje denní limit stanovený na 2500 požadavků na vyhledání souřadnic dle adresy. V konkrétním případě vytvořené aplikace uživatel zadává název ulice nebo části města, k čemuž je připojen textový řetězec „Pardubice“ a výsledek pak předán jako adresa pro vlastní dotaz. Tím je dosaženo jednak toho, že uživatel nemusí při každém dotazu psát název města a také toho, že v případě překlepu nebo zadání neexistující ulice se jako relevantní výsledek považuje název města a střed mapy se nastaví nad centrum.

Výsledná podoba aplikace resp. její vizuální rozhraní je zobrazeno v příloze (Příloha A). V pravém horním rohu je umístěna legenda obsahující „checkboxy“ pro možnost vypnutí vrstvy. Součástí mapového pole je i posuvník pro změnu přiblížení obsahující ikonu služby StreetView, kterou Google mapy nabízejí. Oproti tradičnímu vzhledu je mezi položky v pravém dolním rohu přidáno grafické měřítko. Výsledná aplikace se tedy skládá z HTML souboru *mapy.html*, který obsahuje vlastní kód stránky včetně vloženého mapového pole, souboru *mapa.js*, který se stará o chování stránky především tedy o zabezpečení funkcí popsaných výše, a podpůrných souborů (obrázky, knihovna pro kontextové menu,...).

4.2 Řešení pomocí ArcGIS API

4.2.1 Popis nástroje ArcGIS API

Platforma ArcGIS od firmy ESRI dlouhodobě patří k nejvyužívanějším nástrojům svého druhu na trhu [68]. Pro vývojáře ArcGIS nabízí mnoho API napříč platformami, nástroji a programovacími jazyky [69]. Rozhraní *ArcGIS API for JavaScript* (dále jen ArcGIS API) je podporovaný nástroj pro vkládání map do obsahu HTML stránek sloužící ke správě chování mapy včetně velkého počtu funkcí typických pro použití v tomto druhu internetových aplikací (geokódování, práce s vrstvami apod.). ArcGIS API se může chlubit podporou nejnovějších webových technologií (HTML 5, CSS 3, ...), webových prohlížečů (Chrome, Firefox, Safari 3+, IE 7+), dalších JavaScript rámců a integrací s nástroji ArcGIS Online a ArcGIS Server [70]. V současné době je dostupná nejnovější verze *ArcGIS API for JavaScript 3.9* [71] s novou podporou CSV vrstev a s využitím nástrojů *Dojo 1.9.1* (viz dále) a *xstyle*. ArcGIS API je hostováno na serveru ArcGIS Online a je volně k dispozici pro nekomerční a výukové účely [72] jednoduše vložením odkazu na externí JavaScript soubor do kódu stránek. Pro použití map neexistují omezující limity. [73]

ArcGIS API pro svou práci využívá nástroje Dojo. Jedná se o nástroj usnadňující práci v JavaScriptu (podobně jako jQuery) poskytováním množství podpůrných funkcí. Tvůrci ArcGIS API využívají nástroje Dojo k zjednodušení procesu vývoje a k zajištění jednotného chování v podporovaných prohlížečích. Využití Dojo odlišuje obslužný JavaScript kód při použití ArcGIS API především použitím funkce `require()`, která slouží pro importování součástí (tříd) API, jež chce vývojář použít. [74]

Existují dva přístupy využití ArcGIS API resp. dva způsoby psaní JavaScript kódu využívajícího ArcGIS API a Dojo. Jedná se o *Asynchronous module definition (AMD)* a *Legacy* přístup. AMD se považuje za novější a jednodušší, proto i současná verze dokumentace je psaná pro AMD, přičemž je možné přepnutí do přístupu *Legacy*. Hlavní rozdíl mezi oběma přístupy je v tom, že starší *Legacy* využívá úplného zápisu v třídní hierarchii, kdežto AMD společně s importováním daných tříd pomocí `require()` funkce umožňuje zápis zkrácený. Oba přístupy je možné kombinovat. Využití plného zápisu v hierarchii tříd může být použito především pro zajištění přehlednosti struktury API u některých použitých funkcí. [74] [75]

Kromě vložení mapy do obsahu stránek – včetně integrovaného intuitivního chování mapy (posun, přibližování,...) – nabízí použití ArcGIS API dle [76] [75] [77] následující funkce a nástroje:

- **importování a práce s externími vrstvami** – ArcGIS API podporuje formáty CSV, KML a služby GeoRSS, WMS a WMTS. Existuje i přímá podpora importování vrstev z OpenStreetMap;
- **prostorové analýzy** – možnost vytváření obalových zón, zónových statistik, výpočtu obsahu obrazců apod. je buď integrovaná přímo v ArcGIS API nebo využívá propojení s ArcGIS Server, který poskytuje výpočtovou kapacitu pro poskytování analýz. U analýz využívajících ArcGIS server mohou existovat omezení vzhledem k tomu, že služba ArcGIS Server je placená;
- **řídící struktury** – součástí API jsou některé přednastavené řídící struktury pro přibližování a posun mapy, dále jsou zde integrovány třídy přímo podporující kompoziční prvky mapy (legenda, měřítko,...);
- **vyhledávač trasy** – ArcGIS API disponuje vyhledávačem tras a integrovaným graficky zpracovaným panelem pro vyhledávání tras. Při asociování panelu s mapou se výsledná trasa zobrazí na mapě včetně jednotlivých zastávek, které je možno po mapě dynamicky přesouvat. Nástroj dává k dispozici i popis trasy včetně možnosti tisku; [76] [78]
- **geokódování** – možnost geokódování ve smyslu vyhledání pozice na mapě dle zadaného místa je pomocí ArcGIS API řešená jako integrovaný panel s defaultním vzhledem a umístěním v levém horním rohu mapy. Není tedy třeba pro účely poskytované služby vytvářet speciální textové pole jako součást HTML kódu stránek. Služba *Geocoder widget* volitelně umožňuje také našeptávání zadávané adresy a vytvoření tzv. „spotlight effect), tedy efektu kdy je mapa vycentrovaná na vyhledaný bod, přičemž okolí je v určitém okruhu zakryto efektem stínu; [79]
- **podpora souřadnicových systémů** – ArcGIS API umožňuje pracovat s velkým množstvím souřadnicových systémů, včetně na území ČR nejpoužívanějších S-JTSK a WGS84 [80]. Definování konkrétního souřadnicového systému přímo vyžadují některé funkce ArcGIS API. Seznam všech použitelných souřadnicových systémů seřazených dle tzv. Well-known ID (WKID) je k dispozici prostřednictvím ArcGIS API reference [77] [81]

- **další funkce** – ArcGIS API je velice bohatý nástroj na podpůrné funkce. Pomocí integrovaných tříd lze tedy kupříkladu také vytvářet samostatné vrstvy s popisky, pracovat s geometrickými prvky (linie, polygony,...) a s grafikou.

4.2.2 Vlastní tvorba aplikace pomocí nástroje ArcGIS API

Tvorba aplikace pomocí ArcGIS API byla provedena za použití přístupu AMD (viz předchozí kapitola) s občasným využitím přístupu Legacy. Pro vložení mapového pole do obsahu stránek je tedy třeba nahrát externí JavaScript s kódem ArcGIS API a externí stylpis. O zobrazení mapy na stránkách, ale zároveň i o nastavení veškerého chování a také všech funkcí využívajících ArcGIS API, se stará Dojo funkce `require()`. Tato funkce vyžaduje dva argumenty, prvním je pole textových řetězců, kdy každý textový řetězec představuje třídu, jež je potřeba importovat a druhým samotná inicializační funkce. Pole řetězců připomíná přístup importování balíčků v jiných programovacích jazycích (po importování daného balíčku není třeba uvádět úplnou cestu v hierarchii tříd k požadované třídě nebo funkci). Inicializační funkce má dynamický počet argumentů, kdy každý argument představuje název třídy, jejíž cesta je uvedena v poli textových řetězců. Pro správné fungování je tedy třeba cesty ke třídám (v poli řetězců) i jejich názvy (argumenty inicializační funkce) uvést všechny a v odpovídajícím pořadí (výjimku tvoří importování *dojo/domReady!*). Možnou podobu funkce `require()` a inicializační funkce ilustruje obrázek (Obrázek 9). Inicializační funkce pak ve svém těle obsahuje stěžení JavaScript kódu stránky. Jakýkoli kód využívající funkce a třídy ArcGIS API musí být obsažen v těle této funkce.

```
require([
  "esri/map", "esri/layers/KMLLayer", "dojo/domReady!"],
function(Map, KMLLayer) {
  var map = new Map("map", {
    center: [15.764, 50.033],
    zoom: 13,
    basemap: "streets"
  });
});
```

Obrázek 9 - Možná podoba inicializační funkce v ArcGIS API

Zdroj: vlastní

Inicializace mapy a nastavení základních parametrů probíhá pomocí objektu třídy *Map*. Koncepce ArcGIS API je taková, že velké množství funkcí je navázáno přímo na objekt třídy *Map* (obsluha událostí, práce s vrstvami,...) [77]. Obrázek výše (Obrázek 9) ilustruje také vytvoření objektu třídy *Map* a nastavení základních parametrů mapy.

Kromě základních parametrů pro inicializaci a zobrazení mapy můžeme na obrázku (Obrázek 9) spatřit také importování třídy *KMLLayer*. Ta je potřebná pro načtení a práci s vrstvami využívajícími jako zdroj dat soubor ve formátu KML. Řešení pomocí ArcGIS API využívá stejných KML souborů dle stejného formátování jako předchozí řešení (4.1.2 Vlastní tvorba aplikace pomocí Google Maps API). Podpora KML formátu byla do ArcGIS API integrována od verze 2.4. Konstruktor třídy *KMLLayer* vyžaduje jako jediný argument URL adresu KML souboru umístěného a volně dostupného na webu. Konstruktor z poskytnutého KML souboru vytvoří objekt třídy *Layer* (tedy obecnou vrstvu), kterou je možné pomocí funkce `addLayer()` přidat do mapy. Podpora práce s KML soubory je v ArcGIS API poměrně rozsáhlá. Třída *KMLLayer* propůjčuje načtení vrstvě metody, které nastavují minimální a maximální rozsah viditelnosti vrstvy, průhlednost a další metody, které usnadňují případnou práci s vrstvou za použití technologie AJAX. Pomocí metody `getLayers()` lze soubor načíst do proměnné a pomocí metod, jež nabízí Dojo, soubor procházet podobně jako při použití jQuery. Přestože Dojo přístup ke struktuře KML souboru nabízí, pro vlastní implementaci bylo využito řešení pomocí jQuery, již dříve vytvořené pro řešení pomocí Google Maps API. Výhodnou využití KML je fakt, že se do aplikace vytvořené pomocí ArcGIS API přenáší již dříve provedené formátování.[77]

ArcGIS API nabízí i třídu pro vytvoření legendy. Ta se stará o to, aby byl pro každou vrstvu v mapě automaticky vygenerován obrázek a popis (samozřejmě pokud je uveden v konstruktoru vrstvy). Vytvořená legenda je pak dynamická, a pokud je tak nastaveno, dokáže zobrazovat například pouze vrstvy viditelné v zobrazeném rozsahu mapy. Legenda nepodporuje všechny vrstvy dostupné v ArcGIS API, KML vrstvy však mezi ně patří. Možnost vytvoření legendy je prospěšné zejména při velkém počtu vrstev přidávaných do mapy, kdy se výrazně projeví výhoda automatického generování. V rámci zachování jednoty vytvořených řešení a především kvůli snazší implementaci možnosti vypínat jednotlivé vrstvy mapy bylo pro legendu použito podobné řešení jako v aplikaci vytvářené pomocí Google Maps API. Odlišné formátování a využití externího stylopisu poskytovaného ArcGIS API zapříčiňuje mírně odlišný vzhled panelu legendy. Ten je stejně jako v ostatních řešeních možné skrývat, přičemž není nijak ovlivněno, jaké vrstvy jsou vybrány pro zobrazení.

Podobně jako u Google Maps API ani ArcGIS API nenabízí třídu přímo určenou pro tvorbu kontextového menu. Řešení ze strany ArcGIS však existuje v podobě využití třídy *dijit/Menu* nástroje Dojo [76]. Ta umožňuje vytvořit kontextové menu pomocí kliknutí pravého tlačítka myši v místě kliknutí. Dále nabízí řadu tříd a funkcí pro přidávání rozličných

prvků do kontextového menu a vytvářet tak pod-menu apod. Pro účely vytvářené aplikace mají význam pouze třídy *dijit/MenuItem* a *dijit/MenuSeparator*. Při vytváření instance třídy *MenuItem* je jako argument vyžadován objekt nesoucí vlastnosti položky menu včetně názvu (tedy textu položky menu) a především názvu funkce obsluhující specifikovanou událost (v případě vytvářené aplikace událost *OnClick*). [82]

Jako součást kontextového menu byla implementována funkce pro vyhledání nejbližšího kontejneru. Řešení kontextového menu pomocí třídy *Menu* umožňuje odečtení souřadnic kliknutí pouze pro účely zobrazení menu, tedy v souřadnicích určujících pozici na obrazovce nikoli v zeměpisných souřadnicích. O získání souřadnic pro zobrazení kontextového menu se stará funkce `getMapPointFromMenuPosition()`. Pro získání výchozích pozic k vyhledávání nejbližšího kontejneru byl zvolen způsob, kdy je uživatel po kliknutí na danou položku menu vyzván k vybrání pozice z mapy. Po potvrzení dialogového okna se cursor mapy změní v zaměřovací kříž a uživatel může kliknutím vybrat, z jakého místa na mapě si přeje vyhledat nejbližší kontejner daného typu. V rámci tohoto řešení je po potvrzení dialogového okna (myšlena JavaScript funkce `alert()`) vytvořen posluchač události kliknutí na mapu. Jako obslužná pak slouží funkce `projectToWebMercator()`, která jako první provede odstranění případných grafických prvků na mapě a odstranění zmíněného posluchače události *onClick* tak, aby uživatel nemohl znovu kliknout na mapu a zadat tak příkaz k vyhledávání nejbližšího kontejneru v průběhu procesu vyhledávání. Po provedení vyhledávání je uživateli stejně jako u řešení pomocí Google Maps API předán název a vzdálenost k nejbližšímu kontejneru prostřednictvím dialogového okna. Současně s tím je do mapy v místě pozice vybrané pro hledání nejbližšího kontejneru umístěna značka a zobrazena linie spojující tuto značku s vyhledaným kontejnerem. Vyhledávání nejbližšího kontejneru probíhá za použití stejného algoritmu jako při tvorbě aplikace pomocí Google Maps API. Jak již bylo zmíněno výše, k přístupu do struktury KML dokumentu bylo opět použito metod nástroje jQuery a pomocí funkcí `find()` a `each()` v cyklu byly postupně získány koordináty každého jednotlivého kontejneru daného typu. Pro správné fungování tohoto postupu v ArcGIS API je třeba nejdříve definovat správný souřadnicový systém, protože jak bylo zmíněno ArcGIS API dokáže pracovat s velkým množstvím souřadnicových systémů. V tomto případě se jedná o souřadnicový systém WGS84 který předáme proměnné pomocí konstruktoru třídy *SpatialReference* a kódového označení tohoto souřadnicového systému: 4326 [77][81]. Získání vlastní vzdálenosti mezi zadaným bodem a kontejnerem (resp. postupně v cyklu každým kontejnerem) je pomocí nástroje ArcGIS API třeba řešit

jiným způsobem, než při použití Google Maps API, protože ArcGIS API nenabízí metodu, která by přímo počítala vzdálenost dvou bodů při zadání těchto bodů jako argumentů funkce. Od verze 2.0 třídy *GeometryService* [77] existuje konstrukce počítající vzdálenost dvou geometrických objektů (nemusí se jednat pouze o body) pomocí funkce `distance()`. Ta však vyžaduje konstrukci za použití polí a dalších nastavení argumentů funkce [76]. ArcGIS API však také ve třídě *geodesicUtils* obsahuje metodu `geodesicLengths()`, která počítá délku linie v zadaných jednotkách [77]. Protože pro zobrazení spojnice mezi zadaným bodem a vyhledaným kontejnerem byla vytvořena linie, zdá se jako jednodušší řešení použití funkce měřící délku této linie jako vzdálenost daných bodů.

