



VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA INFORMAČNÍCH TECHNOLOGIÍ

FACULTY OF INFORMATION TECHNOLOGY

ÚSTAV INFORMAČNÍCH SYSTÉMŮ

DEPARTMENT OF INFORMATION SYSTEMS

**WEBOVÁ GRAFICKÁ VIZUALIZACE HIERARCHICKÝCH
GEOGRAFICKÝCH DAT**

WEB GRAPHICAL VISUALIZATION OF HIERARCHICAL GEOGRAPHICAL DATA

BAKALÁŘSKÁ PRÁCE

BACHELOR'S THESIS

AUTOR PRÁCE

AUTHOR

VEDOUcí PRÁCE

SUPERVISOR

VOJTĚCH MALÝ

Ing. JIŘÍ HYNEK, Ph.D.

BRNO 2022

Zadání bakalářské práce



Student: **Malý Vojtěch**
Program: Informační technologie
Název: **Webová grafická vizualizace hierarchických geografických dat**
Web Graphical Visualization of Hierarchical Geographical Data
Kategorie: Uživatelská rozhraní

Zadání:

1. Prostudujte problematiku zpracování a vizualizace geografických dat na webu a prozkoumejte existující technologie určené pro tento účel (např. Leaflet, d3-geo, Geovisto, apod.). Prostudujte jednotlivé typy tématických map.
2. Prostudujte existující API poskytující definice geografických objektů (GeoJSON) na různých úrovních detailu (např. kontinenty, státy, regiony, města, apod.).
3. Analyzujte požadavky uživatelů na uživatelsky přívětivé zobrazení hierarchických dat v tématických mapách při různých úrovních přiblížení (vhodné shlukování/agregace dat do větších logických celků dle odpovídajících zeměpisných celků).
4. Navrhněte model pro hierarchické uspořádání geografických objektů a promyslete jeho využití v existujících tématických mapách knihovny Geovisto.
5. Navržené rozšíření implementujte formou nového nástroje knihovny Geovisto využívající API pro rozšíření knihovny. V případě potřeby proveďte vhodné úpravy existujících nástrojů knihovny Geovisto.
6. Otestujte výsledné řešení na vhodných hierarchických datech získaných s využitím znalostí z bodu 2.

Literatura:

- Hynek, J., Kachlík, J. a Rusňák, V.: *Geovisto: A Toolkit for Generic Geospatial Data Visualization*. In *VISIGRAPP (3: IVAPP)* (pp. 101-111).
- Tlčina, A.: *Webová aplikácia pre definíciu grafických objektov na mape*. Brno, 2021. Diplomová práce. Vysoké učení technické v Brně, Fakulta informačních technologií.
- Dent, B., D., a spol.: *Cartography: Thematic Map Design*. McGraw-Hill Higher Education 2009, ISBN 978-128-3388-023.
- Butler, et al. *The GeoJSON Format* [online]. Internet Engineering Task Force (IETF). 2016 [cit. 2021-10-09]. Dostupné z: <https://datatracker.ietf.org/doc/html/rfc7946>
- Leaflet: *Leaflet API reference* [online]. 2021 [cit. 2021-10-09]. Dostupné z: <https://leafletjs.com/reference-1.7.1.html>

Pro udělení zápočtu za první semestr je požadováno:

- Body 1 až 4.

Podrobné závazné pokyny pro vypracování práce viz <https://www.fit.vut.cz/study/theses/>

Vedoucí práce: **Hynek Jiří, Ing., Ph.D.**
Vedoucí ústavu: Kolář Dušan, doc. Dr. Ing.
Datum zadání: 1. listopadu 2021
Datum odevzdání: 11. května 2022
Datum schválení: 18. října 2021

Abstrakt

Cílem této práce je vytvořit nástroj, který umožní využít při vytváření webových geografických vizualizací, hierarchických vztahů mezi geografickými objekty. A to například formou rozpadnutí či shlukování různých administrativních celků, jako jsou okresy nebo kraje, na základě přiblížení mapového podkladu. Uživateli tím poskytne další možnosti pro vytváření efektivnějších geografických vizualizací, které mohou zobrazované jevy ukázat v jiných úhlech pohledu. Nástroj také poskytne uživatelské prostředí pro stahování datové reprezentace geografických objektů pomocí API a umožní agregaci dat podle jednotlivých úrovní hierarchie. Tento nástroj je volitelnou součástí aplikace Geovisto, která poskytuje prostředky pro vizualizaci generických dat v prostředí webových prohlížečů na různých tematických mapách.

Abstract

This work aims to create a tool that can be used in creating web geographic visualizations with the use of hierarchical relationships between geographical objects. This is done for example by disintegration or clustering of various administrative units such as districts or regions based on different zoom levels of the map. This will provide the user with additional options for creating more effective geographic visualizations that can show the displayed phenomena from different angles. Tool will also provide a user interface for downloading the data representation of geographic objects using API and will allow aggregation of data according to individual levels of the hierarchy. This tool is an optional part of Geovisto, which provides tools for the visualization of generic data in web browsers on various thematic maps.

Klíčová slova

geografické vizualizace, tematické mapy, administrativní celky, hierarchie, Geovisto

Keywords

geographical visualization, thematic maps, administrative units, hierarchy, Geovisto

Citace

MALÝ, Vojtěch. *Webová grafická vizualizace hierarchických geografických dat*. Brno, 2022. Bakalářská práce. Vysoké učení technické v Brně, Fakulta informačních technologií. Vedoucí práce Ing. Jiří Hýnek, Ph.D.

Webová grafická vizualizace hierarchických geografických dat

Prohlášení

Prohlašuji, že jsem tuto bakalářskou práci vypracoval samostatně pod vedením pana Ing. Jiřího Hynka Ph.D. Uvedl jsem všechny literární prameny, publikace a další zdroje, ze kterých jsem čerpal.

.....
Vojtěch Malý
9. května 2022

Poděkování

Děkuji především svému vedoucímu, za mnoho cenných rad a připomínek, velmi přátelský přístup a obrovské množství času, které mi poskytl při psaní této práce. Dále bych chtěl také poděkovat svým rodičům za podporu při celém mém studiu.

Obsah

1	Úvod	3
2	Geografická data	5
2.1	Geografická poloha	5
2.2	Geografické objekty	6
2.2.1	Datová reprezentace	7
3	Geografická vizualizace	12
3.1	Tematické mapy	12
3.2	Webová geografická vizualizace	15
3.2.1	Přístupy k prezentaci webových vizualizací	15
3.2.2	Programové knihovny	16
3.2.3	Autorské systémy	18
3.2.4	Geovisto	20
4	Administrativní celky a jejich hierarchické dělení	22
4.1	Dělení administrativních celků	23
4.1.1	ISO 3166	24
4.1.2	NUTS	24
4.2	API poskytující definice objektů	25
4.2.1	Overpass API	25
4.2.2	GeoNames	25
5	Analýza	27
5.1	Uživatelé	27
5.2	Vstupy	28
5.2.1	Chybějící vstupy a jejich stažení	28
5.3	Současná řešení	29
5.4	Proces použití	29
5.5	Shrnutí	30
6	Návrh	31
6.1	Definice hierarchie geografických objektů	31
6.1.1	Úroveň přiblížení	33
6.2	Nástroj hierarchie	34
6.3	Nástroj pro stažení geografických objektů	35
7	Implementace	36

7.1	Nástroj hierarchie	36
7.1.1	Podpora mapových vrstev	38
7.2	Nástroj pro stažení objektů	40
7.2.1	OverpassAPI	41
7.2.2	Použití nástroje	42
8	Testování	44
8.1	Nástroj hierarchie	44
8.1.1	Testování mapových vrstev	44
8.1.2	Uživatelské testování	45
8.2	Nástroj pro stažení objektů	46
9	Závěr	48
	Literatura	49

Kapitola 1

Úvod

V dnešní době vzniká velké množství nových statistických dat. Může se jednat o počty nakažených obyvatel určitou nemocí, počet aut na rodinu nebo i údaje o výši minimálního příjmu. V některých případech jsou také jednotlivé záznamy v datových souborech opatřeny informací o geografické poloze. Taková data je následně vhodné zobrazit na mapovém podkladu a doplnit o další grafické prvky, které dokáží danou problematiku lépe přiblížit běžnému uživateli a umožní z nich získat mnohem větší informační hodnotu. Vytvořená geografická vizualizace se tak stane prostředkem pro předávání informací v širším kontextu dalšímu publiku. Dnes je jedním z nejpoužívanějších prostředků pro předávání takových vizualizací internet, a proto se v této práci budu věnovat zejména geografickým vizualizacím ve webovém prostředí.