Jak bylo uvedeno v popisu nástroje, ArcGIS API disponuje možností pro vyhledávání trasy. K tomuto účelu slouží speciální panel (widget) tzv. „*Directions widget*“. O jeho zobrazení, nastavení a správu veškerých jeho vlastností se stará třída *Directions* [77], která disponuje velkým počtem obslužných metod a vlastností. Vyhledávač trasy používá nástroj „*network analysis services*“, který je poskytován prostřednictvím ArcGIS Online [83]. Tento nástroj je možné využít nejenom k vyhledávání trasy, ale také k jiným síťovým analýzám, analýzám dostupnosti atp. Podpora ze strany ArcGIS API je však zaměřena především na prosté vyhledávání trasy a práci s ní pomocí obslužného panelu a je pouze složitě implementovatelná pro použití k vyhledání trasy mezi dvěma zadanými souřadnicemi. ArcGIS API také nabízí řešení k přímému vyhledání nejbližšího bodu, který bere v úvahu délku trasy. Nejedná se tedy o vyhledání trasy k již nalezenému nejbližšímu bodu, ale opravdu o najetí nejkratší trasy. Nástroj sloužící k tomuto účelu se jmenuje *ClosestFacilityTask* a umožňuje také nastavit, kolik míst si uživatel přeje vyhledat a řadu dalších parametrů, jako je například stanovení nejzazší hranice pro vyhledávání [76]. Stejně jako v případě Google Maps API je však použití takového řešení pro větší počet bodů (v řádu stovek) natolik výpočetně náročné, že toto řešení není vhodné pro vytvářenou aplikaci. Vyhledání trasy od zadaného místa k nejbližšímu kontejneru i použití nástroje (nástroj) k nalezení nejkratší trasy bylo odzkoušeno. Pro velkou časovou prodlevu mezi zadáním a ukončením procesu vyhledávání v obou případech, pro dosažení neuspokojivých výsledků v prostředí Pardubic a vzhledem k okolnostem uvedeným dále, nebyla tato funkce do aplikace vytvořené pomocí ArcGIS API implementována. Byla zvažována možnost vyhledání menšího počtu nejbližších kontejnerů (vzdušnou čarou), na které by byl posléze aplikován nástroj *ClosestFacilityTask*. Tento postup byl myšlenkově odzkoušen na modelovém případě, kdy byl jako výchozí vybrán bod na souřadnicích 50°2'21.607"N, 15°45'36.691"E tedy na pravém

břehu Labe severně od nábřeží Závodu míru. V tomto případě je nejbližší kontejner (všech druhů) ve vzdálenosti 136 metrů na adrese „Závodu Míru - čp. 1858“ tedy na opačném břehu Labe. V tomto modelovém případě by nejbližším dostupným kontejnerem však byl kontejner na adrese „Polabiny IV - Labský palouk 495“. Algoritmus, který by vyhledával předem určené množství nejbližších kontejnerů a poté na ně aplikoval funkci *ClosestFacilityTask* by byl stále příliš výpočetně náročný, protože kontejnerů, které se vzdušnou čarou nacházejí blíže než nejbližší dostupný kontejner je více jak deset. Problém vyhledávání nejbližší trasy při řešení pomocí ArcGIS API také naráží na relativně malou podrobnost podkladové mapy poskytované ArcGIS API.

Nástroj pro vyhledávání míst na mapě v ArcGIS API nazývaný „Geocoder widget“ umožňuje do mapy umístit textové pole sloužící k zadávání adresy pro vyhledání. Jak bylo uvedeno výše, díky tomuto řešení odpadá nutnost do stránek umístit samostatné HTML textové pole a pomocí tlačítka přenášet zadanou hodnotu ke zpracování pomocí JavaScriptu. Přesto je třeba pro nástroj vyhledávání v HTML kódu stránky vytvořit `<DIV>` element. ID tohoto elementu poté přebírá funkce `dojo.byId()` ekvivalentní k Javascriptové funkci `getElementById()`. Dále je také třeba umístit do kódu stránky stylpis upřesňující vzhled objektu [79]. Subjekt Geocoder widget je implementován pomocí třídy Geocoder, která ve svém konstruktoru vyžaduje soubor parametrů a příslušný HTML element předávaný právě zmíněnou funkcí `dojo.byId()`. V konkrétním případě vytvořené aplikace jsou konstruktoru předávány parametry, jejichž součástí je kromě nastavení mapového objektu, ve kterém bude vyhledávání probíhat, také parametr *arcgisGeocoder*. Parametr *arcgisGeocoder* může nabývat buď logických hodnot *true/false* (v praxi se používá hodnota *false*, pokud je použit jiný geokódovací nástroj než primárně využívaný *Esri World Locator* [77]) nebo hodnoty ve tvaru objektu reprezentující konfiguraci *Esri World Locator*. Mezi tyto konfigurační hodnoty patří i parametry *placeholder* a *suffix*. Parametr *placeholder* slouží pouze k zadání textového řetězce, který se objeví ve vyhledávacím textovém poli s výzvou: „Zadejte ulici nebo čtvrt“, parametr tedy funguje podobně jako hodnota atributu *value* HTML formulářového prvku `<input>` [84]. Větší význam má hodnota parametru *suffix*. Ta udává textovou hodnotu, jež má být připojena za zadaný textový řetězec. V konkrétním případě vytvořené aplikace je to hodnota „Pardubice“, která zajišťuje vyhledávání ulice nebo čtvrti v rámci Pardubic. Bez zadání této hodnoty celá funkce funguje nesprávně a služba neposkytuje kýžené výsledky.

Výsledný vzhled i chování aplikace vytvořené pomocí ArcGIS API jsou velice podobné jako u aplikace vytvořené pomocí Google Maps API, což je logickým důsledkem

faktu, že oba nástroje poskytují stejné nebo podobné funkce. Vizualní rozhraní aplikace vytvořené pomocí ArcGIS API je možné vidět v příloze (Příloha B).

Legenda s možností jejího skrytí a možností vypínání vrstev kontejnerů dle typu je umístěna v pravém horním rohu mapového pole. Součástí mapového pole jsou také tlačítka pro přiblížení a oddálení. Na rozdíl od řešení pomocí Google Maps API je textové pole pro vyhledávání lokality umístěno v pravém horním rohu mapového pole. Při použití pozicování pomocí kaskádových stylů by bylo možné docílit, aby se toto textové pole nacházelo nad mapou [85], byla však zvolena tato varianta pro lepší vizuální dojem a shodné umístění v modelových příkladech pro vyhledávání [76]. Dalším prvkem umístěným do mapového pole je grafické měřítko, pro něž je v ArcGIS API k dispozici třída *Scalebar* umožňující nastavení umístění, jednotek a vzhledu. Při nastavení jednotek na metry (pomocí hodnoty „*metric*“ parametru *scalebarUnit*) se dynamicky mění nejen velikost měřítka při přibližování a oddalování, ale také jednotky z kilometrů na metry a naopak.

Pro lepší práci se soubory byl tentokrát zvolen způsob umístění JavaScript kódu interně do HTML kódu stránky. Aplikace se tedy skládá pouze ze souboru *mapyARC.html*, přičemž je opět třeba, aby byly na serveru umístěny i KML soubory a využívané obrázky.

4.3 Řešení pomocí Mapy API

4.3.1 Popis nástroje Mapy API

Nejpoužívanější český mapový portál Mapy.cz (www.mapy.cz) [24][86] také nabízí JavaScript API sloužící k vložení mapy do obsahu webových stránek a podporující základní práci s mapou. Mapy.cz je produkt společnosti Seznam.cz a.s. Aplikační rozhraní nazvané Mapy API (někdy též SMap API) je k dispozici od roku 2007, kdy bylo spuštěno pro nekomerční využívání s omezením na 1000 zobrazení mapy za den [87]. V roce 2011 dosáhla společnost Seznam.cz dohody s poskytovatelem mapových zdrojů a upravila licenční ujednání v tom smyslu, aby využití mapového API mohlo sloužit také aplikacím určeným pro komerční účely [88]. Tímto krokem nejen že oproti jiným poskytovatelům mapových API získala společnost konkurenční výhodu, ale také zahájila bouřlivý vývoj nových verzí API s cílem rozšířit možnosti použití API a být konkurentem rozsáhlejšího Google Maps API [89]. Zajímavostí také je, že od verze 4.2 začali tvůrci jednotlivé verze API pojmenovávat podle slavných cestovatelů [90]. Postupně tedy bylo možné využívat například zpětně pojmenovanou verze 4.0 – Marco Polo, 4.1 – Kryštof Kolumbus nebo současně (duben 2014)

nejnovější verzi 4.8 Hanzelka a Zikmund [91]. Přestože Mapy API co do bohatosti funkcí na první pohled stále nemůže soupeřit s Google Maps API nebo ArcGIS API, vývoj posledních verzí naznačuje, že ještě před pár lety v porovnání s konkurencí zatracované aplikační rozhraní [89] nechtějí vývojáři nechat odsunout na vedlejší kolej a nejen, že ho stále udržují při životě, ale vhodně volenou implementací funkcí a redukcí omezení používání se snaží pozvednout kvalitu produktu na úroveň srovnatelnou s konkurencí.

V současné době (oproti starším verzím API) Mapy API umožňuje použít všechny mapové podklady, které jsou k dispozici na Mapy.cz s výjimkou podrobné mapy Evropy [52]. Mapy API nevyužívá žádných externích JavaScriptových konstrukcí jako jsou Dojo nebo jQuery (pouze v některých případech se spoléhá na framework JAK [92], čemuž se dá vyhnout) a vývojář/uživatel pracující s Mapy API musí pouze importovat externí JavaScript knihovnu Mapy API příslušné verze [93]. Mapy API nabízí kompletní dokumentaci poskytovaných funkcí v češtině (názvy funkcí jsou v angličtině). Celkově stejně jako u portálu Mapy.cz je chování vývojářů uzpůsobeno faktu, že firma Seznam.cz se zaměřuje pouze na český trh [94]. Omezujícím faktorem je také slabá integrace dalších navazujících nástrojů daná faktem, že Mapy API není součástí větší rodiny aplikačních rozhraní (Seznam.cz poskytuje kromě Mapy API pouze CAPTCHA API [95]) jako je tomu u Google Maps API (Google Developers) nebo ArcGIS API (ArcGis Developers).

Stejně jako ostatní API mapových serverů i Mapy API nabízí kromě vložení mapy vzhledově odpovídající Mapy.cz do obsahu uživatelových stránek i řadu dalších funkcí. Jsou to dle [96] a [97] následující:

- **základní práce s mapou a podkladovými vrstvami** – Mapy API nabízí stejné spektrum podkladových mapových vrstev jako na portále Mapy.cz, přičemž kolik, jaké a jakým způsobem budou do aplikace implementovány, záleží na vývojáři/uživateli [96]. Základní poskytované mapy jsou: Obecná, Turistická, Historická (1836-1852), Letecká, Letecká (2002-2003) a Letecká (2004-2006). Mapy API také umožňuje prostřednictvím WMS služby importovat dlaždicovou vrstvu od externího poskytovatele (například z ČÚZK), přičemž lze použít Mercatorovu nebo UTM33 projekci (defaultně UTM33) [96];
- **práce se značkami a geometrickými prvky** – s ohledem na primární účely využívání Mapy API, tedy umístování mapových náhledů například s vyznačenými pozicemi sídel firem a podobně, je již od prvních verzí

implementována řada funkcí podporujících umístování a práci se značkami [96]. Dále je také možno do mapy umístovat takřka libovolné vektorové konstrukce (linie, křivky, polygony) za využití třídy *Geometry*. Konstrukce mohou mít i složitější strukturu, k čemuž lze využít buď externí XML soubor, nebo tzv. obecnou posloupnost vektorových primitiv *path* [98]; [96]

- **importování a práce s externími vrstvami** – primárně je pro import dat podporován formát GPX, ale je možné importovat také data ve formátu KML (oboje od verze 4.0 – Marco Polo). Vzhledem k tomu, že oba dva formáty patří do standardu XML, je při jejich importování vyžadován právě XML formát daného dokumentu (tedy soubor s příponou *.xml*) [96] [97]. Mapy API nenabízí žádné další funkce pro práci s KML a GPX vrstvami;
- **geokódování** – dopředné i zpětné geokódování bylo do Mapy API implementováno až s verzí 4.4 – David Livingstone [91]. Aby geokódování správně fungovalo, je nutné, aby uživatelův webový prohlížeč podporoval techniku „cross-origin resource sharing“ (CORS) [99]. Služba geokódování pro své fungování potřebuje explicitně vytvořené vstupní textové pole pomocí HTML; [96]
- **vyhledávač trasy** – hledání nebo plánování trasy mezi dvěma body a až třemi průjezdnými body je v Mapy API k dispozici od verze 4.7 – Reinhold Messner [91]. Stejně jako je tomu na portále Mapy.cz i prostřednictvím Mapy API je možná volit vyhledávání trasy pro jízdu autem, na kole nebo chůzi pěšky. Mapy API však na rozdíl od Mapy.cz neposkytuje funkci ručního měření; [97]
- **další funkce** – okrajově lze pomocí Mapy API upravovat i další funkcionality jako jsou řídicí struktury (přibližování, posuvník, výběr podkladové mapy ap.), práce s kontextovým menu a úpravu souřadnicového systému. Defaultně Mapy API používá vnitřní souřadnicový systém nazývaný PP. Dále používají systémy WGS84, S-JTSK a UTM33. [97]

4.3.2 Vlastní tvorba aplikace pomocí Mapy API

Přístup tvůrců Mapy API naznačuje fakt, že je toto programovací rozhraní určeno primárně pro tvorbu jednodušších aplikací, především pro prosté vložení mapy do obsahu stránek, nastavení centra mapy na dané souřadnice popřípadě vložení značky do mapy (první verze API toho více ani neumožňovaly [87]).

Stejně jako v případě obou předchozích API i Mapy API vyžaduje pro svou práci vložení externí JavaScriptové knihovny. Prvním řádkem kódu pak musí být funkce `Loader.load()`, která se stará o načtení tříd API [100]. Dále musí následovat HTML kód obsahující vložení mapového pole pomocí tagu `<DIV>`, který jako svoje ID má uveden název objektu mapy, dále používaného v kódu. Z toho vyplývá, že tento tag musí předcházet vlastnímu obslužnému JavaScript kódu, jinak se mapa nenačte. Vložení mapy probíhá jednoduše vytvořením objektu třídy *SMap*, jejíž konstruktor vyžaduje jako parametr kontejner, do něhož bude mapa umístěna, tedy výše zmíněný HTML `<DIV>` tag, volitelně pak nastavení výchozího vycentrování mapy, konfigurační objekt, minimální a maximální přiblížení, natočení mapy a dobu trvání animace přiblížení [97]. V případě vytvářené aplikace je mapový objekt nazvaný „mapa“ umístěn do stejně identifikovaného tagu, střed nastaven na koordináty Pardubic a výchozí úroveň přiblížení na hodnotu 10. Dále jsou vloženy defaultní ovládací nástroje a přidána defaultní podkladová mapová vrstva pomocí funkcí `addDefaultControls()` a `addDefaultLayer()`. K zajištění, aby mapové pole dynamicky zabíralo plnou šířku (a procentně volitelnou výšku) stránky pro všechna rozlišení nestačí jako v předchozích případech pomocí kaskádových stylů nastavit šířku obalového tagu na 100%. Tag sloužící jako kontejner pro mapový objekt totiž vyžaduje pevně nastavenou výšku a šířku. Jedním z možných řešení je nastavení relativní velikosti obalovému tagu, zjištění této velikosti v pixelech (například pomocí funkcí `width()` a `height()` poskytovaných jQuery) a nastavení velikosti kontejneru na zjištěné hodnoty. Mapy API však nabízí řešení dynamické velikosti mapy, i když je třeba ho explicitně uvést v kódu. K tomuto účelu slouží třída *Sync* umožňující také volitelné nastavení vzdálenosti od spodního okraje [97].

Jak bylo uvedeno v popisu nástroje, Mapy API umí nahrávat GPX a KML soubory. Děje se tak prostřednictvím požadavku na server pomocí technologie AJAX [96]. V případě vytvářené aplikace je to provedeno za pomoci rozhraní JAK, kdy je vytvořen objekt, jemuž je předána adresa souboru umístěného na server a funkce, která se stará o obsluhu získaného souboru. Pomocí konstruktoru třídy *Layer.KML* je poté vytvořena KML vrstva. Vrstva je přidána do mapy pomocí funkce `addLayer()` a zobrazena pomocí funkce `enable()` [97]. Problém ale je, že zpracování dotazu očekává XML soubor nikoli KML soubor. Proto je třeba použitý KML soubor změnit na XML. Protože KML je pouze speciálním případem XML, lze toho dosáhnout bez využití konvertorů pouze přepsáním přípony `.kml` na `.xml`. Mapy API však nenabízí žádné další funkce pro práci s KML vrstvou pouze funkce pro práci s obecnou

vrstvou umožňující upravovat její zviditelňování, překreslování apod. Takto importovaná KML vrstva se chová v podstatě shodně jako v aplikacích vytvořených pomocí Google Maps API a ArcGIS API. Výjimkou je pouze ignorování tagu `<scale>`, který nastavuje procentní velikost zobrazované ikony vůči původnímu obrázku. Veškeré hodnoty jsou tedy brány jako 1, což představuje sto procent původní velikosti obrázku. Pro dosažení přívětivějšího vzhledu bylo tedy v tomto případě nutno nahradit zdrojový obrázek obrázkem menší velikosti. Pokud by stejná situace nastala v jednom ze dvou předchozích použitých nástrojů (Google maps API, AcGIS API), nastal by problém, protože obě řešení využívají stejný zdrojový KML soubor. Jak však bylo zmíněno výše, problémy s přístupem ke KML v řešení pomocí Mapy API vyžaduje použití vlastního XML souboru, takže změna velikosti původního obrázku předchozí řešení nijak neovlivní. Oproti srovnání s předchozími řešeními jsou zobrazené body v prostředí mapy poskytované Mapy API o několik málo pixelů posunuty (v měřítku mapy cca 2 metry). Důvod této odchylky se nepodařilo zjistit. Přestože díky vysoké podrobnosti map (některé kontejnery jsou vykresleny do silnice) může tento fakt působit rušivě, na funkčnost aplikace vliv nemá. Volitelné zobrazování vrstev dle typu kontejneru (opět prostřednictvím panelu legendy) je řešeno pomocí metod `enable()` a `disable()`, které nabízí třída obecné vrstva představovaná třídou *Layer*. Legenda samotná má stejnou podobu jako v obou předchozích řešeních. Nicméně velice omezené možnosti pro práci s řídicími prvky (posouvání, přibližování, změna podkladové mapy) znemožňují zcela volitelné umístění těchto prvků. Volné pozicování pomocí kaskádových stylů je možné pouze u vytvořených kontrolních prvků, přičemž nelze nikdy dosáhnout plně stejného chování a vzhledu jako u přednastavených struktur do mapy přidávaných pomocí metody `addDefaultControls()`. Vzhledem k tomu byla oproti předchozím řešením legenda umístěna vlevo a to pod panel pro změnu podkladové mapy. Legenda opět disponuje možností jejího skrytí, v případě, že by její umístění uživatele rušilo.

Na rozdíl od předchozích dvou API, nástroj Mapy API nabízí třídu přímo určenou pro tvorbu kontextového menu. Tato třída je ve struktuře API zařazena jako potomek obecné třídy *Control* seskupující řídicí prvky a nese název *ContextMenu* [97]. Samotní tvůrci ve zveřejněných příkladech uvádějí, že v budoucnu se plánuje lepší konfigurovatelnost kontextové nabídky a v uvedeném příkladu dávají k dispozici jen předem vytvořené defaultní kontextové menu obsahující pouze souřadnice kliknutí v systému WGS84 a možnost přiblížení a oddálení mapy [96]. Vytvořit na položky bohatší kontextovou nabídku reagující na libovolné podporované události je však možné [104]. Mapy API nabízí kromě výše

zmíněné třídy *ContextMenu* i další třídy pro práci s kontextovou nabídkou. Z nabízených tříd umožňujících také práci s koordinátami kliknutí a s přibližováním a oddalováním mapy byly použity třídy *Item* a *Separator* (obě jsou potomky třídy *ContextMenu*). Třída *Separator* slouží pouze k vytvoření objektu oddělovače, který je posléze metodou `addItem()` přidán do nabídky, a nenabízí žádné další možnosti formátování oddělovače [97]. Konstruktor třídy *Item* pak slouží k vytvoření objektu položky menu. Na objekt třídy *Item* lze poté navázat událost `click` (resp. se jedná přímo o metodu třídy *Item* [97]) spouštěnou při kliknutí na příslušnou položku menu. Vytvořenou položku je stejně jako oddělovač třeba přidat do objektu kontextové nabídky pomocí metody `addItem()`, jež je součástí třídy *ContextMenu* a vyžaduje jako argument objekt položky (popřípadě separátoru) a volitelně pozici na kterou se má položka v menu umístit. Při nezadání pozice se položka automaticky umístí na konec. Kontextovou nabídku, která se po kliknutí pravým tlačítkem myši zobrazí nad mapou v místě kliknutí je třeba po vybrání položky explicitně odstranit. To se provede pomocí metody `close()` parametru menu předávaného jako argument funkce spouštěné při kliknutí na položku. To, aby každá položka menu v obsluze své události měla přístup k souřadnicím kliknutí, zajišťuje metoda `setCoords` zděděná po třídě *Item* [104]. Získání samotných souřadnic probíhá konstrukcí `SMap.Coords.fromWGS84(this._coords.toWGS84(0))`.

Položkami kontextového menu, podobně jako u předchozích dvou řešení, jsou funkce vyhledání nejbližšího kontejneru dle typu a vyhledání trasy k nejbližšímu kontejneru dle typu. Vyhledání nejbližšího kontejneru probíhá podle stejného algoritmu jako v předchozích řešeních. Pomocí technologie jQuery je proveden dotaz na server a vrácen KML soubor (v tomto případě nejsou využity žádné třídy Mapy API, které by vyžadovaly volat XML soubor na místo KML souboru). Dále jsou opět v cyklu zjištěny koordináty všech kontejnerů a výpočtem vzdálenosti mezi souřadnicemi kontejneru a souřadnicemi kliknutí zjištěna vzdálenost a nalezena ta nejmenší. Informace o nalezeném kontejneru a vypočítaná nejmenší vzdálenost jsou uživateli sděleny skrze dialogové okno. Dále je na mapě zobrazen bod kliknutí a bod, kde se nalézá nejbližší kontejner a mezi nimi zobrazena linie spojující tyto body. Stejně jako ArcGIS API, ani Mapy API nenabízí metodu pro zjištění vzdálenosti dvou bodů. Na rozdíl od ArcGIS API neumožňuje ani žádnou zástupnou konstrukci a je tedy třeba pro výpočet vytvořit konstrukci vlastní. Navíc je třeba počítat s faktem, že Mapy API používají svůj vnitřní souřadnicový systém PP. Postupy jak tento problém řešit oficiálně MapyAPI neposkytuje, ale tento problém je řešen na fóru pro vývojáře používající Mapy API (<http://napoveda.seznam.cz/forum>). Ve vlákně fóra, které se věnuje této problematice, se

uvádí jako možnost přepočtu vzdálenosti mezi PP systémem a metry použití metody *ppToMeter*, ta však v dokumentaci nebyla nalezena, jedná se tedy buď o omyl, nebo tato metoda byla z novějších verzí vyřazena. Funkce, která se stará o výpočet vzdálenosti mezi dvěma body v metrech, byla pojmenována `getDist()` a vyžaduje jako argumenty souřadnice obou bodů ve formátu PP koordinátů. Pro převod ze souřadnic ve formátu WGS84 do souřadnic ve formátu PP slouží metoda `toPP()` třídy *Coords* [97]. K výpočtu vzdálenosti mezi body metoda používá euklidovská vzdálenost dle vzorce:

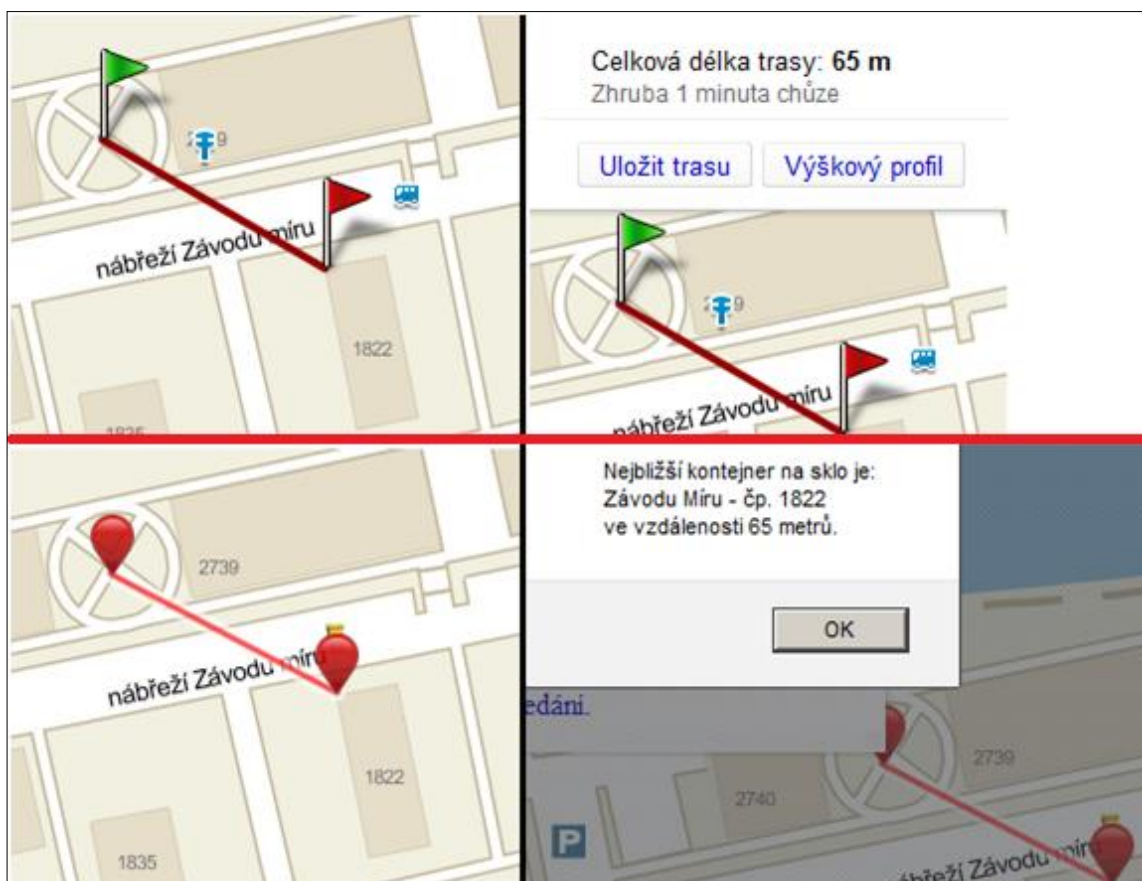
$$D_E = ((x_1, x_2), (y_1, y_2)) = \sqrt{(x_1 - y_1)^2 + (x_2 - y_2)^2} \quad (1)$$

kde D_E je vzdálenost bodů daných souřadnicemi x_1 a x_2 pro bod x a y_1 a y_2 pro bod y [6 s. 31]. Ze vzorce vyplývá, že na pořadí bodů předaných jako argumenty funkce nezáleží. Souřadnice ve formátu PP jsou velká celá čísla (řádově stovky milionů). Vzdálenost vypočítaná dle vzorce vyjde tedy v hodnotách PP. Při absenci metody *ppToMeter* je tedy převodní vztah mezi získanou vzdáleností v hodnotách PP a metry neznámý. Proto byla za pomoci nástroje měření vzdálenosti na portále *Mapy.cz* experimentálně zjištěna převodní konstanta. Převodní vztah se tedy řídí vzorcem:

$$D_{[m]} = D_{[PP]}/k \quad (2)$$

kde $D_{[m]}$ je výsledná vzdálenost v metrech, $D_{[PP]}$ je vzdálenost získaná pomocí vztahu (1) dosazením příslušných souřadnic a k je experimentálně zjištěná tzv. bulharská konstanta stanovená na hodnotu 32 (viz dále). Při neznalosti vnitřního vztahu pro převod souřadnic se toto řešení jeví jako nejpřesnější. Experimentálně bylo při použití $k = 32$ dosaženo průměrné odchylky 0,7 m při maximální chybě 2 m a směrodatné odchylce 0,78 (měření provedeno na vzdálenostech od 10m do 5km). Takto nízká velikost odchylky je dostačující pro správný běh algoritmu vyhledávání nejbližšího kontejneru a naznačuje, že vnitřní převodní konstanta nabývá právě této hodnoty. O konstantě nabývající hodnoty 32 se také zmiňuje ve své bakalářské práci Lukáš Jelínek [102], kde ovšem nejsou uvedeny žádné zdroje ani způsob výpočtu této konstanty. Použití hodnoty konstanty ze zmíněného zdroje by tak bez uvedených referencí bylo třeba experimentálně ověřit stejným způsobem, jako došlo k jejímu stanovení. Způsob experimentálního zjišťování hodnoty konstanty ilustruje obrázek (Obrázek 10). Obrázek ukazuje porovnání naměřené vzdálenosti pomocí nástroje *Měření trasy* v *Mapy.cz* (horní polovina obrázku) oproti vypočtené vzdálenosti pomocí nástroje *Vyhledej nejbližší kontejner na sklo* ve vytvořené aplikaci (spodní polovina obrázku). Na obrázku je zobrazeno

pouze jedno ilustrační měření, samotné zjišťování proběhlo na vzorku 10 měření se vzdáleností od 10m do 5km, při volitelném umístění obou. Naměřené hodnoty a výpočet včetně podrobnějšího popisu metodiky je uveden v příloženém souboru MS Excel (Příloha D). Stejná odchylka při různých vzdálenostech tak potvrzuje, že převodní vztah je lineární dle vztahu (2).