Vizualizace většinou pracují s geografickými prvky jako jsou body, křivky nebo polygony, ze kterých se následně vytváří pohledy vrcholů kopců, cest nebo hranic území. Existují ale i geografické objekty, jako jsou státy, kraje či okresy, které mezi sebou mají i určité hierarchické vazby. Například určité okresy jsou součástí jednoho kraje a jednotlivé kraje zase územími určitého státu. Tyto objekty jsou tedy děleny do různých administrativních celků. Pokud zobrazíme datové hodnoty v jednotlivých administrativních úrovních, můžeme získat nové pohledy na celou zobrazovanou problematiku. Například při vizualizaci dat o průměrném platu osob v krajích zjistím, ve kterém žijí bohatší lidé. Pokud ale data zobrazíme v jednotlivých okresech, můžeme zjistit zase naprosto jinou skutečnost, třebaže nejbohatší lidé se shlukují ve velkých městech a podobně. Takové objekty je tedy v rámci vizualizace vhodné shlukovat či nechat rozpadnout na objekty vyšší či nižší hierarchické úrovně a to po přiblížení nebo oddálení mapového podkladu. Pro lepší pochopení čtenářem jsem přidal i dva obrázky zobrazující administrativní celky kraje, jeho podřízených okresů, které si lze prohlédnout na obrázku 1.1.

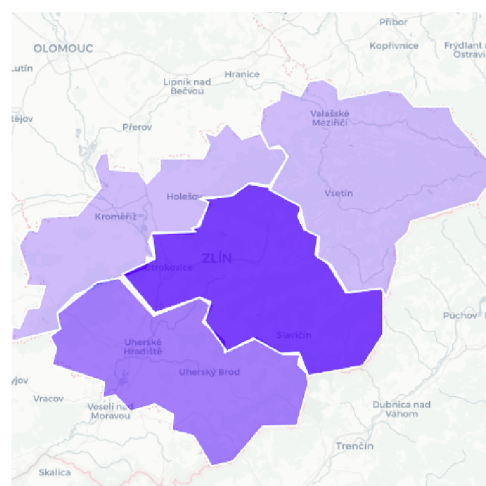
Podstatou této práce je tak implementace nástroje, který umožní uživateli definovat hierarchické vazby mezi různými geografickými objekty. Ty tak získají schopnost rozpadnout se na menší či větší části, podle uživatelem nastavených pravidel, a poskytnou mu tak nové možnosti k vytvoření webové vizualizace. To zlepší jeho kontrolu nad chováním celé vizualizace a zobrazenými informacemi, které chce předat publiku. Další funkcí vytvářeného nástroje bude i možnost stažení těchto geografických objektů pro různé státy v několika administrativních úrovních za pomoci webové API.

Ve druhé kapitole práce se zabývám problematikou zpracování geografických informací, objektů a jejich datové reprezentace. Ve třetí kapitole různými druhy tematických map, webovou geografickou vizualizací a systémy pro jejich vytváření. Ve čtvrté různým dělením administrativních celků do úrovní a také několika poskytovateli přístupových bodů API

pro stahování těchto objektů v jejich datové reprezentaci. V páté pak analyzuji uživatele vytvářeného nástroje, jeho vstupy, současná řešení a procesy použití. Nakonec v šesté a sedmé kapitole samotným návrhem a implementací nástroje, který bude součástí aplikace Geovisto, jež poskytuje prostředky pro geografickou vizualizaci generických dat ve webovém prostředí. Na závěr v poslední kapitole pak testováním vytvořeného řešení.



(a) Území Zlínského kraje



(b) Okresy patřící Zlínskému kraji

Obrázek 1.1: Ukázka dvou administrativních úrovní geografických celků, mezi kterými existuje hierarchická vazba. A to taková, že jednotlivé okresy jsou nižší administrativní celky kraje.¹

¹Vytvořeno pomocí aplikace Geovisto.

Kapitola 2

Geografická data

Geografická data, nebo také někdy nazývaná geodata či geoprostorová data, jsou informace pevně spjaté s určitým místem v reálném světě. Taková data můžeme pokládat za nositele geografické informace. Dle databáze národní knihovny ČR [18] jsou geografické informace jevy explicitně nebo implicitně přidružené k místu na Zemi a zahrnují data týkající se vzhledu těchto jevů. A to převážně popisem polohy, geometrického tvaru a topologických vztahů.

Z historie víme, že tyto informace jsou potřebné pro lidstvo již od pradávna. V dnešní době pak mají mapy obrovské množství využití, nejvíce na poli navigace pro dopravní prostředky, ale i v rozdělení správních celků, pozemků nebo při vizualizaci jiných, geograficky závislých dat. Jejich hlavní podstatou je ale abstrakce či zjednodušení okolního světa tak, aby bylo možné chápat naše okolí jako větší prostor než na který sami dohlédneme.

V úvahu ale musíme vzít dynamiku okolního prostoru, který se každým momentem mění, a proto využíváme geografických informací pouze u objektů, které jsou z krátkodobého časového hlediska neměnné a z našeho pohledu tedy stálé. To se ale taktéž s technologickým vývojem mění a máme dnes již prostředky pro sledování polohy v reálném čase. V informatice jsou obvykle geografická data reprezentována jako množina objektů s různými atributy definujícími základní vlastnosti objektu.

2.1 Geografická poloha

Poloha je základním stavebním kamenem celé disciplíny geografie. Vyjadřuje potřebnou informaci, a to polohu vůči reálné pozici v naší realitě. Ať už je to poloha auta, zakřivení říčního koryta nebo střed města. Právě z tohoto důvodu byla potřeba vytvořit dostatečně přesný, ale zároveň univerzální systém na značení takové polohy. Těchto pokusů bylo mnoho, některé úspěšnější a používáme je dnes běžně, jako je například systém GPS, a některé naopak neúspěšné. Samotné vyjádření polohy by se tak dalo rozdělit do dvou obecných kategorií.

Absolutní pozice

Známější formou vyjádření polohy je pozice absolutní. Jelikož existujeme v trojrozměrném prostoru, stačí nám tak k určení polohy maximálně tři hodnoty. Nejčastěji tedy pozice na ose X , Y a Z v kartézské soustavě souřadnic. V geografii si ale většinou vystačíme pouze se dvěma hodnotami, a to zeměpisnou šířkou φ (*angl. latitude*) a délkou λ (*angl. longitude*). Šířka se udává ve stupních, kde 0° odpovídá rovníku a 90° pólům. Rovněž délka se udává ve

stejných jednotkách, pouze nultý stupeň odpovídá Greenwichskému (nultému) poledníku. Pokud bychom se ale rozhodli zanést do mapy i informaci o výškových bodech, budeme potřebovat i třetí rozměr, a to nejčastěji nadmořskou výšku h . Tento způsob ale implicitně předpokládá, že popisovaný bod leží na zemském povrchu a samotná výška se považuje obvykle pouze za dodatečný atribut polohy [23].

Relativní pozice

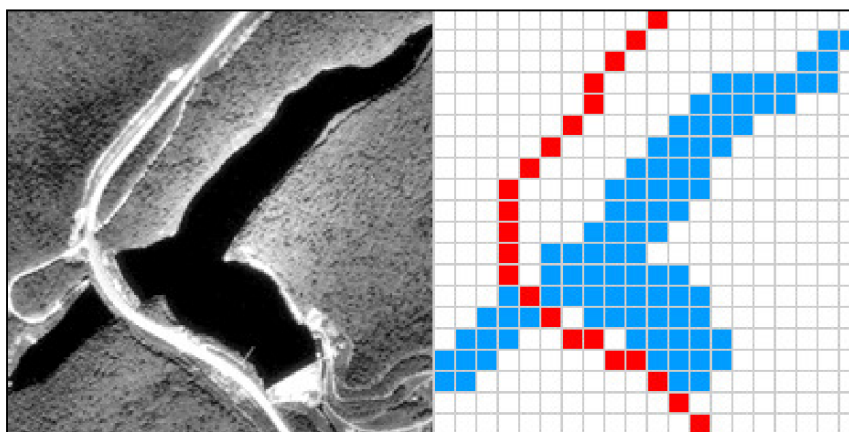
Určuje místo v závislosti na jiný objekt. Například vzdálenost a směr Prahy od Brna je relativním určením pozice Prahy, kdy nejsme schopni přesně říct, kde se Praha nachází bez znalosti pozice města Brna. Tento způsob, přestože většinou je přesnější než pozicování absolutní, se v informatice moc nepoužívá. Setkáme se s ním spíše v zeměměřičství a podobných oborech, kde se využívá k vytvoření lokálního souřadnicového systému, který bývá obvykle mnohem více přesnější.