Obrázek 10 – Způsob experimentálního zjišťování převodní konstanty k

Zdroj: vlastní

Funkce `getDist()` výsledný výpočet ještě zaokrouhluje na celé metry. Výslednou podobu funkce `getDist()` ilustruje obrázek níže (Obrázek 11), kdy souřadnice x_1 a x_2 pro bod x a y_1 a y_2 pro bod y ze vztahu (1) představují a_1 , a_2 , b_1 , a b_2 .

```

function getDist(coord1, coord2){
    var dist=0;
    var a1, a2, b1, b2, a, b;

    a1 = coord1.toString().substring(0,coord1.toString().indexOf(","));
    a2 = coord1.toString().substring(coord1.toString().indexOf(",")+1,
        coord1.toString().length);

    b1 = coord2.toString().substring(0,coord2.toString().indexOf(","));
    b2 = coord2.toString().substring(coord2.toString().indexOf(",")+1,
        coord2.toString().length);

    a=parseInt(a1)-parseInt(b1);
    b=parseInt(a2)-parseInt(b2);

    dist= Math.sqrt((a*a)+(b*b));
    dist= Math.round(dist/32);
    return dist;
}

```

Obrázek 11 - Funkce pro zjištění vzdálenosti dvou bodů v Mapy API

Zdroj: vlastní

Nutno zmínit, že existují i alternativní možnosti výpočtu vzdálenosti dvou bodů, přičemž by bylo možné využití přímo souřadnic ve formátu WGS84 a to například využitím Haversinova vzorce [101], ale při vzdálenostech v rámci zájmového území (obecně do 20km) je způsob výpočtu pomocí rovinné euklidovské metriky dostatečně přesný [103]. V rámci zájmového území je možné použít libovolnou metodu jak pro souřadnice WGS84 tak pro souřadnice formátu PP, který je však dle informací od vývojářů uvedených na fóru (viz. výše) vhodnější.

Další položkou kontextového menu je možnost vyhledávání trasy k nejbližšímu kontejneru. Omezení této funkce je stejné jako u řešení pomocí Google Maps API, tedy že vyhledání trasy probíhá k nejbližšímu kontejneru nikoli ke kontejneru nejbližší dostupnému. Plánovačem trasy je v Mapy API třída *Route*, která se stará o volání požadavku na vyhledání trasy [97]. Konstruktoru třídy *Route* se předává pole zeměpisných souřadnic, které obsahuje minimálně dva prvky (start a cíl), volitelně až pět prvků (včetně třech průjezdných bodů), dále název funkce, která se vykoná po nalezení trasy (tzv. callback), a nakonec volitelně objekt nastavujících parametrů obsahující i nastavení druhu trasy (pro pěší, cyklisty, auto,...) [96]. V obslužné funkci je zadáno přesunutí středu mapy nad vyhledanou trasu a to pomocí funkce *computeCenterZoom()*, která vypočítá střed ze zadané množiny bodů včetně odpovídajícího přiblížení [97]. Stejně jako v předchozích řešeních je tato funkce omezena

podrobností s jakou plánovač trasy dokáže trasu vyhledat. Obzvláště se tento fakt projevuje u menších vzdáleností a nastavení režimu hledání trasy pro chůzi.

Jak bylo zmíněno v popisu nástroje, i Mapy API umožňuje vyhledávání lokality dle adresy (geokódování). K jejímu využívání je potřeba webový prohlížeč podporující techniku CORS (v současnosti ji podporuje přibližně 90% používaných prohlížečů, plně ji podporují Firefox, Chrome, Safari, Opera a IE10 a 11 [105]). Dále je třeba do HTML obsahu stránky umístit vstupní textové pole, stejně jako je to u řešení pomocí Google Maps API. Geokódování probíhá pomocí konstruktoru třídy *Geocoder*, který jako argument vyžaduje dotaz ve formě textového řetězce (ten je poskytnut jako hodnota vstupního textového pole) a název funkce, která se vykoná po obdržení odpovědi. K dotazu je opět potřeba připojit textový řetězec „Pardubice“. Funkce zpracovávající odpověď ve svém těle testuje druh vrácené odpovědi a v případě že *Geocoder* nevrátí nic (resp. pole *Results* je prázdné), vypíše hlášku: „Nenalezeno!“. To se stane například v případě, kdy uživatel zadá nesmyslný název ulice nebo v případě že ulice či čtvrť v Pardubicích neexistuje.

Chování a vzhled aplikace vytvořené pomocí Mapy API odpovídá předchozím dvěma řešením. Rozdílem je umístění panelu legendy, což je zdůvodněno výše. V levém dolním rohu je umístěno dynamicky se měnící měřítko. Textové pole pro vyhledávání je umístěno nad mapou. Je také možno měnit podkladovou mapu pomocí kontrolního panelu, přičemž byly dány k dispozici relevantní podkladové mapy. Vizuální podoba aplikace je zobrazena v příloze (Příloha C) JavaScript kód byl v tomto řešení umístěn do HTML kódu stránky ze stejných důvodů jako je tomu u řešení pomocí ArcGIS API. Vlastní aplikace se tedy skládá ze souboru pojmenovaného *mapamapy.html* a dalších souborů včetně XML souborů s vrstvami reprezentujícími kontejnery.

5. VÝBĚR NEJLEPŠÍHO ŘEŠENÍ

5.1 Stanovení a popis hodnotících kritérií

Hodnocení informačních systémů, softwarových produktů, internetových aplikací a různých řešení v oblasti IT obecně, je součástí mnoha článků a odborných prací. Přesto konkrétně v oblasti hodnocení kvality web GIS aplikací neexistuje mnoho prací [43 s. 56]. V případě této diplomové práce se nejedná o porovnání (resp. nikoli pouze) výsledných vytvořených aplikací, ale dle zadání a cílů práce především o zhodnocení, které z nástrojů je nejvhodnější pro tvorbu podobného druhu aplikací. Způsob hodnocení, výběru a především stanovení kritérií je tomuto faktu uzpůsobeno. Proto například hodnocení použitelnosti [43 s. 57] jako představitel hodnotícího kritéria webových GIS není pro hodnocení použitých nástrojů vhodné, neboť všechny tři řešení dodržují stejná pravidla pro vzhled a chování. V procesu hodnocení a výběru nejvhodnějšího nástroje bylo využito zejména informací z publikace *Kvalita webových geografických informačních systémů* od J. Komárkové [43] a informací z portálu *Rozhodovací procesy* (www.rozhodovacicprocesy.cz). Hodnocení vhodnosti použití API mapových portálů se například věnuje článek „*Které mapy jsou pro váš web nejlepší?*“ [106]. Lze zde nalézt inspiraci pro výběr hodnotících kritérií, nicméně porovnání nástrojů je provedeno spíše populární nežli vědeckou formou a bylo provedeno již v roce 2007, takže zcela neodpovídá současné (2014) realitě.

Při stanovování kritérií pro hodnocení jednotlivých řešení jsou brány v potaz především stanovené základní a rozšířené požadavky na podobu a funkce aplikace (kapitola 3.5 Stanovení požadavků na podobu a funkce aplikace). Pro zajištění srovnatelnosti jednotlivých řešení se také předpokládá, že žádný z nástrojů nebyl zvýhodněn nenulovou počáteční znalostí ze strany autora práce. Předpokládá se pouze dobrá počáteční znalost HTML a JavaScript, jakožto technologií potřebných pro tvorbu aplikace pomocí všech třech zvolených nástrojů. Dobrá znalost jednoho z nástrojů oproti ostatním by ho zvýhodňovala, protože by autor mohl nejen rychleji pracovat například díky znalosti API, ale také rychleji implementovat dané požadavky díky znalosti konstrukčních možností a limitů konkrétního nástroje. Tento fakt byl zohledněn i při výběru nástrojů a tato podmínka tedy byla splněna.

S ohledem na daný problém a požadavky na podobu a funkce aplikace byla stanovena tato hodnotící kritéria:

- základní požadavky

- přepínání vrstev
- vyhledání nejbližšího kontejneru
- vyhledání trasy k nejbližšímu kontejneru
- vyhledání lokality
- čas potřebný pro tvorbu aplikace
- rychlost
- podkladové mapy
- bohatost nástrojů API
- zpracování a podpora API
- licenční podmínky

Jednotlivá kritéria podrobněji popisují následující podkapitoly včetně hodnocení řešení dle daných kritérií. Hodnocení jednotlivých řešení je nejprve provedeno verbálně (pokud to umožňuje povaha kritéria) a v následujících kapitolách je provedeno srovnání. Obvyklá kritéria jako je cena nebo vzhled nebyla vzhledem k povaze řešení do hodnocení zařazena.

5.1.1 Základní požadavky

Prvním z hodnotících kritérií stanovuje míru, s jakou jednotlivá řešení naplňují základní požadavky na podobu a funkce aplikace uvedené v dané kapitole (3.5 Stanovení požadavků na podobu a funkce aplikace). Vzhledem k tomu, že se jedná o základní požadavky, jeví se jako vhodné tomuto kritériu dát velkou váhu (podrobněji dle vybrané metody dále) a zároveň se nepředpokládá, že by se mělo naplnění tohoto kritéria napříč řešeními významně lišit. Verbální hodnocení naplnění kritéria zobrazuje tabulka (Tabulka 3).

Tabulka 3 - Hodnocení nástrojů - základní požadavky

Google Maps API	ArcGIS API	Mapy API
Kontejnery jsou zobrazeny přesně a dle zásad, rozděleny dle typu a s možností zobrazit vhodné atributové informace při kliknutí na daný kontejner. Podoba stránek dodržuje dané zvyklosti.	Kontejnery jsou zobrazeny přesně a dle zásad, rozděleny dle typu a s možností zobrazit vhodné atributové informace při kliknutí na daný kontejner. Podoba stránek dodržuje dané zvyklosti.	Kontejnery jsou zobrazeny až o několik metrů od správného umístění. Kontejnery jsou rozděleny dle typu a s možností zobrazit vhodné atributové informace při kliknutí na daný kontejner. Podoba stránek dodržuje dané zvyklosti.

Zdroj: vlastní

5.1.2 Přepínání vrstev

Možnost přepínání (vypínání a zapínání) jednotlivých vrstev kontejnerů dle typu bylo jedním z rozšířených požadavků na funkce aplikace. Ve všech třech aplikacích je tato funkcionality řešena stejným způsobem a tedy panelem legendy, kde je vedle odpovídajících ikon umístěno zaškrtačkové políčko. Funkce pracuje dobře a spolehlivě ve všech řešeních v podstatě bez časové prodlevy, i když je vypínání vrstev v každém API řešeno odlišným způsobem. Implementace tohoto požadavku v žádném z nástrojů nečinila větší potíže.

5.1.3 Vyhledání nejbližšího kontejneru

Jedná se o možnost vyhledání kontejneru dle daného typu z množiny všech kontejnerů zobrazených ve vrstvě. Ve všech třech aplikacích je řešení funkční a pracuje správně. Stejně tak je ve všech tato funkce přístupná skrze položku v kontextovém menu. Jediná odlišnost je v řešení pomocí ArcGIS API, kdy přenos souřadnic z kliknutí zobrazující menu je problematický, proto je zadání výchozího bodu řešeno následným kliknutím do mapy, ke kterému je uživatel vyzván. Tento způsob se však nedá považovat za horší. Rychlost výpočtu a odezvy na zadání příkazu vyhledávání je součástí hodnotícího kritéria *rychlost*.

5.1.4 Vyhledání trasy k nejbližšímu kontejneru

V jisté podobě možnost vyhledávání trasy (routing) nabízí všechna tři API. Výsledky provedení této operace se však napříč aplikacemi poměrně výrazně liší. V řešení pomocí ArcGIS API tato funkce nebyla vůbec implementována z důvodů uvedených v dané kapitole. V obou zbývajících řešeních je vyhledávání trasy nastaveno na možnost hledání trasy pěšky. Paradoxně o něco lepší výsledky dosahuje vyhledávání trasy v řešení pomocí Google Maps API a to zejména na kratších vzdálenostech. Vzhledem k funkčnosti a opodstatnění implementace této funkcionality mu bude přiřazena poměrně menší váha. Zhodnocení naplnění tohoto kritéria zobrazuje tabulka (Tabulka 4).

Tabulka 4 - Hodnocení nástrojů – vyhledání trasy k nejbližšímu kontejneru

Google Maps API	ArcGIS API	Mapy API
Funguje s omezeními danými přesností vyhledávání. Podporuje možnost „pro pěší“. Dosahuje dobré přesnosti i na krátkých vzdálenostech.	Nebylo implementováno.	Funguje s omezeními danými přesností vyhledávání. Podporuje možnost „pro pěší“. Dosahuje jen slabé přesnosti na krátkých vzdálenostech.

Zdroj: vlastní

5.1.5 Vyhledání lokality

Funkce vyhledávání lokality dle zadané adresy, respektive v případě vytvořených aplikací dle ulice nebo městské části Pardubic, je označováno jako geokódování (geocoding). Ve všech třech API je implementování takovéto funkce možné a to přesně tak, jak je žádoucí. To znamená, že je možné vyhledávat ulice v rámci zájmového území připojením textového řetězce k zadanému vyhledávání (v každém API řešeno trochu jinak). Tato funkcionalita se jeví jako užitečnější nežli předchozí dvě a tento fakt bude zohledněn při přiřazování vah kritériím. Verbální hodnocení naplnění tohoto kritéria zobrazuje následující tabulka (Tabulka 5).

Tabulka 5 - Hodnocení nástrojů - vyhledání lokality

Google Maps API	ArcGIS API	Mapy API
Funguje přesně a správně s omezením 2500 požadavků denně.	Funguje přesně, správně a bez omezení.	Funguje přesně, správně a bez omezení.

Zdroj: vlastní

5.1.6 Čas potřebný pro tvorbu aplikace

Čas potřebný pro tvorbu aplikace představuje celkový čistý čas věnovaný danému řešení. To znamená studiu API a vytvoření aplikace pomocí daného nástroje. Do tohoto času se nepočítá popis nástroje v rámci práce. Aby tento ukazatel dával relevantní informace je důležité splnění výše zmíněného předpokladu počáteční nulové znalosti daného nástroje, což bylo splněno. Přestože ze strany autora byla maximální snaha o zachování objektivitu a zamezení zkreslení tohoto kritéria mírou úsilí věnovaného danému řešení, vypovídací hodnota toho ukazatele je stále ovlivněná subjektivními schopnostmi autora, což bude zohledněno při stanovování váhy tohoto kritéria. Výsledné časové hodnoty jsou uvedeny v tabulce (Tabulka 6) v hodinách a minutách.

Tabulka 6 - Hodnocení nástrojů - čas potřebný pro tvorbu aplikace

Google Maps API	ArcGIS API	Mapy API
39h 15m	42h 30m	31h 15m

Zdroj: vlastní

5.1.7 Rychlost

Toto kritérium představuje rychlost vykonávání jednotlivých funkcionalit poskytovaných aplikací. Měření rychlosti proběhlo na operacích: načtení mapy a vrstev,

vyhledání nejbližšího kontejneru, vyhledání trasy k nejbližšímu kontejneru. V rámci tohoto hodnotícího kritéria byla všem těmto operacím přisouzena stejná váha, proto byl v rámci každého řešení naměřený čas za všechny operace sečten (bylo však třeba jej normalizovat). Operace načtení mapy představuje časový úsek mezi zasláním požadavku na server a plného načtení zájmového území včetně všech třech vrstev kontejnerů. Bylo zvažováno prosté měření rychlosti odezvy serveru, jako je řešeno například v [43 s. 79], ale tento ukazatel byl vyhodnocen jako nedostatečně vypovídající charakteristika z pohledu uživatele. Také problematika měření by byla složitější, neboť stanovená operace načtení mapy a vrstev je dána odezvou serveru i způsobem vykonávání skriptu na straně klienta (C/S architektura). Ostatní dvě operace jsou měřeny jako časový úsek mezi zadáním příkazu a obdržetím výsledku. Pro maximální možné očištění od nežádoucích vlivů byly všechny tři aplikace umístěny na stejný server a měření provedeno na stejném připojení, ve stejném prohlížeči a s časovým rozestupem (podrobněji viz. soubor Vypočty.xlsx, Příloha D). Pro každou operaci bylo provedeno deset měření, přičemž jako výsledná hodnota byl vzat průměr očištěný od extrémních hodnot pomocí Dixonova testu [107 s. 97]. Zprůměrované hodnoty byly znormalizovány, aby se dosáhlo porovnatelnosti mezi časy jednotlivých operací. Normalizované průměry za jednotlivé operace byly poté sečteny, čímž byla získána výsledná hodnota zobrazená v tabulce níže (Tabulka 7) jako reprezentant celkové doby potřebné k vykonání těchto třech operací. U operace vyhledávání trasy k nejbližšímu kontejneru, která v řešení pomocí ArcGIS API nebyla implementována je pro účely porovnání u tohoto řešení vzat průměr naměřených hodnot ze zbylých dvou řešení. Všechny naměřené hodnoty a provedené testy jsou uvedeny v příloženém souboru MS Excel (Příloha D). Vzhledem k povaze tohoto hodnotícího kritéria mu bude přidělena velká váha.

Tabulka 7 - Hodnocení nástrojů - rychlost

Google Maps API	ArcGIS API	Mapy API
0,748	1,619	0,633

Zdroj: vlastní

5.1.8 Podkladové mapy

Jedním z významných hodnotících kritérií odlišujících jednotlivá řešení, jsou nabízené podkladové mapy. Jednotlivé nástroje se od sebe liší počtem poskytovaných podkladových mapových vrstev, jejich typem a podrobností. Vzhledem k faktu, že všechny tři nástroje umožňují importování externích dlaždicových vrstev [76] [96] [108] a tedy lze teoreticky v jakémkoli z nástrojů použít z poskytovaných map nejpodrobnější podkladovou mapu

distribuovanou Mapy.cz [109], váha tohoto kritéria musí být tomuto faktu přizpůsobena. Například ArcGIS API a priori počítá s případným importováním vlastních podrobnějších map distribuovaných přímo firmou ESRI nebo z jiného zdroje. Nicméně počet a podrobnost v základu poskytovaných map stále svou vypovídající hodnotu v rámci hodnocení má. Souvislost podrobnosti s měřítkem je u dynamicky volitelného měřítka jen částečná. Všechny použité základní mapové podklady vycházejí z map o měřítkách kolem 1:10 000, ale opticky se dají přiblížit na větší měřítka. Do sumarizace poskytovaných mapy byly zařazeny pouze relevantní mapy vzhledem k řešenému problému, tedy například u Mapy API nebyly započítány nabízené historické mapy. Stejně tak vzhledem k tomu, že všechny tři nástroje poskytují satelitní nebo letecké snímky, nebudou do sumarizace započítány. Počet a hodnocení poskytovaných map je uveden v tabulce (Tabulka 8).

Tabulka 8 - Hodnocení nástrojů - podkladové mapy

Google Maps API	ArcGIS API	Mapy API
Počet map: 1 Dostačující podrobnost v rámci zájmového území. Viditelné jsou ulice, domy a většina cest pro pěší.	Počet map: 1 Pouze slabá podrobnost v rámci zájmového území. Některé cesty pro pěší nejsou v mapě vůbec zachyceny. Podpora češtiny v místních názvech je špatná.	Počet map: 2 Velmi vysoká podrobnost v rámci zájmového území. Jsou k dispozici nejen všechny cesty pro pěší, ale i čísla popisná domů.

Zdroj: vlastní

5.1.9 Bohatost nástrojů API

Vzhledem k tomu, že řešený problém z daleka nevyčerpává všechny dostupné nástroje a funkce zkoumaných API, bylo zařazeno do hodnocení toto kritérium, které postihuje míru (množství) dalších dostupných funkcí, technologií, podporovaných formátů, rozšíření atp. Jedním z výstupů práce je dle zadání doporučení jednoho z nástrojů pro projekty obdobného charakteru. Přesto nebo právě proto se zdá vhodné zařadit do hodnocení i kritérium, které by postihlo možnosti daných API nad rámec funkcí použitých při tvorbě aplikací v rámci této práce. Vzhledem tomu, že je toto kritérium jen obtížně kvantifikovatelné, je uvedeno v tabulce níže (Tabulka 9) verbální hodnocení. To bude později vhodně převedeno na porovnatelné hodnoty. Hodnocení vychází z popisu jednotlivých nástrojů v předchozí kapitole a tabulky srovnání nástrojů (Tabulka 2).

Tabulka 9 - Hodnocení nástrojů - bohatost nástrojů API

Google Maps API	ArcGIS API	Mapy API
Google Maps API nabízí velkou řadu možností a v současné době se jedná v podstatě o profesionální řešení s možností integrace s dalšími Google API a nástroji jako jsou například Google Charts ap. Přesto je v neplacené verzi využití některých možností omezeno. Množství podporovaných formátů je velké.	ArcGIS API se jeví jako profesionální nástroj skýtající nejen velké množství funkcí, ale i různé prostorové analýzy s návazností na integrovatelné nástroje firmy ESRI jako jsou ArcGIS Server a ArcGIS Online. ArcGIS API podporuje velké množství formátů a umožňuje práci s bohatou škálou souřadnicových systémů.	Bohatost všech nástrojů poskytovaných Mapy API je z porovnávaných zdaleka nejnižší, což svědčí o tom, že není primárně určena pro tvorbu složitějších, profesionálních aplikací. Podporuje pouze formáty KML a GPX a služby WMS a WMTS.

Zdroj: vlastní

5.1.10 Zpracování a podpora API

Toto hodnotící kritérium je ze všech zvolených hodnotících kritérií nejhůře kvantifikovatelné. Představuje přehlednost, bohatost a srozumitelnost nápovědy, tutoriálu a dokumentace, také je zohledněno, jak jsou nástroj a komunita s ním pracující „živé“. Toto kritérium také hodnotí aktuálnost a četnost vydávaných verzí a přístup k uživatelům/vývojářům. Stejně jako u předchozího kritéria, tabulka (Tabulka 10) obsahuje verbální hodnocení.

Tabulka 10 - Hodnocení nástrojů - zpracování a podpora API

Google Maps API	ArcGIS API	Mapy API
Existuje velká podpora ze strany uživatelů/vývojářů. Podpůrné podklady (nápověda ap.) jsou zpracovány dobře, přehledně a s množstvím příkladů. Při přechodu na nové verze dochází k omezování nabízených možností. Existuje mnoho příkladů a řešení v angličtině, ale i v češtině.	Nástroj není příliš populární v širokém spektru uživatelů/vývojářů, lze však nalézt řešení různých problémů, v podstatě pouze v angličtině. Vývojáři poskytují množství příkladů a bohatou dokumentaci. Nové verze jsou vydávány pravidelně a rozšiřují možnosti nástroje.	Nástroj se zaměřuje v podstatě výhradně na české uživatele/vývojáře. Komunita uživatelů/vývojářů není příliš rozsáhlá. Podklady jsou zpracovány velmi přehledně, ale s malým množstvím příkladů a vysvětlení. Nové verze jsou vydávány pravidelně a rozšiřují možnosti nástroje.

Zdroj: vlastní

5.1.11 Licenční podmínky

Všechny porovnávané nástroje jsou k dispozici zdarma. Jejich licenční podmínky se však liší různým omezením na používání API. Všechny nástroje bez výjimky podmiňují použití zachováním loga firmy v obsahu mapy (v jednom z dolních rohů) neboli je zakázáno překrývat mapový podklad obsahující logo poskytovatele. Popis omezení používání je popsán níže v tabulce (Tabulka 11).

Tabulka 11 - Hodnocení nástrojů - licenční podmínky

Google Maps API	ArcGIS API	Mapy API
Zdarma pro nekomerční využití. Limit 2500 požadavků denně na vyhledávání lokality.	Je zdarma a bez omezení k dispozici pouze pro nekomerční a výukové účely.	Je k dispozici bez limitů na služby a zobrazení i pro komerční využití.

Zdroj: vlastní

5.2 Ohodnocení řešení dle kritérií

Protože u některých hodnotících kritérií bylo v předchozí kapitole provedeno pouze verbální hodnocení jednotlivých řešení, pro usnadnění dalšího postupu bylo hodnocení zjednodušeno a částečně kvantifikováno. Dle [110] byla vytvořena rozhodovací tabulka (Tabulka 12), kde je pro každou z variant uvedeno hodnocení za každé kritérium a dále typ každého kritéria (MIN – minimalizační, MAX – maximalizační). V tabulce je také názvům jednotlivých kritérií přiřazeno označení K1-K9. Kritéria *Přepínání vrstev* a *Vyhledání nejbližšího kontejneru* byla z procesu vyřazena, protože ve všech variantách (řešeních) bylo dosaženo stejného (nebo kvalitativně shodného) výsledku.

Tabulka 12 - Rozhodovací tabulka

Kritérium	Typ	Varianta (řešení)		
		Google Maps API	ArcGIS API	Mapy API
K1 – Základní požadavky	MAX	Splněny	Splněny	Částečně splněny
K2 – Vyhledání trasy k nejbližšímu kontejneru	MAX	Plně funkční	Neimplementováno	Plně funkční
K3 – Vyhledání lokality	MAX	Funkční s omezením	Funkční bez omezení	Funkční bez omezení
K4 – Čas potřebný pro tvorbu aplikace	MIN	39,25 hodin	42,5 hodin	31,25 hodin
K5 – Rychlost	MIN	0,748	1,619	0,633
K6 – Podkladové mapy	MAX	Dobré	Dostačující	Výborné
K7 – Bohatost nástrojů API	MAX	Velmi dobré	Výborné	Dostačující
K8 – Zpracování a podpora API	MAX	Velmi dobré	Dobré	Dobré
K9 – Licenční podmínky	MAX	Dobré	Velmi dobré	Výborné

Zdroj: vlastní

5.3 Stanovení vah kritérií

Problém stanovení vah kritérií včetně popisu jednotlivých metod podrobně popisuje kapitola 2.1 – *Metody stanovení vah kritérií* na webovém portále *Rozhodovací procesy* [110]. Dle povahy problému byla vybrána pro stanovení vah kritérií jedna z *metod bez znalostí důsledků variant*. Pro svoji přesnost, kvalitu srovnání a možnost dobrého porovnání nekvantifikovaných hodnot kritérií byla vybrána metoda kvantitativního párového srovnávání neboli tzv. Saatyho metoda [110]. Tato metoda díky svému principu nevyžaduje předběžnou kvantifikaci hodnocení, jak byla provedena v rozhodovací tabulce výše (Tabulka 12), neboť párové porovnávání umožňuje porovnávat jednotlivá hodnocení mezi sebou i pro nečíselné nebo pouze verbálně vyjádřené hodnoty. Hodnocení bylo provedeno autorem práce. U některých položek bylo při vypracovávání stanovení vah kritérií přihlédnuto i k verbálnímu hodnocení uvedenému v hodnotících tabulkách příslušných kritérií. Vypracování hodnocení pomocí Saatyho metody je součástí přílohy (Vypocty.xlsx Příloha D). Vzhledem k počtu kritérií bylo rozhodnuto použít jemnější rozlišení pro hodnoty ohodnocení (tzv. Saatyho stupnice relativních důležitostí [110]), kdy hodnoty ohodnocení $s_{ij} \in \{1,2,3,4,5,6,7,8,9\}$, přičemž nižší hodnoty představují menší relativní důležitost kritéria a vyšší hodnoty představují větší relativní důležitost kritéria. Stanovení relativní důležitosti kritérií pomocí Saatyho matice včetně normovaného vektoru vah \mathbf{v} ukazuje tabulka níže (Tabulka 13). Hodnoty b_i jsou hodnotami vektoru nenormovaných vah vypočtené jako geometrický průměr daného řádku matice.

Tabulka 13 - Saatyho matice a váhy kritérií

	K1	K2	K3	K4	K5	K6	K7	K8	K9	b_i	v_i
K1	1,00	9,00	9,00	7,00	3,00	6,00	3,00	5,00	4,00	4,3954	0,3150
K2	0,11	1,00	1,00	0,33	0,13	0,50	0,14	0,17	0,17	0,2756	0,0198
K3	0,11	1,00	1,00	0,33	0,13	0,50	0,14	0,17	0,17	0,2756	0,0198
K4	0,14	3,00	3,00	1,00	0,20	2,00	0,20	0,33	0,25	0,5893	0,0422
K5	0,33	8,00	8,00	5,00	1,00	7,00	2,00	3,00	3,00	2,8755	0,2061
K6	0,17	2,00	2,00	0,50	0,14	1,00	0,14	0,25	0,20	0,4117	0,0295
K7	0,33	7,00	7,00	5,00	0,50	7,00	1,00	4,00	3,00	2,4706	0,1771
K8	0,20	6,00	6,00	3,00	0,33	4,00	0,25	1,00	0,50	1,1530	0,0826
K9	0,25	6,00	6,00	4,00	0,33	5,00	0,33	2,00	1,00	1,5066	0,1080

Zdroj: vlastní

Konzistenční poměr CR vychází roven 0,077. To znamená, že matice je logicky správně sestavená [110] (výpočet viz Vypocty.xlsx Příloha D).