2.2 Geografické objekty

Geografické objekty mohou mít mnoho podob a reprezentací, v této práci se ale budu zabírat pouze dvěma, a to grafickou nebo datovou.

Rastrová

Geografická informace je v případě rastrového zobrazení reprezentována jako pole hodnot, které představují jednotlivé pixely. Obvykle zbarvení resp. hodnota vyjadřuje různé vlastnosti daného místa, třeba zda je zde les, silnice či vodní plocha. Ukázkou takového případu lze vidět na obrázku 2.1. Výhodou takové reprezentace je možnost provádět různé analýzy a datové těžby. Nevýhodou naopak nepřesnost, která se odvíjí od podrobnosti a velikosti takového pole, jelikož jeden pixel zabere určitou část povrchu a tím zobecní vše, co se na jeho místě nachází, do jedné hodnoty [13]. Při velmi malé podrobnosti se tak může stát, že na mapě nebudou některé informace vůbec vidět. Třeba v případě, že jeden pixel obsahuje 51% lesa a 49% řeky, zobrazí se místo jako plně zalesněné.



Obrázek 2.1: Porovnání fotografie vodní plochy a její rastrové reprezentace.¹

¹Převzaté z:<https://www.e-education.psu.edu/geog160/node/1935>

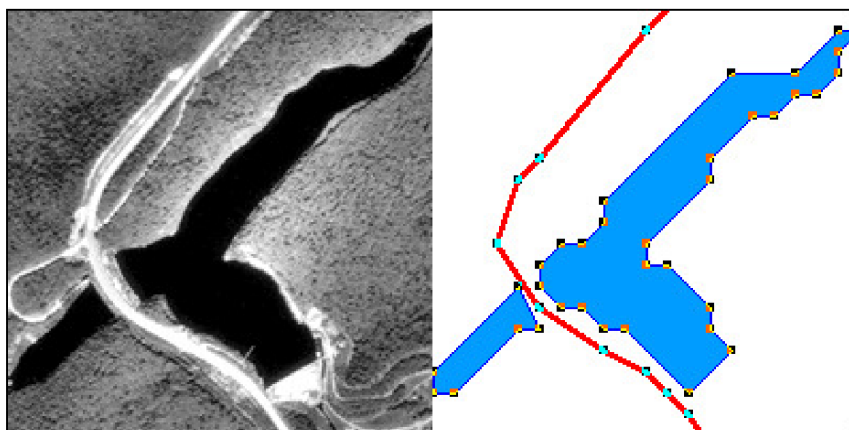
Vektorová

Vektorová reprezentace geografických objektů se obvykle vytvoří složením různých primitivních typů do větší celků. Příklad vektorové reprezentace geografických objektů si lze prohlédnout na obrázku 2.2. Tyto primitivní typy dělíme na tři různé druhy.

Bod – reprezentuje základní primitivní typ. K jejich vizualizaci nám postačí jen informace o poloze, kde se bod nachází. Používají se převážně pro označení malých objektů jako jsou lavičky, zastávky hromadné dopravy a další [8].

Úsečka – vzniká spojením dvou bodů, které mezi sebou mají již nějakou formu vztahu. Ve většině případů má každý bod úsečky svou absolutní souřadnici. Můžeme se ale setkat s přístupem, kdy má pouze první, počáteční bod absolutní souřadnici a další body jsou již pouze relativně pozicovány k tomuto místu. Mimo svou pozici může uchovávat již i další informace, jako svou délku nebo šířku. Využívá se například k definici cest či řek.

Polygon – je tvořen minimálně třemi úsečkami, které vytvářejí uzavřený objekt. Takový objekt pak jasně definuje určitou plochu, kterou pokrývá. Může mít i atribut uchováající informaci o své hloubce.



Obrázek 2.2: Porovnání fotografie vodní plochy a silniční komunikace převedené na její vektorovou reprezentaci. ²

2.2.1 Datová reprezentace

Přestože se odborná i laická veřejnost shodne na základních principech geometrické reprezentace, v případě reprezentace dat v elektronické podobě je již více rozdělená. Existuje velké množství formátů, které různými způsoby implementují ať už vektorovou či rastrovou podobu. Jejich velký, ale přesto nekompletní seznam si je možné prohlédnout na webové stránce projektu GISGeography³. Problém ale nastává v ostatních aspektech jako je výsledná velikost souboru, jeho zpracování, dělení na podsoubory, možnosti uložení atributů různého typu nebo jejich vzájemné provázání. S dynamicky se zvětšujícím odvětvím geografických informačních systémů navíc vznikají další a další formáty šité právě na míru dané

²Převzaté z:<https://www.e-education.psu.edu/geog160/node/1935>

³<https://gisgeography.com/gis-formats/>

aplikaci. Můžeme se tak setkat se speciálními formáty pro výškové informace, například CDED nebo i s formátem pro uložení map pro orientační sporty CAD. Převádění mezi jednotlivými formáty je tak ve většině případů velmi obtížné.

Vzhledem k povaze této práce se nebudu zabírat formáty pro reprezentaci rastrových dat, které se ve světě informatiky používají téměř vždy pouze pro uložení satelitních snímků nebo starých dobových map jako podklad pro novější geografická data. Stejně tak existují i specializovaná řešení, například geografické databáze, mapové dlaždice nebo specifické formáty pro použití při vytváření či kreslení map a geografických objektů. V následujících podkapitolách jsou popsány nejznámější formáty pro reprezentaci geografických dat.

ESRI Shapefile

Tento formát je považován za jeden z nejrozšířenějších, a to převážně pro svou dlouhou dobu existence již od počátku 90. let 20. století. Dle [10] Shapefile ukládá netopologickou geometrii a přidružené atributy pro prostorová data v datové sadě. Samotný objekt je pak reprezentován tvarem složeným z množiny souřadnic. Tento formát nepodporuje uložení topologické struktury, což vede k rychlejšímu vykreslení či úpravě dat. Problém ale nastává v samotném dělení formátu do souborů. Shapefile totiž není reprezentován pouze jedním souborem s příponou `.shp`, který reprezentuje seznam záznamů s informací o pozici, ale je dle technické dokumentace nutné jej doplnit o další dva soubory. Těmi jsou indexový soubor s příponou `.shx` který propojuje objekty s jejich atributy obsaženými v druhém souboru `.dbf`.

Samotný formát je modulární a dá se doplnit o různé typy souborů přidávající další funkčnost a informace jako třeba prostorové indexy prvků nebo popis projekce. Tento formát je regulován a vyvíjen firmou ESRI, tedy jej není možno upravit a přizpůsobit pro vlastní potřebu.

GML – Geography Markup Language

Geography Markup Language je standard pro definici geografických dat pomocí známé notace XML [21]. O správu této definice se stará Open Geospatial Consortium, zkráceně OGC. Data jsou tedy uložena čistě v textové podobě, což činí soubor čitelný jak pro člověka, tak pro počítač. Díky této reprezentaci je možné datovou množinu jednoduše posílat i po internetu a využívá se tedy převážně u webových aplikací. Jednoduše se upravuje a vytváří ucelenou logickou strukturu dat, ve které mohou být definovány i atributy objektů nebo statistická data potřebná pro vizualizaci.

Zajímavou vlastností je propojení s obrazovým formátem *JPEG*, které umožňuje pomocí GML namapovat `.jpeg` soubor na mapový podklad. V ukázce 2.1 je vytvořen základní trojstranný polygon a lze z ní odvodit i další informace, jako je kódování souboru nebo použitý souřadnicový systém.


```

1 <?xml version="1.0" encoding="utf-8" ?>
2 <ogr:FeatureCollection>
3   <gml:featureMember>
4     <ogr:map fid="map.0"><ogr:geometryProperty>
5       <gml:Polygon srsName="EPSG:4326">
6         <gml:outerBoundaryIs><gml:LinearRing>
7           <gml:coordinates>
8             17.5053405761719,49.1376980786619
9             17.9983520507812,49.3412305125646
10            17.3309326171875,49.3295971091282
11            17.5053405761719,49.1376980786619
12          </gml:coordinates>
13        </gml:LinearRing></gml:outerBoundaryIs>
14      </gml:Polygon>
15    </ogr:geometryProperty></ogr:map>
16  </gml:featureMember>
17 </ogr:FeatureCollection>

```

Výpis 2.1: Ukázka definice polygonu ve formátu GML.⁴

GeoJSON

GeoJSON je formát pro reprezentaci geografických objektů postavený na „*Javascript Object Notation – JSON*“. Jeho přesnou formu definuje RFC7946 [7] a popisuje jej jako formát pro výměnu geoprostorových dat využívající referenční souřadnicový systém WGS1984. Navíc jsou zde specifikována i pravidla pro zápis jednotlivých prvků („*feature*“) a jejich vlastností. Tento formát je plně kompatibilní se zmíněným formátem JSON, čímž se velmi snadno používá v prostředí webových prohlížečů a je poměrně přehledný i pro běžného člověka.