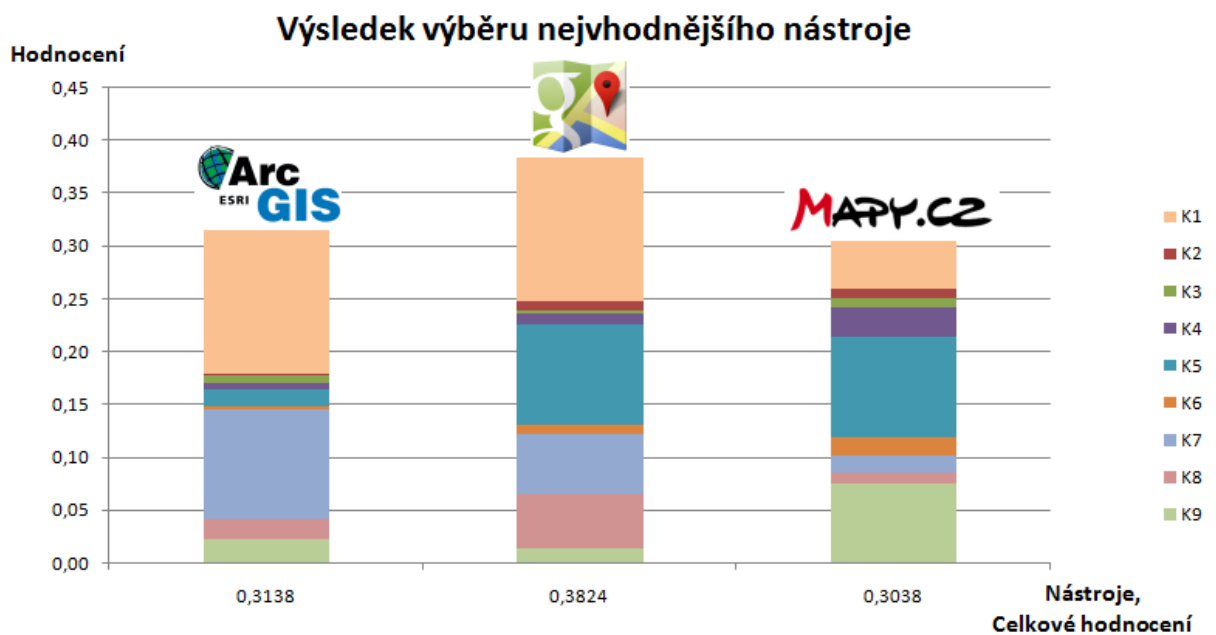
5.4 Výběr nejlepší varianty

Podrobněji je proces výběru nejlepší varianty uveden v příloze (Vypočty.xlsx Příloha D). K celkovému ohodnocení variant byla taktéž použita Saatyho metoda v tom smyslu, že pro každé kritérium byla vytvořena Saatyho matice tak, že pro každé kritérium byl získán vektor normovaných hodnot představující ohodnocení variant dle daného kritéria. Výsledek výběru nejlepší varianty (řešení) zobrazuje tabulka níže (Tabulka 14) a graficky jej vykresluje obrázek (Obrázek 12) pojatý jako „stupně vítězů“.

Tabulka 14 - Výsledné hodnocení

Kritérium	Váha	Ohodnocení variant		
		Google Maps API	ArcGIS API	Mapy API
K1	0,3150	0,4286	0,4286	0,1429
K2	0,0198	0,4737	0,0526	0,4737
K3	0,0198	0,1429	0,4286	0,4286
K4	0,0422	0,2297	0,1220	0,6483
K5	0,2061	0,4615	0,0769	0,4615
K6	0,0295	0,3000	0,1000	0,6000
K7	0,1771	0,3234	0,5876	0,0890
K8	0,0826	0,6250	0,2385	0,1365
K9	0,1080	0,1168	0,1998	0,6833
Celkové hodnocení		0,3824	0,3138	0,3038
Pořadí		1.	2.	3.

Zdroj: vlastní



Obrázek 12 – Výsledek výběru nejvhodnějšího nástroje

Zdroj: vlastní

Jak je vidět na obrázku (Obrázek 12) a podrobněji uvedeno v tabulce (Tabulka 14), nejvhodnějším nástrojem pro tvorbu podobného druhu aplikace (ve smyslu zadaných cílů práce) byl za použití metody multikriteriálního rozhodování (konkrétně Saatyho metody) vybrán nástroj Google Maps API. Výsledné rozdíly v hodnocení jednotlivých nástrojů nejsou příliš markantní. Z obrázku je také patrné jak které kritérium přispělo do výsledného hodnocení a také fakt, že velkou roli na výsledek měly stanovené váhy kritérií.

5.5 Závěrečné zhodnocení a možnosti dalšího využití

Vhledem k tomu, že důležitost kritérií byla mimo jiné stanovena s ohledem na povahu vytvářené aplikace, lze říci, že při vytváření obdobné aplikace je možné řídit se výsledným doporučením. Popis jednotlivých nástrojů pak může sloužit jako opora při implementaci funkcí aplikace v jiném budoucím řešení obdobného problému.

Výsledné řešení pomocí nástroje Google Maps API, které je možno brát jako „vítězné“, bylo drobně upraveno tak, aby byl vzhled aplikace optimalizován pro možnost veřejného využívání. Byla doplněna nápověda a odkaz v podobě banneru na stránky Univerzity Pardubice a Google Developers (Příloha A zobrazuje výslednou podobu včetně úprav). Případná volba jiného řešení pro využívání, nežli řešení pomocí Google Maps API, by měla vycházet z další analýzy s větším zaměřením na pohled ze strany uživatele.

Všechna tři řešení jsou bez obtíží umístitelná obecně na libovolný webový server a připravena k využívání nebo k dalšímu studiu problematiky. Povaha řešení také umožňuje relativně snadnou aktualizaci sebraných dat, obohacení o data nová a podobně. Distribuovaný charakter použití datového formátu KML těmto možnostem napomáhá. Pro další efektivní využívání aplikace je však nutno zvážit vhodný způsob sběru a aktualizace dat. Jako nejvhodnější se jeví získávání informací o aktuálních umístěných kontejnerech (nových, přemístěných apod.) ze strany jejich správce, tedy města Pardubice.

Za úvahu také stojí možnosti dalšího využívání a rozšíření vytvořené „vítězné“ aplikace. Při možnostech současných mobilních aplikací je na zvážení, zda by byl ochoten uživatel využívat služby poskytované případnou mobilní verzí vytvořené aplikace. Stejně tak některé aplikací poskytované služby by mohly být rozšířeny v rámci omezení daných nástrojem. Například vyhledávání trasy by mohlo nabízet alternativní trasy, vzdálenost apod. Data využitá v rámci práce by šla také dále upravit a distribuovat je pomocí jiných metod jako je kupříkladu využití WMS.

ZÁVĚR

Recyklování odpadů je velmi významný a aktuální problém a cílem této práce bylo řešení problematiky prezentace umístění kontejnerů v Pardubicích. První část práce byla věnována prezentaci prostorových dat především se zaměřením na interaktivní prezentaci prostorových dat. Byla definována specifika prostorových dat a jejich vizualizace v prostředí internetu. Prostorová data v podobě kontejnerů na separovaný odpad použitá v této práci byla získána v rámci diplomové práce Lenky Skopalíkové: Dostupnost kontejnerů pro separovaný odpad v Pardubicích. Dále byly zhodnoceny možnosti využití vybraných nástrojů pro tvorbu webové aplikace. Z těchto nástrojů byly vybrány tři nástroje pro tvorbu dané aplikace tak, že aplikace byla vytvořena nezávisle v každém z nich. Nástroje byly vybrány tak, aby bylo v poslední části práce možné nástroje zhodnotit a vybrat z nich ten nejvhodnější pro tvorbu aplikace obdobného charakteru. Předem byly stanoveny požadavky na podobu a funkce aplikace společné pro tvorbu pomocí všech třech nástrojů.

Prostřední část práce je věnována popisu tvorby aplikace pomocí vybraných nástrojů Google Maps API, ArcGIS API a Mapy API včetně popisu těchto nástrojů. Pomocí všech třech zmíněných nástrojů byla webová aplikace pro prezentaci kontejnerů na recyklovaný odpad vytvořena včetně úspěšné implementace stanovených požadavků. Pouze požadavek na vyhledání trasy k nejbližšímu kontejneru nebyl implementován v řešení pomocí nástroje ArcGIS API a tento fakt byl zohledněn v závěrečném porovnání použitých nástrojů.

Porovnání použitých nástrojů proběhlo za použití metod multikriteriálního rozhodování. Byla stanovena hodnotící kritéria a provedeno verbální hodnocení použitých nástrojů. Dále byly pomocí metody kvantitativního párového srovnávání stanoveny váhy jednotlivých kritérií a provedeno ohodnocení. Jako nejvhodnější nástroj byl vybrán Google Maps API a lze ho tedy z vybraných nástrojů nejvíce doporučit pro tvorbu aplikace obdobného charakteru. Přínosem práce je mimo jiné návod, jak takovouto aplikaci realizovat ve zvolených třech nástrojích, a výběr sady kritérií porovnání zvolených nástrojů. Na základě výše zmíněného lze konstatovat, že cíl práce byl splněn.

POUŽITÁ LITERATURA

1. RAPANT, Petr. *Úvod do geografických informačních systémů* [online]. Ostrava: VŠB - TU Ostrava, 2002 [cit. 2014-04-24]. Dostupné z: <http://gis.vsb.cz/dokumenty/ugis>
2. LONGLEY, Paul A. *Geographic information systems & science*. 3. vyd. Hoboken John Wiley & Sons, 2011, 539 s. ISBN 978-0-470-72144-5.
3. ROUSE, Margaret. Spatial data. *TextTarget* [online]. [cit. 2014-04-24]. Dostupné z: <http://searchsqlserver.techtargget.com/definition/spatial-data>
4. LALIBERTÉ, Larry. What is Spatial/GIS Data?. In: *University of Alberta* [online]. 2013 [cit. 2014-04-25]. Dostupné z: <http://guides.library.ualberta.ca/spatial>
5. HOWARTH, J. What is Geospatial Data?. In: *York University* [online]. 2014 [cit. 2014-04-25]. Dostupné z: <http://researchguides.library.yorku.ca/content.php?pid=245987&sid=2176375>
6. KOMÁRKOVÁ, Jitka a Hana KOPÁČKOVÁ. *Geografické informační systémy: pro kombinovanou formu studia*. 1. vyd. Pardubice: Univerzita Pardubice, 2005, 55 s. ISBN 80-7194-019-5.
7. The GIS Spatial Data Model. UNIVERSITY OF WASHINGTON. *University of Washington* [online]. [cit. 2014-04-24]. Dostupné z: http://courses.washington.edu/gis250/lessons/introduction_gis/spatial_data_model.html
8. HRUBÝ, Martin. Geografické Informační Systémy (GIS). [dokument ve formátu PDF]. 2006 [cit. 2014-04-25]. Dostupné z: <http://perchta.fit.vutbr.cz/vyuka-gis/uploads/1/GIS-final2.pdf>
9. FU, Pinde a Jiulin SUN. *Web GIS: Principles and Applications*. 1. vyd. Redlands ESRI Press, 2011, 296 s. ISBN 978-1-58948-245-6.
10. VOŽENÍLEK, Vít a Jaromír KAŇOK. *Metody tematické kartografie: vizualizace prostorových jevů*. Olomouc Univerzita Palackého, 2011, 216 s. ISBN 978-80-244-2790-4.
11. KRAAK, M. J. *Beyond Geovisualization*. IEEE Computer Graphics and Applications 2006, roč. 26, č. 4. DOI: 10.1109/MCG.2006.74.

12. ARMBRUST, Michael et al. A View of Cloud Computing. *Communications of the ACM* [online]. 2010, roč. 53, č. 4 [cit. 2014-04-25]. DOI: 10.1145/1721654.1721672. Dostupné z: <http://cacm.acm.org/magazines/2010/4/81493-a-view-of-cloud-computing/fulltext>
13. ROUSE, Margaret. Software as a Service (SaaS). In: *Searchcloudcomputing* [online]. 2010 [cit. 2014-04-25]. Dostupné z: <http://searchcloudcomputing.techtarget.com/definition/Software-as-a-Service>
14. FEW, Stephen. Introduction to Geographical Data Visualization. *Visual Business Intelligence Newsletter* [online]. 2009 [cit. 2014-04-25]. Dostupné z: http://www.perceptualedge.com/articles/visual_business_intelligence/geographical_data_visualization.pdf
15. NEUMANN, Andreas. SWISS FEDERAL INSTITUTE OF TECHNOLOGY ZURICH. *Geodata Visualisation*. 2010.
16. HAVLÍČEK, Jakub. Digitální technologie v geoinformatice, kartografii a DPZ. [dokument ve formátu PDF]. Praha, 2012.
17. DUŠEK, Radek a Jakub MIŘIJOVSKÝ. Vizualizace prostorových dat: Chaos v dimenzích. In: *Geografie: Sborník České geografické společnosti*. Praha, 2009, roč. 114, č. 3, s. 169-178. ISSN 1212-0014.
18. Trojrozměrné budovy. GOOGLE.COM. *Google Earth* [online]. 2014 [cit. 2014-04-25]. Dostupné z: <http://www.google.cz/intl/cs/earth/explore/showcase/3dbuildings.html>
19. THROWER, Norman J. W. *Maps and Civilization: Cartography in Culture and Society*. University of Chicago Press, 2008, 362 s. 3. ISBN 978-0226799742.
20. SLOCUM, Terry A. et al. *Thematic cartography and geovisualization*. Upper Saddle River Pearson Education, 2009, 561 s. 3. ISBN 978-0-13-801006-5.
21. BRINEY, Amanda. Thematic Maps. *Geography.about.com* [online]. 2009 [cit. 2014-04-25]. Dostupné z: <http://geography.about.com/od/understandmaps/a/thematicmaps.htm>

22. Jak a který souřadnicový systém pro území ČR lze přiřadit datům v ArcGIS Desktop?. ARCDATA PRAHA, s.r.o. [online]. 2014 [cit. 2014-04-25]. Dostupné z: <http://www.arcdata.cz/podpora/tipy-a-triky/Detail/?contentId=95823>
23. Internet Users. *Internet live stats* [online]. 2014 [cit. 2014-04-25]. Dostupné z: <http://www.internetlivestats.com/internet-users>
24. How popular is mapy.cz?. AMAZON.COM. *Alexa* [online]. 2014 [cit. 2014-04-25]. Dostupné z: <http://www.alexa.com/siteinfo/mapy.cz>
25. PENG, Zhong-Ren. *Internet GIS: distributed geographic information services for the internet and wireless networks*. Hoboken John Wiley & Sons, 2003. ISBN 0-471-35923-8.
26. TAIT, Michael G. Implementing geoportals: applications of distributed GIS. In: *Computers, Environment and Urban Systems*. 2005, s. 33-47. 29: 1. ISSN 0198-9715.
27. LEMBO, Arthur J. et al Creating affordable Internet map server applications for regional scale applications. In: *Journal of Environmental Management*. ELSEVIER, 2007, 1120–1131. 85 (2007). ISSN 0301-4797.
28. About web GIS. ESRI. *ArcGIS Resources* [online]. 2014 [cit. 2014-04-25]. Dostupné z: <http://resources.arcgis.com/en/help/main/10.1/index.html#//0154000002ws000000>
29. GOLDSTONE, Robert a James GILL. Web Site Operators & Liability for UGC - Facing up to Reality?. In: *SCL: The IT law community* [online]. 2008 [cit. 2014-04-25]. Dostupné z: <http://www.scl.org/site.aspx?i=ed9981>
30. GOODCHILD, Michael F. Citizens As Sensors: The World Of Volunteered Geography. [online]. 2014 [cit. 2014-04-25]. Dostupné z: http://www.ncgia.ucsb.edu/projects/vgi/docs/position/Goodchild_VGI2007.pdf
31. MCKENNA, Frank. Thin-Client Versus Fat-Client Computing. [online]. 2002 [cit. 2014-04-25]. Dostupné z: <http://www.knowledgeonecorp.com/news/pdfs/Thin%20client%20vs%20Fat%20client%20Computing.pdf>
32. KOČÍ, Michal. Co je XML?. In: *Interval.cz* [online]. 2000 [cit. 2014-04-25]. Dostupné z: <http://interval.cz/clanky/co-je-xml/>