GeoJSON specifikuje několik geometrických typů:

- **Point** – reprezentace jednoho bodu na mapě pomocí souřadnic.
- **LineString** – jedná se o křivku, která je reprezentována seřazeným polem souřadnic.
- **Polygon** – uzavřený objekt o minimálně čtyřech bodech reprezentován pomocí typu **LineString**, který ale má první a poslední souřadnici identickou. Samotné souřadnice pak musí být seřazeny pomocí „pravidla pravé ruky.“
- **MultiPoint** – více bodů na mapě se stejnými vlastnostmi zahrnutý do jednoho objektu.
- **MultiLineString** – soubor více objektů křivek se stejnými vlastnostmi.
- **MultiPolygon** – soubor více polygonů se stejnými vlastnostmi.
- **GeometryCollection** – typ, který zahrnuje libovolný počet libovolných výše specifikovaných typů, a tím je sdružuje do větších objektů.

⁴Vytvořeno pomocí: <https://geojson.io/>

Všechny výše zmíněné geometrické typy se dají zapouzdřit do objektu „*Feature*“, který může reprezentovat například geoprostorová data fakulty. Několik takových objektů pak tvoří jeden celek typu „*FeatureCollection*“, který může definovat geografické objekty celé vysoké školy. Formát jasně udává názvy a specifikuje obsah všech atributů. V případě potřeby umožňuje přidat další dodatečné informace či uživatelem definované atributy pro které je předdefinována položka `properties`. Ta musí být umístěna pouze v objektu typu `Feature` a obsahuje další JSON objekt nebo nulovou hodnotu [7]. Nejčastějším využitím je uložení informací o oficiálním názvu, dodatečných identifikátorech nebo vizuálních proměnných objektu, jako je barva, hodnota, tvar a další. Ukázkou zápisu objektu ve formátu GeoJSON si lze prohlédnout ve výpisu 2.2, kde se jedná o stejný polygon jako v ostatních příkladech.

```
1 {"type": "FeatureCollection",
2   "features": [{
3     "type": "Feature",
4     "properties": {},
5     "geometry": {
6       "type": "Polygon",
7       "coordinates": [[[
8         17.505340576171875,
9         49.13769807866191
10        ], [
11         17.99835205078125,
12         49.341230512564564
13        ], [
14         17.3309326171875,
15         49.32959710912821
16        ], [
17         17.505340576171875,
18         49.13769807866191
19        ]]]]]}
```

Výpis 2.2: Ukázka polygonu ve formátu GeoJSON. ⁵

TopoJSON

Posledním formátem pro reprezentaci geografických objektů, který v této práci popíši, je TopoJSON. Topologický formát pro výměnu geoprostorových dat založený na GeoJSON. Topologie TopoJSON představuje jednu nebo více geometrií, které sdílejí sekvence pozic nazývané oblouky. Geometrie mohou obsahovat další vlastnosti pro kódování negeometrických dat [5]. Hlavní výhodou je především snížení redundance dat oproti formátu GeoJSON a s tím i spojená velikost výsledného souboru. K odstranění redundance jsou využívány tzv. `arcs`, což jsou oblouky definované v dedikovaném poli a jedná se o společné hranice vícero objektů. Na tyto oblouky je pak v samotných definicích objektů pouze odkazováno.

Další výhodou je využití celočíselných hodnot pro definici bodů. Číslo tak označuje pouze posun oproti minulému bodu a není nutné použít plovoucích čísel pro definici geografických souřadnic. Nevýhodou se ale stává nutnost využití základních grafických transformačních technik jako je posun, rotace nebo `scaling` pro plné navrácení tvaru do původní podoby. Ukázkou zápisu v tomto formátu si lze prohlédnout ve výpisu 2.3.

⁵Vytvořeno pomocí: <https://geojson.io/>

```

1 {
2   "type":"Topology",
3   "objects":{
4     "collection":{
5       "type":"GeometryCollection",
6       "geometries":[{
7         "type":"Polygon",
8         "arcs":[[0]]}],
9     "arcs":[[[
10      2613, 0
11      ],[
12      7386, 9999
13      ],[
14      -9999, -572
15      ],[
16      2613, -9427
17      ]]],
18   "transform":{
19     "scale":[
20       0.00006674861822119711, 0.00002035527891815698
21     ],
22     "translate":[
23       17.3309326171875, 49.13769807866191
24     ]
25   },
26   "bbox":[
27     17.3309326171875, 49.13769807866191,
28     17.99835205078125, 49.341230512564564
29   ]
30 }

```

Výpis 2.3: Ukázka polygonu ve formátu TopoJSON.⁶

⁶Vytvořeno pomocí: <https://geojson.io/>

Kapitola 3

Geografická vizualizace

Vše, co nás obklopuje, se nachází v prostoru. Takový geografický objekt, definovaný výše, ale nemusí mít nutně jen svou polohu. Do hry tak vstupují další dimenze, jako je čas, tlak, teplota a mnoho dalších, které nám pomohou lépe pochopit celý problém [9]. Ve světě geografických informačních systémů, ale nejen v nich, je potřeba tyto dimenze jednoduše vyjádřit. Ideálně tak, aby i běžný uživatel již na první pohled poznal, co vidí a jaké souvislosti se dají z daného pohledu vyvodit. Pohled na takováto data mu pak může okamžitě sdělit další informaci, jako třeba která trasa je nejrychlejší, nebo který okres má největší počet lékáren na jednoho obyvatele. K tomuto slouží jasně definovaná základní designérská pravidla a poučky.

Podle [6] je lidský zrak nejsilnějším systémem pro vnímání světa jaký známe. Kombinací technologií, jako je zpracování obrazu, počítačová grafika, animace, simulace, multimédia a virtuální realita, mohou počítače pomoci s prezentací informací v úplně jiném světle. Vždy tak, aby se daly najít nové závislosti, vzory a zlepšilo se celkové pochopení problému.

3.1 Tematické mapy

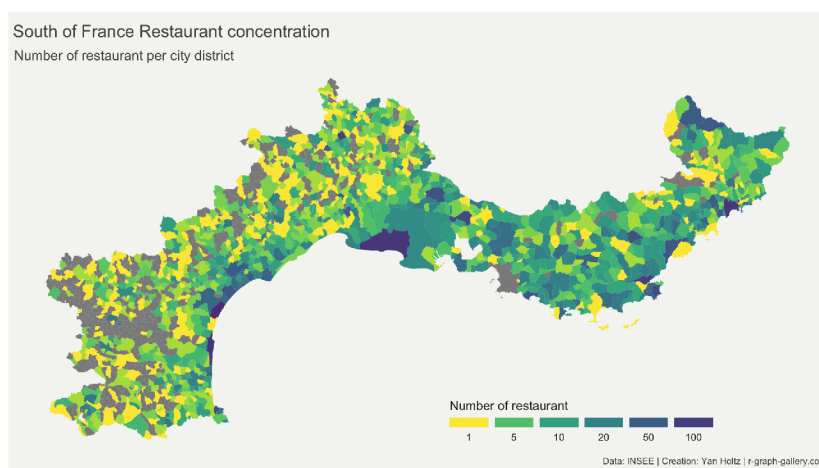
Tematická mapa je typ mapy kombinující referenční topografický podklad s dalšími společenskými objekty, jevy nebo jejich charakteristikami. Těmi jsou myšleny statistické informační údaje namapované na jednotlivé geografické objekty [22]. Vzniká nám tak mapa zaměřená převážně na sdělení určité specifické informace k danému tématu. Existuje mnoho různých druhů tematických map. Výběr druhu závisí vždy na tématu vizualizace tak, aby bylo informační sdělení co nejvíce efektivní. Při vytváření by se také mělo myslet na scénář finálního použití a druh publika, kterému bude mapa předkládána. O problematice výběru vhodné mapy pojednává průzkum [24], který doporučuje použití kartogramu na základě všeobecného podvědomí veřejnosti. Uznává ale mnohem efektivnější využití ostatních typů v jiných specifických případech.