33. XML Tutorial. REFSNES DATA. W3schools [online]. 2014 [cit. 2014-04-26]. Dostupné z: <http://www.w3schools.com/xml/default.ASP>
34. Extensible Markup Language (XML). W3C. W3C Ubiquitous Web Domain [online]. 2013 [cit. 2014-04-26]. Dostupné z: <http://www.w3.org/XML/>
35. Úvod do JSON. Json.org [online]. 2014 [cit. 2014-04-26]. Dostupné z: <http://www.json.org/json-cz.html>
36. LINDO, Sean. XML vs. JSON – A Primer. PROGRAMMABLEWEB.COM. ProgrammableWeb [online]. 2013 [cit. 2014-04-26]. Dostupné z: <http://blog.programmableweb.com/2013/11/07/xml-vs-json-a-primer/>
37. Keyhole Markup Language: KML Tutorial. GOOGLE.COM. Google Developers [online]. 2014 [cit. 2014-04-26]. Dostupné z: https://developers.google.com/kml/documentation/kml_tut
38. KML. OGC. OGC Making location count [online]. 2014 [cit. 2014-04-26]. Dostupné z: <http://www.opengeospatial.org/standards/kml>
39. Keyhole Markup Language: KMZ Files. GOOGLE.COM. Google Developers [online]. 2014 [cit. 2014-04-26]. Dostupné z: <https://developers.google.com/kml/documentation/kmzarchives?csw=1>
40. Web Map Service. OGC. OGC Making location count [online]. 2014 [cit. 2014-04-26]. Dostupné z: <http://www.opengeospatial.org/standards/wms>
41. FICHTER, Darlene. What Is a Mashup?. UNIVERSITY OF SASKATCHEWAN LIBRARY. [online]. 2009 [cit. 2014-04-26]. Dostupné z: <http://books.infotoday.com/books/Engard/Engard-Sample-Chapter.pdf>
42. Mashup Dashboard. PROGRAMMABLEWEB.COM. ProgrammableWeb [online]. 2014 [cit. 2014-04-26]. Dostupné z: <http://www.programmableweb.com/mashups>
43. KOMÁRKOVÁ, Jitka. Kvalita webových geografických informačních systémů. Vyd. 1. Pardubice: Univerzita Pardubice, 2008, 127 s. ISBN 978-80-7395-056-9.

44. ESRI Shapefiles (SHP). REGENTS OF THE UNIVERSITY OF MINNESOTA. Map Server: open source web mapping [online]. 2014 [cit. 2014-04-26]. Dostupné z: <http://mapserver.org/input/vector/shapefiles.html>
45. ČEPICKÝ, Jáchym. Mapový server snadno a rychle (1). ROOT.CZ. Root.cz [online]. 2005 [cit. 2014-04-26]. Dostupné z: <http://www.root.cz/clanky/mapovy-server-snadno-a-rychle-1/>
46. DUBROVSKÝ, Jakub. Mapy API. Mapy API [online]. 2009 [cit. 2014-04-26]. Dostupné z: <http://www.fi.muni.cz/~xobsivac/PV219/prezentace09/index.html>
47. Deliver GIS as a Service. ESRI. ArcGIS for Server [online]. 2014 [cit. 2014-04-26]. Dostupné z: <http://www.esri.com/software/arcgis/arcgisserver>
48. Welcome. ATLISSIAN CONFLUENCE. GeoServer [online]. 2014 [cit. 2014-04-26]. Dostupné z: <http://geoserver.org/display/GEOS/Welcome>
49. Welcome to MapServer. REGENTS OF THE UNIVERSITY OF MINNESOTA. Map Server: open source web mapping [online]. 2014 [cit. 2014-04-26]. Dostupné z: <http://mapserver.org/>
50. T-MapServer. *T-MAPY* [online]. 2014 [cit. 2014-04-25]. Dostupné z: http://www.tmapy.cz/public/tmapy/cz/_software/twist/_mapy.html
51. Google Maps Javascript API v3. GOOGLE.COM. Google Developers [online]. 2014 [cit. 2014-04-25]. Dostupné z: <https://developers.google.com/maps/documentation/javascript/?hl=cs>
52. Mapy API verze 4.8 – Hanzelka a Zikmund. SEZNAM.CZ, a.s. Mapy.cz [online]. 2014 [cit. 2014-04-25]. Dostupné z: <http://api.mapy.cz/>
53. ArcGIS API for JavaScript. ESRI. ArcGIS for Developers [online]. 2014 [cit. 2014-04-25]. Dostupné z: <https://developers.arcgis.com/javascript/>
54. OpenLayers: Free Maps for the Web. *OpenLayers* [online]. 2014 [cit. 2014-04-25]. Dostupné z: <http://openlayers.org/>

55. Competitive Intelligence. ALEXA INTERNET, Inc. Alexa [online]. 2014 [cit. 2014-04-25]. Dostupné z: <http://www.alexa.com/siteinfo/maps.google.com>
56. KRAVAL, Ilja. Rozdíl mezi vztahem extend a include v use case diagramech: 2 . část. Objects.cz [dokument ve formátu PDF]. 2009, červen 2009, s. 6 [cit. 2014-04-25]. Dostupné z: <http://www.objects.cz/clanky/clanek65/clanek65.pdf>
57. KUČEROVÁ, Helena. Use Case model. VYŠŠÍ ODBORNÁ ŠKOLA INFORMAČNÍCH SLUŽEB. Vývoj informačních systémů 2011/2012 [online]. 2011 [cit. 2014-04-25]. Dostupné z: <http://web.sks.cz/users/ku/pri/usecase.htm>
58. SEHLHORST, Scott. What Are Use Case Scenarios?. Tynerblain [online]. 2007 [cit. 2014-04-25]. Dostupné z: <http://tynerblain.com/blog/2007/04/10/what-are-use-case-scenarios/>
59. LEMAY, Renai. Google mapper: Take browsers to the limit. CNET [online]. 2005 [cit. 2014-04-25]. Dostupné z: http://news.cnet.com/Google-mapper-Take-browsers-to-the-limit/2100-1038_3-5808658.html
60. KISS, Jemima. Secrets of a nimble giant. The Guardian [online]. 2009 [cit. 2014-04-25]. Dostupné z: <http://www.theguardian.com/technology/2009/jun/17/google-sergey-brin>
61. TAYLOR, Bret. The world is your JavaScript-enabled oyster. Google | Official blog [online]. 2005 [cit. 2014-04-25]. Dostupné z: http://googleblog.blogspot.cz/2005/06/world-is-your-javascript-enabled_29.html
62. Google Maps API Subject to the Deprecation Policy. GOOGLE.COM. Google Developers [online]. 2014 [cit. 2014-04-25]. Dostupné z: <https://developers.google.com/maps/maps-api-list>
63. Getting Started. GOOGLE.COM. Google Developers [online]. 2014 [cit. 2014-04-25]. Dostupné z: <https://developers.google.com/maps/documentation/javascript/tutorial>
64. Maps API for Business. GOOGLE.COM. Google Developers [online]. 2014 [cit. 2014-04-25]. Dostupné z: <https://developers.google.com/maps/documentation/business/>
65. JANOVSKEÝ, Dušan. Začlenění skriptu do stránky. Jak psát web [online]. 2014 [cit. 2014-04-25]. Dostupné z: <http://www.jakpsatweb.cz/javascript/zacleneni.html>

66. Google Maps Javascript API V3 Reference. GOOGLE.COM. Google Developers [online]. 2014 [cit. 2014-04-25]. Dostupné z: <https://developers.google.com/maps/documentation/javascript/reference>
67. What is jQuery?. THE JQUERY FOUNDATION. JQuery: write less, do more. [online]. 2014 [cit. 2014-04-25]. Dostupné z: <http://jquery.com/>
68. What is ArcGIS?. ESRI. ArcGIS Resources [online]. 2014 [cit. 2014-04-25]. Dostupné z: <http://resources.arcgis.com/en/help/getting-started/articles/026n00000014000000.htm>
69. The Power of Location. ESRI. ArcGIS for Developers [online]. 2014 [cit. 2014-04-25]. Dostupné z: <https://developers.arcgis.com/en/>
70. ArcGIS API for JavaScript Overview. ESRI. ArcGIS for Developers [online]. 2014 [cit. 2014-04-25]. Dostupné z: <https://developers.arcgis.com/javascript/jshelp/>
71. ArcGIS API for JavaScript Version 3.9 Released. ESRI. ArcGIS Resources [online]. 2014 [cit. 2014-04-25]. Dostupné z: <http://blogs.esri.com/esri/arcgis/2014/04/07/arcgis-api-for-javascript-version-3-9-released/>
72. ArcGIS API for JavaScript. ESRI. ArcGIS for Developers [online]. 2010 [cit. 2014-04-25]. Dostupné z: <https://developers.arcgis.com/javascript/jshelp/terms.html>
73. Get the ArcGIS API for JavaScript. ESRI. ArcGIS for Developers [online]. 2014 [cit. 2014-04-25]. Dostupné z: https://developers.arcgis.com/javascript/jshelp/intro_accessapi.html
74. Working with Dojo. ESRI. ArcGIS for Developers [online]. 2014 [cit. 2014-04-25]. Dostupné z: https://developers.arcgis.com/javascript/jshelp/inside_dojo.html
75. Build your first application. ESRI. ArcGIS for Developers [online]. 2014 [cit. 2014-04-25]. Dostupné z: <https://developers.arcgis.com/javascript/jstutorials/>
76. Samples. ESRI. ArcGIS for Developers [online]. 2014 [cit. 2014-04-25]. Dostupné z: <https://developers.arcgis.com/javascript/jssamples>
77. API Reference Overview. ESRI. ArcGIS for Developers [online]. 2014 [cit. 2014-04-25]. Dostupné z: <https://developers.arcgis.com/javascript/jsapi/>

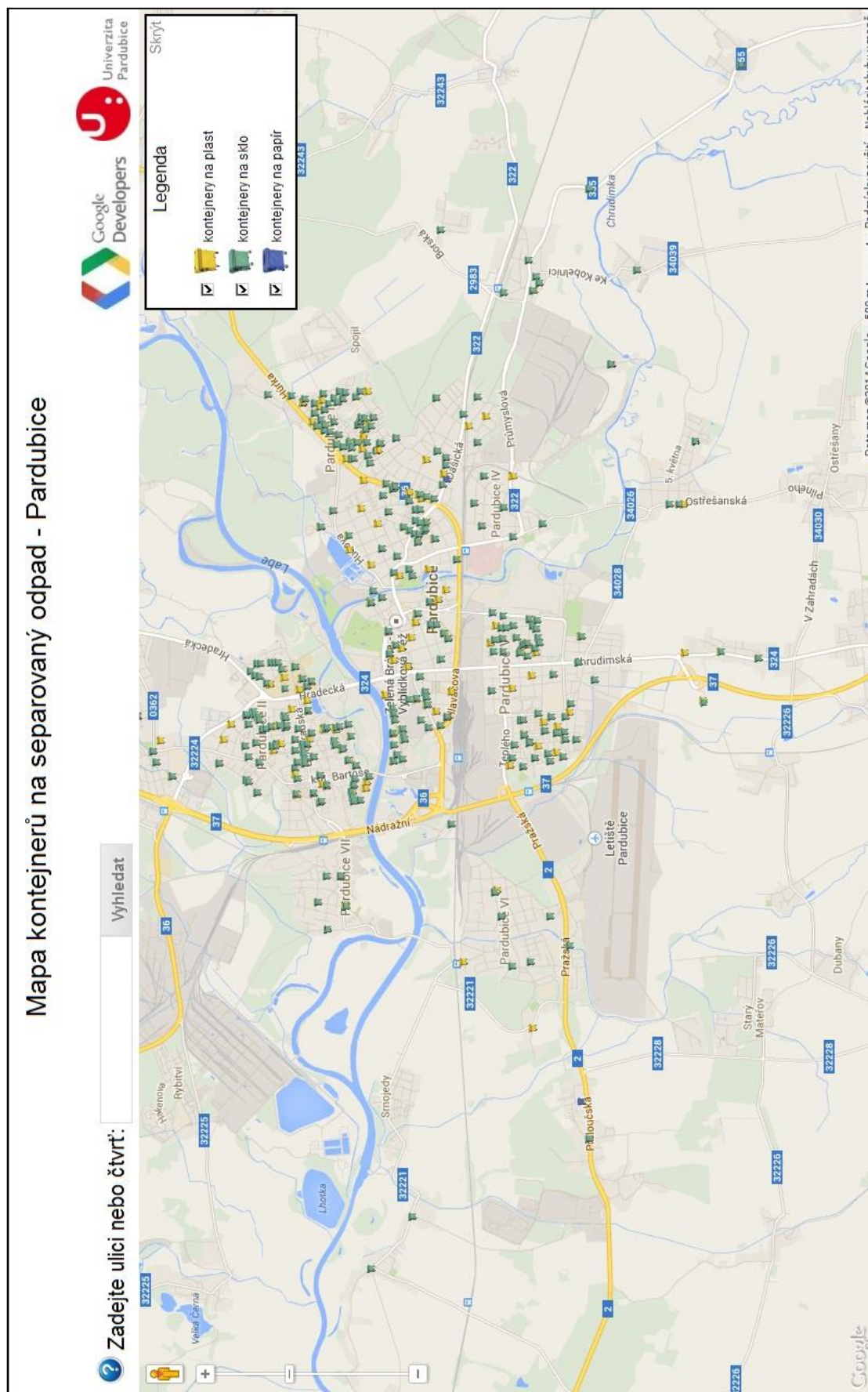
78. Directions widget. ESRI. ArcGIS for Developers [online]. 2014 [cit. 2014-04-25]. Dostupné z: https://developers.arcgis.com/javascript/jstutorials/intro_directions.html
79. Geocoder widget. ESRI. ArcGIS for Developers [online]. 2014 [cit. 2014-04-25]. Dostupné z: https://developers.arcgis.com/javascript/jstutorials/tutorial_geocoder.html
80. Přehled souřadnicových systémů používaných na území ČR a SR v ArcGIS 10. ARCDATA PRAHA, s.r.o. ARCDATA PRAHA: Geografické informační systémy [online]. 2014 [cit. 2014-04-26]. Dostupné z: <http://www.arcdata.cz/podpora/tipy-a-triky/Detail/?contentId=110111>
81. Geographic Coordinate Systems. ESRI. ArcGIS REST API [online]. 2014 [cit. 2014-04-26]. Dostupné z: <http://help.arcgis.com/en/arcgisserver/10.0/apis/rest/gcs.html>
82. Documentation: dijit/Menu. DOJO FOUNDATION. Dojo [online]. 2012 [cit. 2014-04-26]. Dostupné z: <http://dojotoolkit.org/reference-guide/1.9/dijit/Menu.html#id2>
83. ArcGIS REST API: Overview of network analysis services. ESRI. ArcGIS Resources [online]. 2014 [cit. 2014-04-26]. Dostupné z: http://resources.arcgis.com/en/help/arcgis-rest-api/#/Overview_of_Network_Analysis_Services/02r30000001s000000
84. HTML input value Attribute. REFSNES DATA. W3Schools [online]. 2014 [cit. 2014-04-26]. Dostupné z: http://www.w3schools.com/tags/att_input_value.asp
85. CSS Positioning. REFSNES DATA. W3Schools [online]. 2014 [cit. 2014-04-26]. Dostupné z: http://www.w3schools.com/css/css_positioning.asp
86. Mapy.cz. SEZNAM.CZ, a.s. Seznam.cz: O nás [online]. 2014 [cit. 2014-04-26]. Dostupné z: <http://onas.seznam.cz/cz/mapy-cz.html>
87. SNÍŽEK, Martin. Mapy mají těžko použitelné API. Snizekweb.cz [online]. 2014 [cit. 2014-04-26]. Dostupné z: <http://www.snizekweb.cz/weblog/mapy-cz-api/>
88. MALÝ, Martin. Seznam nabízí API svých map i pro komerční využití. DEVEL.CZ LAB S.R.O. Zdroják, o tvorbě webových stránek a aplikací [online]. 2011 [cit. 2014-04-26]. Dostupné z: <http://www.zdrojak.cz/zpravicky/seznam-nabizi-api-svych-map-i-pro-komercni-vyuziti/>