Tematická mapa se obvykle skládá z několika vrstev. Základní vrstvou je samotný mapový podklad a nachází se vždy na nejnižší úrovni pohledu. Většinou je použita základní generická mapa, ve specifitějších případech se setkáme i se speciálními mapami, například vrstevnicemi či vodními plochami. Jako druhou vrstvu nalezneme samotnou vizualizaci dat. Tyto vrstvy se musí přesně překrývat, proto je u obou nutné stále udržovat i souřadnicový systém, pomocí kterého se podklad vycentruje s objekty vyšších vrstev.

V následujících podkapitolách představím základní druhy tematických map, se kterými se uživatel setká asi nejčastěji. Existuje ale velká řada dalších druhů, jako je mapa izolinii, dvouvariantního kartogramu nebo průtokových linek.

Kartogram

Kartogram je prvním druhem tematické geografické vizualizace, kterou popíši. Sledovaný jev je zde nejčastěji vyjádřen intenzitou či zabarvením určité oblasti. Setkáme se ale i s jiným grafickým prvkem, jako je šrafování. Mapovaný jev je tak většinou agregován do dostupných polygonů reprezentující oblasti podle své polohy ať už sumou, průměrem či jinou matematickou metodou. Výsledná mapa tak vyjadřuje zastoupení určitého jevu v určité oblasti. Kritickou součástí takové mapy je stupnice, která dává jasně najevo interval zobrazovaných hodnot a při jejímž vytváření by se nemělo zapomínat na další aspekty grafické reprezentace. Ať už je to správný výběr barvy, krok stupňování její intenzity nebo celkový dojem na čtenáře. [22] Špatně zvolená barva tak může způsobit mnoho nepříjemností a špatnou interpretaci celé mapy čtenářem. Ukázkou geografické vizualizace s využitím mapy kartogramu si lze prohlédnout v obrázku 7.2.



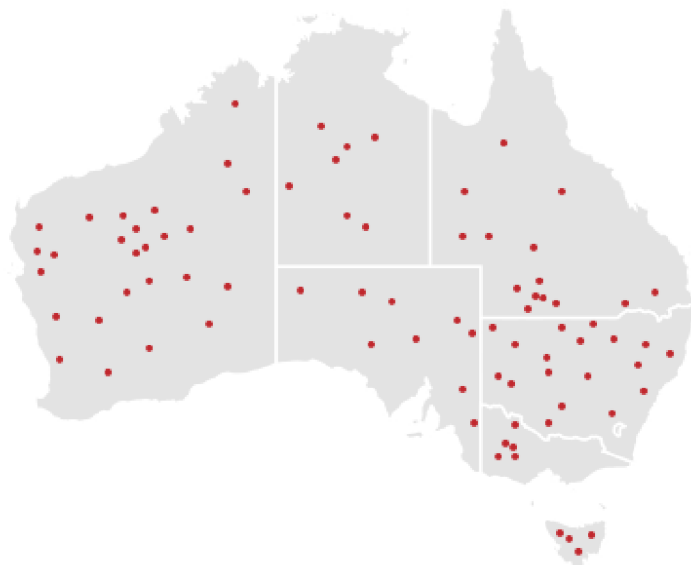
Obrázek 3.1: Ukázka kartogramu s daty o počtu restaurací na jihu Francie. ¹

Bodová mapa

Dle [17] bodové mapy znázorňují geografické rozložení jevů v oblasti, a to umístěním bodů představujících určitou veličinu zobrazované problematiky. Běžně se setkáváme se dvěma druhy takových map. Topografický přístup rozmístí stejný druh bodů po mapě a hlavním informačním sdělením je jejich rozmístění. Příklad takového přístupu si lze prohlédnout na obrázku 3.2. Naopak relativní hodnoty zobrazujeme pomocí kartogramického způsobu. Zde můžou být již hodnoty agregovány dohromady pomocí funkcí a jednotlivé položky datové sady mohou mít i více než jeden druh hodnoty. Výsledná mapa tak může obsahovat více symbolů, každý rozličného tvaru, barvy či velikosti. Tento způsob, na rozdíl od topografického, nevyžaduje přesné geografické umístění v mapě.

¹Převzato z <https://www.data-to-viz.com/graph/choropleth.html>

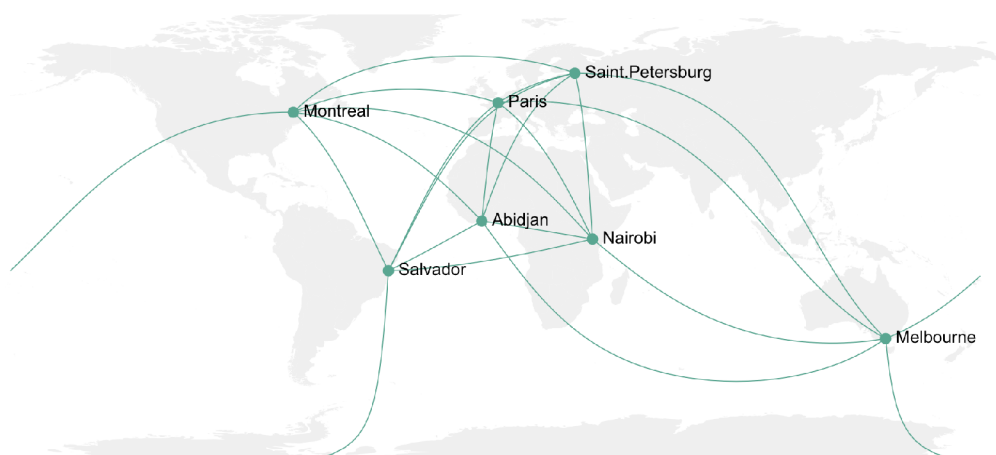
²Převzato z https://datavizcatalogue.com/methods/dot_map.html



Obrázek 3.2: Ukázka bodové mapy na území Austrálie.²

Mapa spojení

Dalším druhem tematické mapy je mapa spojení. Ta ke svému vytvoření potřebuje datovou sadu obsahující data s alespoň dvěma pozičními atributy. Ty se na mapě zobrazí obvykle jako bod a jsou následně propojeny čarou symbolizující určitý vztah mezi těmito body. Díky tomu mapa zobrazuje další informační hodnotu navíc a využije se třeba pro zobrazení leteckých mostů. Podobným případem je spojení opatřené šipkami místo čar, čímž se přidává i směr propojení mezi body. Ukázku si lze prohlédnout na obrázku 3.3.

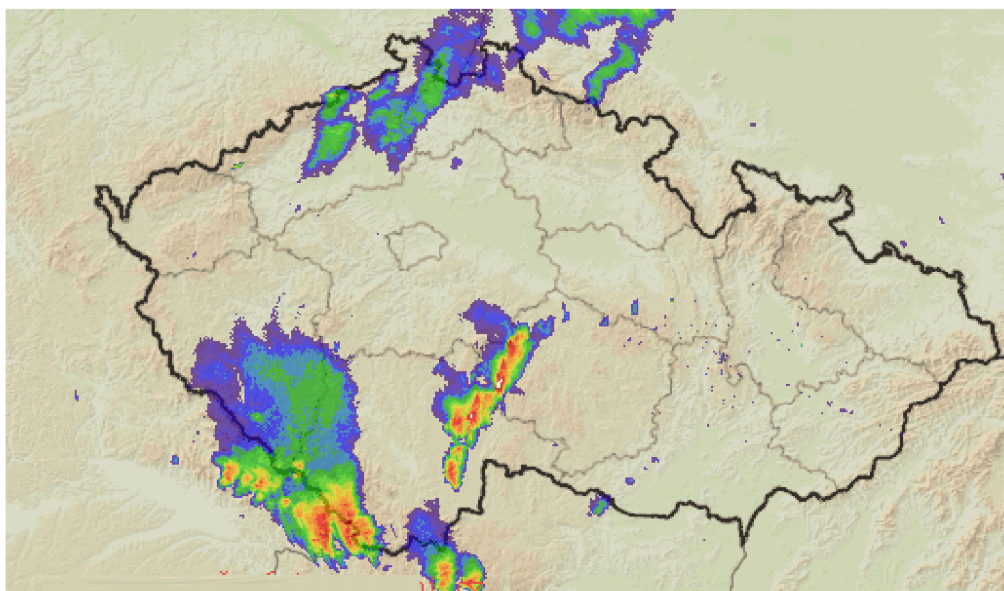


Obrázek 3.3: Ukázka mapy spojení mezi městy.³

³Převzato z <https://www.data-to-viz.com/story/MapConnection.html>

Teplotní mapa

Posledním druhem, který v této práci popíši, je teplotní mapa. Její příklad si lze prohlédnout na obrázku 3.4. Ústřední význam hraje barva, jejíž intenzita naznačuje místa s nejpočetnějším výskytem sledovaného jevu. Využívá se především pro vizualizaci dat ze sledování pohybu osob či zvěře. „Zatímco kartogram a bodové mapy jsou založeny na agregovaných statistických datech, spojených s diskrétními výčtovými jednotkami (např. kraje), Tepelné mapy nejsou agregované přes předem definované, samostatné oblasti.“ [26]