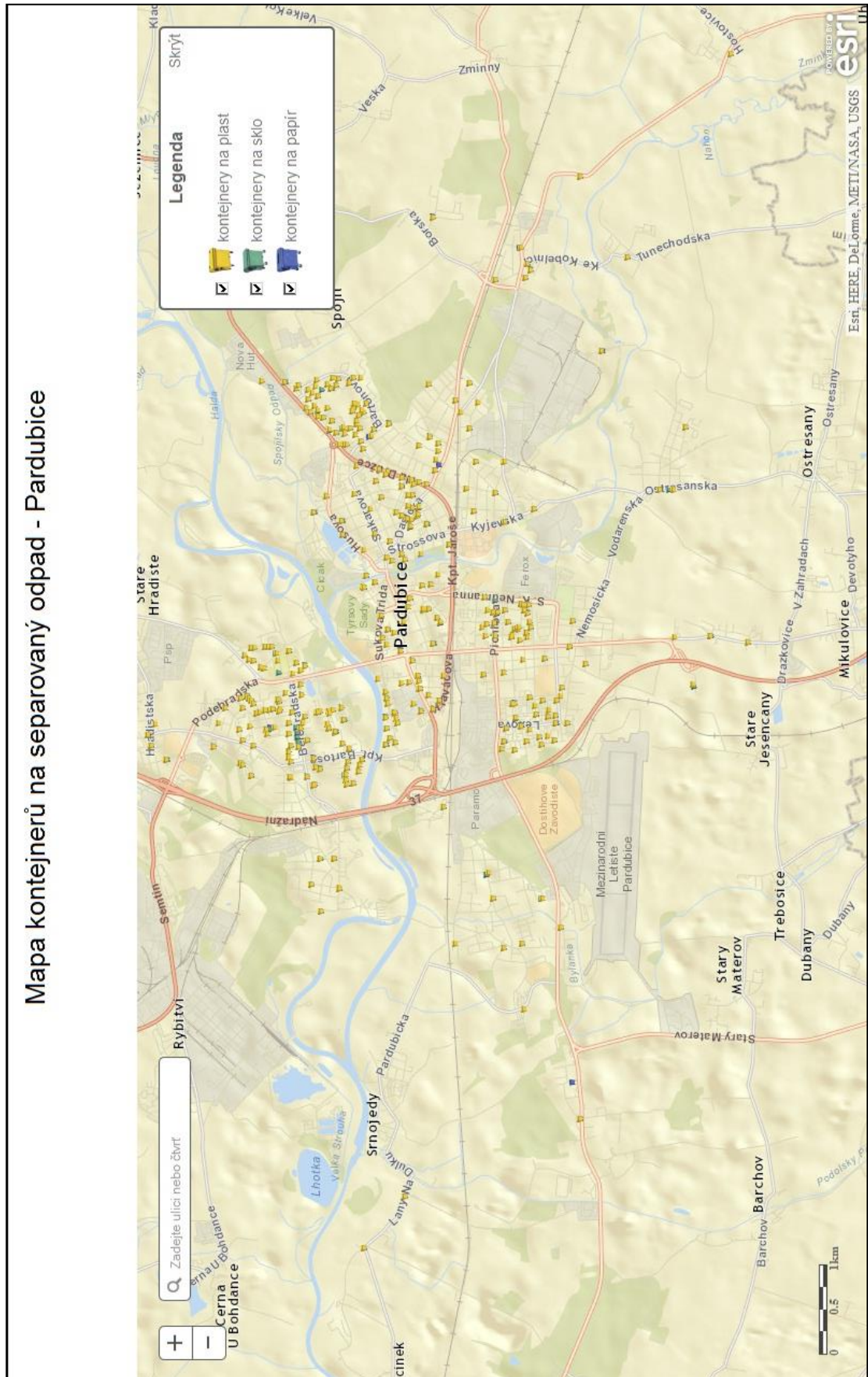
89. JAVOREK, Honza. API k českým turistickým mapám. DEVEL.CZ LAB S.R.O. Zdroják, o tvorbě webových stránek a aplikací [online]. 2010 [cit. 2014-04-26]. Dostupné z: <http://www.zdrojak.cz/clanky/api-k-ceskym-turistickym-mapam/>
90. Beta API ve verzi 4.2. SEZNAM.CZ, a.s. Blog Mapy.cz [online]. 2010 [cit. 2014-04-26]. Dostupné z: <http://mapy.cz.sblog.cz/2010/07/>
91. Changelog – seznam změn mezi verzemi. SEZNAM.CZ, a.s. Mapy.cz [online]. 2014 [cit. 2014-04-26]. Dostupné z: <http://api.mapy.cz/view?page=changelog>
92. Příkladová stránka widgetu Tree. SEZNAM.CZ, a.s. JAK: JavaScriptová knihovna [online]. 2014 [cit. 2014-04-26]. Dostupné z: <http://jak.seznam.cz/example/widgets/tree/>
93. Návod k použití API. SEZNAM.CZ, a.s. Mapy.cz [online]. 2014 [cit. 2014-04-26]. Dostupné z: <http://api.mapy.cz/view?page=instruction>
94. ZANDL, Patrick. Seznam označuje zprávy o porážce Googlem za zavádějící, poukazuje na potíže s metodikou. LUPA.CZ. Lupa.cz [online]. 2014 [cit. 2014-04-26]. Dostupné z: <http://www.lupa.cz/clanky/seznam-oznacuje-zpravy-o-porazce-googlem-za-zavadejici-poukazuje-na-potize-s-metodikou/>
95. Úvod. SEZNAM.CZ, a.s. Captcha api [online]. 2014 [cit. 2014-04-26]. Dostupné z: <http://captcha-api.seznam.cz/>
96. Ukázky práce s API. SEZNAM.CZ, a.s. Mapy.cz [online]. 2014 [cit. 2014-04-26]. Dostupné z: <http://api.mapy.cz/view?page=examples>
97. Seznam tříd. SEZNAM.CZ, a.s. Mapy.cz [online]. 2014 [cit. 2014-04-26]. Dostupné z: <http://api.mapy.cz/doc/index.html>
98. DAHLSTRÖM, Erik et al. Scalable Vector Graphics (SVG) 1.1. W3C. Captcha api [online]. SVG 1.1 (Second Edition). 2011 [cit. 2014-04-26]. Dostupné z: <http://www.w3.org/TR/SVG11/Overview.html>
99. Enable cross-origin resource sharing: Why is CORS important?. ENABLE CORS. Enable CORS [online]. 2014 [cit. 2014-04-26]. Dostupné z: <http://enable-cors.org/>
100. Mapy API. SEZNAM.CZ, a.s. Mapy.cz [online]. 2014 [cit. 2014-04-26]. Dostupné z: <http://ondras.zarovi.cz/slides/mapyapi/#>

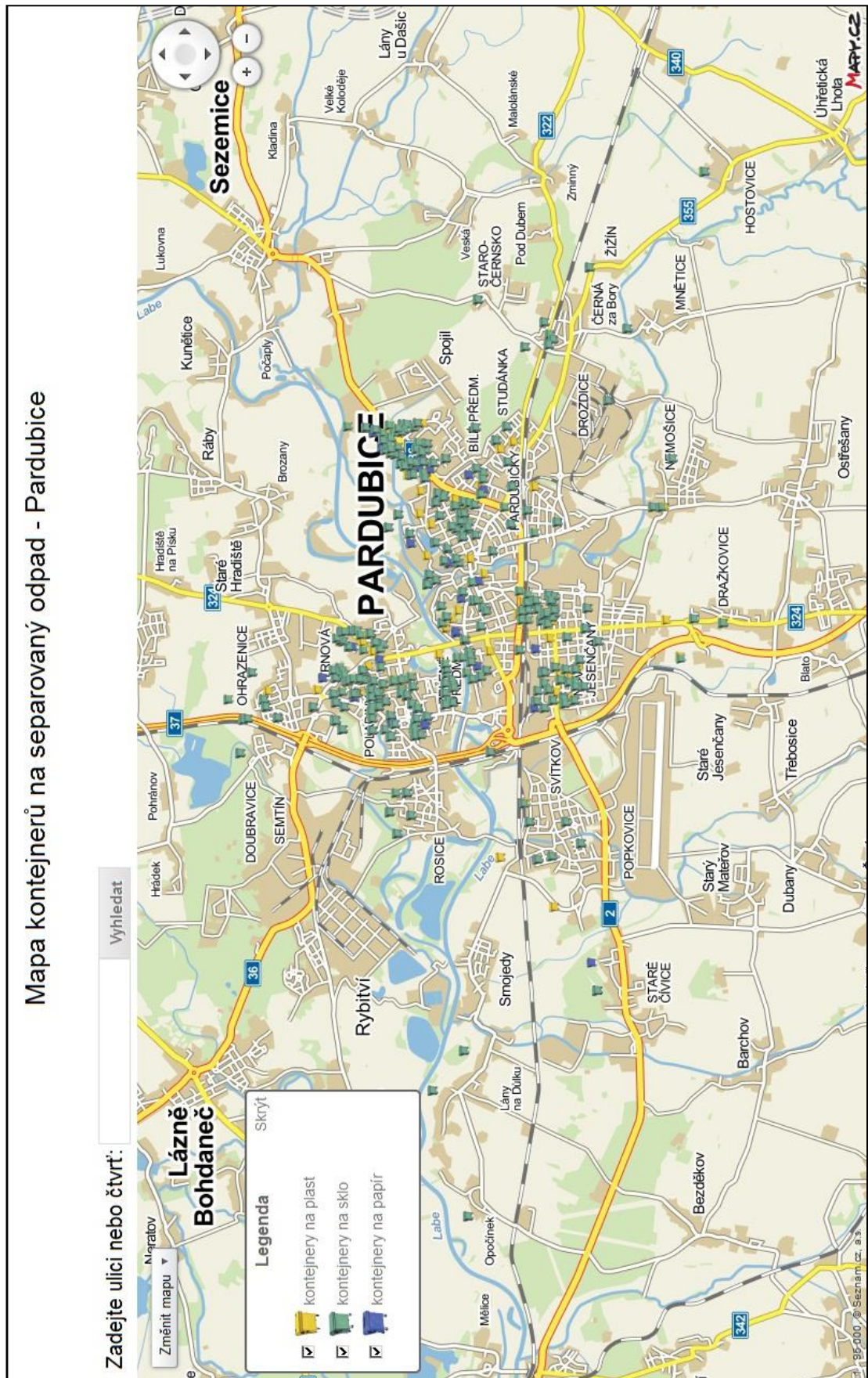
101. TUUPOLA, Mika. *Introduction to Marker Clustering With Google Maps* [online]. 2014 [cit. 2014-04-28]. Dostupné z: <http://www.appelsiini.net/2008/introduction-to-marker-clustering-with-google-maps>
102. JELÍNEK, Lukáš. *Generátor výškového profilu*. Praha, 2008. Bakalářská práce. ČVUT.
103. Great circle distance between 2 points. *Movable Type Scripts* [online]. 2014 [cit. 2014-04-28]. Dostupné z: <http://www.movable-type.co.uk/scripts/gis-faq-5.1.html>
104. Mapy API: Kontextová nabídka. ETHAN.CZ. Ethan.cz [online]. 2013 [cit. 2014-04-26]. Dostupné z: <http://ethan.cz/2013/03/06/kontextova-nabidka>
105. Can I use Cross-Origin Resource Sharing?. FYRD. Cross-Origin Resource Sharing [online]. 2014 [cit. 2014-04-26]. Dostupné z: <http://caniuse.com/cors>
106. ROZSYPAL, Petr. Které mapy jsou pro váš web nejlepší?. Lupa.cz [online]. 2007 [cit. 2014-04-26]. Dostupné z: <http://www.lupa.cz/clanky/ktere-mapy-jsou-pro-vas-web-nejlepsi/>
107. KUBANOVÁ, Jana. *Statistické metody pro ekonomickou a technickou praxi*. Vydání třetí - doplněné. Bratislava: Statis, 2008, 247 s. ISBN 978-80-85659-47-4.
108. WOOD, Harry. OpenStreetMap in Google Maps v3 API. Harry Wood [online]. 2014 [cit. 2014-04-26]. Dostupné z: <http://harrywood.co.uk/maps/examples/google-maps/apiv3.view.html>
109. Podrobná mapa 1 : 10 000. MAPY.CZ, s.r.o. Mapy.cz Seznam.cz: Váš dodavatel map, databází a mapových řešení [online]. 2014 [cit. 2014-04-26]. Dostupné z: <http://firma.mapy.cz/podrobna-mapa-1-10-000.html>
110. KŘUPKA, Jiří, Miloslava KAŠPAROVÁ a Renáta MÁCHOVÁ. *Rozhodovací procesy*. *Rozhodovací procesy* [online]. 2011 [cit. 2014-04-26]. Dostupné z: <http://www.rozhodovaciproceny.cz/>

PŘÍLOHY

PŘÍLOHA A – <i>Vizuální rozhraní aplikace vytvořené pomocí Google Maps API</i>	79
PŘÍLOHA B – <i>Vizuální rozhraní aplikace vytvořené pomocí ArcGIS API</i>	80
PŘÍLOHA C – <i>Vizuální rozhraní aplikace vytvořené pomocí Mapy API</i>	81
PŘÍLOHA D – <i>CD - Výpočty a soubory aplikace</i>	82







PŘÍLOHA D *CD – Výpočty a soubory aplikace*

Součástí přiloženého CD je soubor MS Excel s detaily výpočtů a soubory aplikací.