Obrázek 3.4: Ukázka využití teplotní mapy pro zobrazení předpovědi počasí.⁴

3.2 Webová geografická vizualizace

S geografickými materiály se v dnešní době nesetkáme už jen ve školních atlasech nebo papírových infografikách, ale převážně na internetu či v jiném software a aplikacích. Přesunutím do webového prostředí může autor takové vizualizace umožnit čtenáři lepší pochopení celé problematiky, využít interaktivních prvků, které toto prostředí nabízí, a v neposlední řadě dosáhnout většího publika nebo zajistit pravidelnou a jednoduchou aktualizaci vizualizovaných informací. Zároveň se výrazně zlepší i dostupnost, jelikož uživatel stačí pouze webový prohlížeč s připojením k internetu a není nutné instalovat specifickou aplikaci. Vzhledem k povaze práce se budu v této kapitole zabývat pouze vizualizací ve webovém prostředí a popsáním hlavních přístupů k jejich vytváření. A to za použití již existujících programových knihoven nebo autorských systémů, které se pokusím mezi sebou porovnat.

3.2.1 Přístupy k prezentaci webových vizualizací

Webové technologie zažívají v posledních letech velkého a rychlého vývoje a s ním je spojena i změna přístupů k prezentaci geografických vizualizací ve webovém prostředí. Z historického hlediska byly nejvíce využívány serverové (*angl. server-side*) aplikace, díky kterým nebyl

⁴Převzato z <https://www.meteoradar.cz/predpoved-radar.php>

kladen velký nárok na zdroje uživatele. Hlavní výhodou je kompletní odstřížení uživatele od velmi velkých, geografických dat a aplikační logiky. Tento přístup se ale s postupem času mění, jelikož uživatelé disponují již výkonnějšími zařízeními a větším množstvím dostupných zdrojů. Při využití klientské aplikace se větší množství vykonaných operací přesouvá do prohlížeče uživatele a je zde větší využití jazyka JavaScript.

Základním stavebním kamenem webové aplikace je značkovací jazyk HTML, definující strukturu zobrazovaného dokumentu. V závěru je tedy každá webová aplikace nucena využít tohoto jazyka a právě jeho možnosti značně definují typ výsledné vizualizace. Existuje mnoho rozdělení těchto přístupů. Podle [4] je nejznámější klasifikací ta, která dělí mapy na statické či dynamické a ty následně na kategorie map pouze pro zobrazení nebo s interaktivními prvky.

Prvním a nejjednodušším přístupem je vložení statické mapy. Většinou je tento přístup zvolen v kombinaci se serverovou aplikací, kdy je na základě požadavků uživatele vytvořena na vzdáleném serveru korespondující vizualizace a ve formě obrázku odeslána zpět uživateli. Ten je následně zobrazen pomocí základních HTML značek jako je například ``. Použití tohoto přístupu je sice poměrně jednoduché, nese s sebou ale velké množství nepříjemností, které musí programátor vyřešit. Převážně pak rozlišení obrázku spojené s jeho rastrovou reprezentací a v dnešní době mobilních zařízení i potřebu responzivního designu. Naopak výhodou je možnost využití nevektorových materiálu, jako jsou letecké snímky země.

Částečným řešením problému je využití pokročilejších možností jazyka JavaScript a HTML značky `<canvas>`. Ta poskytuje základní API pro vykreslování rastrových obrazců a její podpora je implementována již ve většině běžných internetových prohlížečů. Aplikace se tak dotazuje asynchronními požadavky serveru, ze kterého získává pouze základní data, a o správné zobrazení se stará již uživatelská strana. Aplikace tak dostává prostor pro využití dalších webových technologií a s nimi spojené interaktivní prvky pro ovládání vizualizace.

V dnešní době asi nejrozšířenějším způsobem geografických vizualizací je použití vektorové grafiky a s ní spojeného standardu SVG. Dle [20] je SVG značkovací jazyk založený na XML. Používá se k popisu dvourozměrné vektorové grafiky popřípadě rastrové grafiky a textu. Její popis je velmi sémanticky bohatý a vysoce strukturovaný. Díky tomu lze jednotlivé objekty seskupovat, stylizovat a skládat do objektů vyšší úrovně. Kromě vysoké grafické kvality nabízí i další funkce, jako je použití při skriptování, animace a další. Díky své implementaci nad standardem XML je taktéž velice jednoduše vložitelný do HTML stránek pomocí `<svg>` tagu nebo rozšiřitelný o další, uživatelem definované prvky.

3.2.2 Programové knihovny

Na internetu je k dispozici velké množství programovacích knihoven poskytujících základní rozhraní pro vytváření webových geografických vizualizací a jejich následné vložení do webových stránek. Použití také poskytuje větší prostor pro soustředění se na samotnou vizualizaci a usnadnění velkého množství práce. Vyžadují ale vyšší znalost programování a webových technologiích obecně. Výběr správné knihovny velmi závisí na očekávaném výsledku. Mnoho knihoven totiž poskytuje pouze abstrakci vykreslování, tedy rozhraní pro vylepšenou manipulaci s SVG grafikou či vykreslováním rastrového výstupu. Naopak existují knihovny, které nabízí velkou škálu funkcí pro stahování mapových podkladů, vykreslování složitých objektů nebo výpočty vzdáleností a transformace mezi různá zobrazení.

Většina těchto programových knihoven je napsána ve funkcionálním programovacím jazyce, JavaScriptu. Jeho hlavní využití je při vytváření a manipulaci s HTML dokumenty

s použitím vestavěného DOM API. Je vyvíjen podle ECMAScript⁵ standardu a je taktéž řízen událostmi. To znamená, že je schopen spustit určité části svého kódu jako reakci na uživatelský vstup. Tím může být stisk tlačítka, zadání hodnoty do formuláře, ale i změna velikosti zobrazované stránky. V posledních letech se také setkáme s větším využitím nové technologie zvané TypeScript. Jedná se open-source programovací jazyk vyvíjený firmou Microsoft jako nadstavba nad již hojně rozšířeným JavaScriptem. Přidává k využití statické typování a další nástroje pro lepší objektově orientované programování. Jazyk je zpětně kompatibilní s JavaScriptem, tudíž všechny kód je přeložitelný a spustitelný v dnešních webových prohlížečích. Stále větší množství projektů a knihoven je do tohoto jazyka překládáno a je vhodné při výběru programovací knihovny vzít tuto technologii v potaz.

V následujících subsekcích popíši tři knihovny, kde každá má trochu jiný přístup ke svému použití a liší se převážně jejich míra abstrakce celého problému. Existuje ale samozřejmě mnoho dalších programovacích knihoven, které poskytují jiné rozhraní nebo funkce. Ať už to je například projekt OpenLayers⁶, Google Maps Platform⁷ nebo Mapbox GL⁸.

D3.js

První knihovna, kterou popíši, se nazývá *D3.js*⁹ a její název je odvozen z anglického slova „Data Driven Documents“. Již z názvu je tedy patrné, že se jedná o generickou knihovnu využívanou pro manipulaci prvků dokumentu pomocí dat. Knihovna je napsána v jazyku JavaScript, obsahuje pokročilé funkce pro zpracování velkého množství dat a k vizualizaci využívá manipulaci s DOM elementy. Poskytuje i podporu pro animace, přechody a dynamickou manipulaci, díky čemuž je možné vizualizovat například data závislá na časové proměnné či různé objekty v pohybu. Knihovna obsahuje velké množství dalších balíčků, které přidávají další funkcionalitu pro specifické potřeby. Pro geografické vizualizace se využívá modulu *d3-geo*¹⁰, který poskytuje základní funkcionalitu pro vytváření geografických vizualizací. Mezi hlavní patří zobrazení různých geografických objektů, jako jsou polygony nebo body ve formátech GeoJSON a TopoJSON. Nad objekty je tak možné vykonávat různé matematické operace, převody mezi mapovými projekcemi či dynamicky měnit jejich styl zobrazení na základě vstupních dat.

Knihovna je nízkourovňová a neposkytuje žádné předdefinované rozhraní pro stažení a zobrazení mapových podkladů. Uživatel je tak nucen si vyšší funkcionalitu zajistit sám. Ve většině případů je proto D3 používána v kombinaci s dalšími knihovnami. I z tohoto důvodu je vhodná spíše pro uživatele, kteří chtějí využít pouze pokročilých animací a předdefinovaného stylu objektů a zbytek aplikace si naprogramovat sami.

Leaflet.js

Další knihovnou je Leaflet, open-source projekt vytvořený v roce 2010 a naprogramován v jazyce JavaScript. Dle oficiální dokumentace [2] má velikost pouze 39 KB, je využitelný i pro vytvoření map na mobilních zařízeních a obsahuje všechny funkce, které kdy většina vývojářů potřebuje. Po importování knihovny stačí definovat hlavní `<div>` element a předat odkaz přístupového bodu API, přes který dojde ke stažení mapových dlaždic tvořících

⁵<https://www.ecma-international.org/>

⁶<https://openlayers.org/>

⁷<https://mapsplatform.google.com/>

⁸<https://docs.mapbox.com/mapbox-gl-js/api/>

⁹<https://d3js.org/>

¹⁰<https://github.com/d3/d3-geo>

nejnižší vrstvu zobrazované mapy. Sami vývojáři knihovny k tomuto doporučují *OpenStreetMap API*¹¹, lze ale využít i jiných poskytovatelů. Mapa vytvořená pomocí Leaflet.js se skládá z několika vrstev, se kterými může vývojář pracovat a nanášet na ně vlastní geografické objekty jako jsou polygony nebo body. A to převážně v implicitně podporovaném formátu GeoJSON. Jednotlivé zobrazené objekty následně mohou být provázané s událostmi, jejichž vyústěním může být například změna zobrazovaného místa na mapě, změna vizuálních proměnných objektu nebo zobrazení vyskakovacího okna, takzvaného „popup“.

Mimo to poskytuje transformační funkce objektů s podporou pro různé mapové projekce. Dokáže vytvořit základní ovládací prvky pro pohyb s mapovým podkladem či jeho přiblížením nebo oddálením. Jedná se tedy o projekt, který vytváří již více abstraktní přístup k webovým vizualizacím a umožňuje specifitější použití pro geografické vizualizace, jelikož je přímo na ně zaměřený, na rozdíl od již zmíněné knihovny D3. Výhodou je také rozšíření o různé pluginy. Díky své popularitě má velkou uživatelskou základnu, která vytváří podporu na internetových fórech a diskuzních skupinách.

Seznam Mapy API

Nástroj pro vizualizaci Mapy.cz a s ním spojená Mapy API¹² je programovací knihovna vytvořená společností Seznam.cz. Díky českému původu nabízí velmi detailní mapové podklady pro Českou a Slovenskou republiku. Ty jsou přístupné v různých verzích ať už s turistickou mapou, leteckými snímky nebo historickým podkladem. Velkou výhodou je také propojení s velmi detailní databází českých objektů, díky které má uživatel k dispozici vyhledávání pomocí klíčových slov.

Knihovna je psána v JavaScript a poskytuje velké množství ovládacích prvků pro uživatele od základních, jako jsou tlačítka pro přiblížení respektive oddálení mapy, až po minimapu či směrovou ružici, která ukazuje na sever mapy. V neposlední řadě taky poskytuje přístup pro ovládání panoramatických snímků a pohybu po nich. Použití knihovny je zdarma, ale to pouze v případě, že je výsledná aplikace veřejně dostupná a není zpoplatněná. V opačném případě je nutné zakoupit prémiovou verzi služby. Její hlavní výhodou tak spočívá při použití v českém prostředí, kde poskytuje podobné funkce a možnosti jako jiné knihovny zároveň s plně českou dokumentací a uživatelským prostředím.

3.2.3 Autorské systémy

Autorskými systémy rozumíme nástroje, které umožňují vytváření multimediálního obsahu pro další publikum. Dle [16] se pak jedná o programy umožňující autorům, tedy těm, kteří autorské systémy používají, propojení různých objektů, jako je text, ilustrace či píseň ve vhodném pořadí tak, že vzniká další atraktivní a užitečný obsah. Největší použití těchto systémů nalezneme převážně při vytváření internetového nebo e-learningového obsahu, kde poskytují zapouzdření HTML značek a programového kódu tak, aby byl systém použitelný i bez hlubší znalosti programátorských technik a zásad.

Existuje řada autorských systémů pro vytváření geografických vizualizací. Jejich hlavní výhodou je právě zmíněné oddělení od nižší úrovně vytváření takového obsahu, zároveň ale poskytují i základní rozhraní pro zpracování dat a možnosti úpravy výsledného produktu. Konfiguraci nástroje je možné také většinou uložit, čímž je umožněno znovupoužití stejné vizualizace nad jinou datovou sadou a vytvoření tak komplexnějšího obsahu, jako je

¹¹<https://wiki.openstreetmap.org/wiki/API>

¹²<https://api.mapy.cz/>

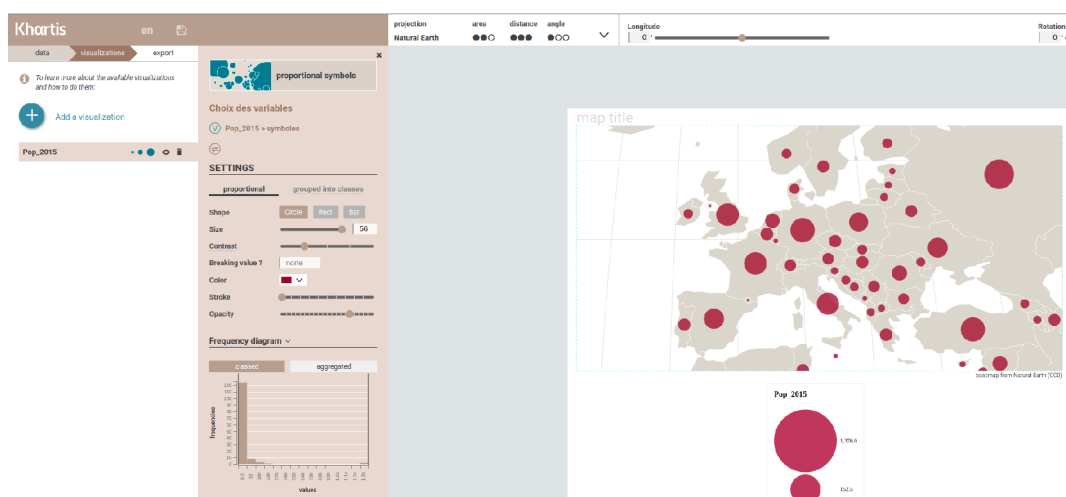
například mřížka několika stejných typů grafů zobrazující rozdíly mezi jednotlivými datovými sadami. Za nevýhodu je považováno omezení na použití pouze takových funkcí, jaké jsou v autorském systému implementovány. Autor tak nemusí mít k dispozici vždy takový nástroj, který mu umožní vytvořit výsledek přesně podle jeho představ.

Vzhledem k povaze této práce popíši tři různé systémy, které jsou dle mého názoru pro tuto práci důležité. Systém Khartis, jenž je svojí funkcionalitou velmi podobný aplikaci Geovisto, do které bude výsledný nástroj implementován. A dále aplikace geojson.io poskytující možnost vytvářet vlastní geografické objekty. Samozřejmě existuje velká řada dalších podobných autorských systémů.

Khartis

Khartis¹³ je open-source nástroj pro vytváření tematických map, jak deklaruji v konferenci o projektu [12], pouze ve třech jednoduchých krocích. Mimo on-line verzi, podporující základní funkcionalitu, je dostupná verze nástroje i pro operační systémy Linux, Windows a Mac. Prvním krokem je nahrání geografických objektů. Uživatel má možnost vybrat jeden z přednastavených setů geografických objektů nebo nahrát svůj vlastní ve formátu GeoJSON, TopoJSON nebo Shapefile. Nástroj tedy nezobrazuje samotný mapový podklad, ale pouze jednotlivé geografické objekty, obvykle polygony hranic územních celků. Následuje vložení datové sady ve formátu CSV.

Druhým krokem je výběr požadovaného typu vizualizace a typ mapové projekce. Autor má tak možnost rozhodnout se pro zachování rozlohy, vzdáleností či úhlů rozmístění objektů. V 3.5 lze vidět typ bodové mapy. Posledním krokem je nastavení barev, odstínů a značek, tedy nastavení výsledné vizuální podoby celé vizualizace. Výsledek je možné uložit pomocí exportování konfigurace, která je umožněna uložit pouze ve speciálním zakódovaném souboru s příponou .kh specifickým pro právě tuto aplikaci. Nástroj ale neposkytuje přímé generování kódu pro vložení do webových stránek a s tím i spojené uživatelské rozhraní vizualizace. Momentálně je přeložen pouze do francouzského a z části anglického jazyka. Použití najde především v případě, kdy je potřeba výsledek vytvořit velmi rychle bez nutnosti věnovat přílišnou pozornost maličkostem.



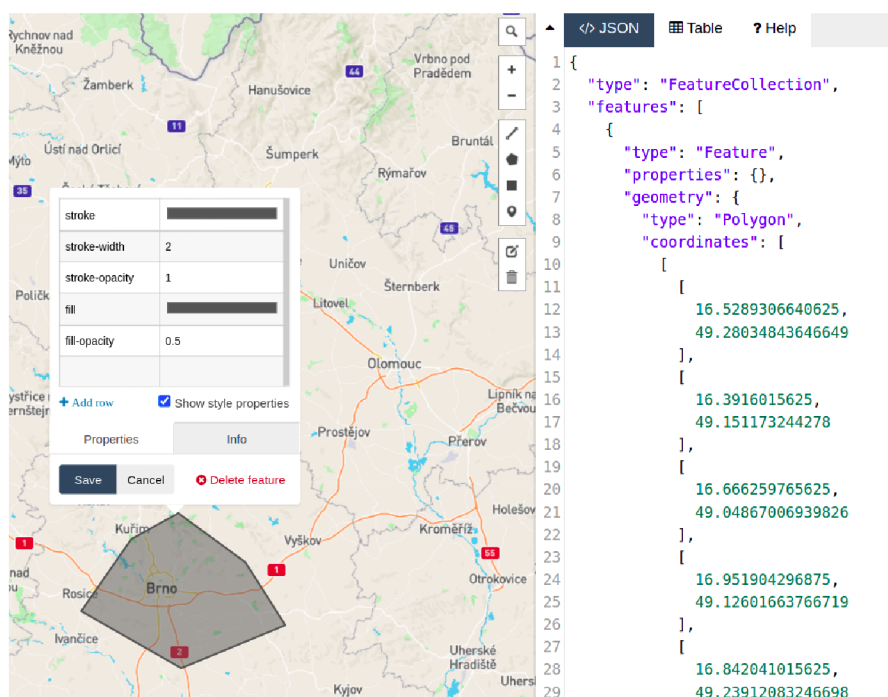
Obrázek 3.5: Uživatelské prostředí a bodová mapa vytvořena nástrojem Khartis.

¹³<https://www.sciencespo.fr/cartographie/khartis/>

geojson.io

Geojson.io¹⁴ je webová aplikace pro správu a vytváření geografických objektů ve formátu GeoJSON. Jeho hlavní využití spočívá v možnosti jednoduše transformovat objekty z jiných formátů jako je TopoJSON, GPX, CSV, KML a GTFS. Ty je následně možné upravovat pomocí uživatelského editačního rozhraní či pomocí přímého zásahu do GeoJSON souboru, jak je vidět na ukázce 3.6. Ten se živě generuje v postranním panelu zároveň se všemi provedenými úpravami.

Nástroj disponuje také možností přidávat vlastní definice pole „properties“ a definovat tak barvu, průhlednost či jiné vlastnosti vytvářeného objektu. Využití najde nástroj převážně při vytváření vlastních objektů nebo testovacích dat, která nejsou pevně spjatá s již známými geografickými objekty, jako například hranice administrativních celků.



Obrázek 3.6: Uživatelské prostředí aplikace Geojson.io s možností editovat přímo datovou reprezentaci vytvářených objektů.

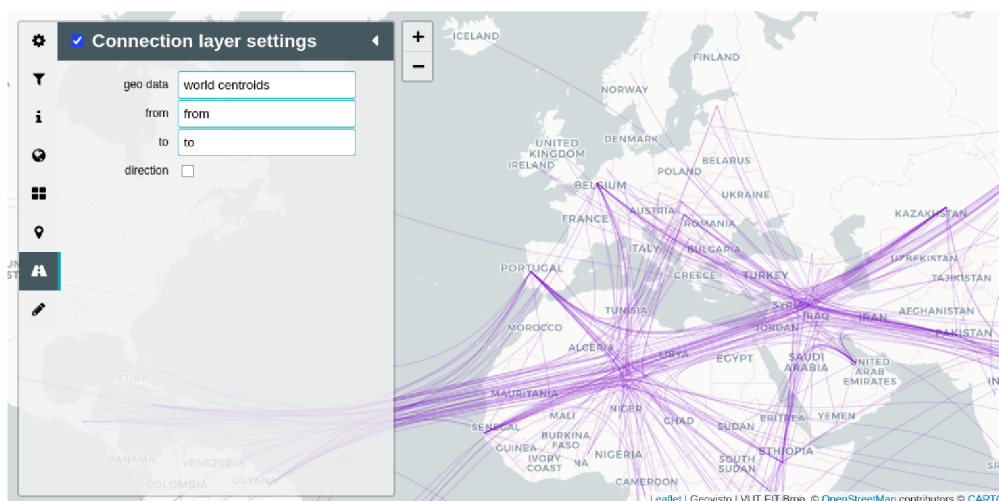
3.2.4 Geovisto

Jelikož se tato práce zabývá implementací podpory hierarchického zobrazení objektů v rámci Geovisto, je v této kapitole tento nástroj popsán pouze z pohledu uživatele a je nastíněna základní funkcionalita a jeho použití. Geovisto je open-source projekt, vytvořen na půdě Fakulty informačních technologií Vysokého učení technického v Brně a Institutu výpočetní techniky Masarykovy univerzity v Brně. Sami autoři si kladou za cíl zobecnit tvorbu geovizualizací pomocí nástroje, který je zaměřen na snadné použití jak profesionály, tak začínajícími uživateli [15]. Geovisto nabízí již hotové vrstvy tematických map, jako je kartogram, bodová mapa nebo mapa spojení, která je zobrazena v ukázce 3.7. Autor však není limitován použitím pouze jedné vrstvy v jeden čas, ale může je navzájem překrývat a tím

¹⁴<https://geojson.io/#map=2/20.0/0.0>

vytvořit mnohem rozsáhlejší výsledek. Součástí jsou také nástroje „ang. tools“, které jsou plně modulární a je pouze na uživateli, zda se je rozhodne přidat a využít, nebo zda se rozhodne pro implementaci vlastních.

Existují dva druhy dat, které musí uživatel do Geovista při vizualizaci nahrát. Prvním je soubor ve formátu GeoJSON, obsahující definice geografických objektů pro vizualizaci kartogramu popřípadě definici souřadnic bodů určující pozici centroidů v bodové mapě a mapě spojení. Druhým vstupem je datový soubor s generickými datovými hodnotami ve formátu JSON. Tento formát nebyl vybrán náhodou, autoři tak reagují na aktuální trendy v dnešní době, kdy se velké datové sady ukládají v hierarchických objektových formátech [15]. Obě datové sady vyžadují, aby obsahovaly atribut, jehož obsahem je jedinečný identifikátor umožňující propojení mezi daty a jejich prostorovou informací. Každá z vrstev pak umožňuje s datovými doménami pracovat samostatně a poskytuje také základní agregační funkce, jako suma a součet počtu výskytů. Hlavní předností Geovista se tak stává jeho modularita, zobrazení více druhů vizualizací na sobě a možnost pro vložení do vlastních webových stránek. Všechna nastavení se dají serializovat do konfiguračního souboru, který si může uživatel uložit pro pozdější použití, nebo pomocí něj nastavit a připojit jednotlivé moduly k jádru aplikace. Ty mezi sebou mohou komunikovat pomocí událostí, které zajišťují adekvátní reakci, například na změnu dat ve vstupu. Středobodem aplikace je poté samotný mapový podklad, který je implementován pomocí knihovny Leaflet popsané v podsekcí 3.2.2.



Obrázek 3.7: Uživatelské prostředí a mapa spojení vytvořená pomocí systému Geovista.

Kapitola 4

Administrativní celky a jejich hierarchické dělení

Většina obyvatelných částí světa je rozdělena na politicky odlišné územní celky, kterými mohou být země nebo nezávislá území. Pro účely vnitřní správy a řízení se často tyto jednotky rozdělují do menších oblastí nazývaných administrativní jednotky. Například USA jsou rozděleny na státy (Alabama, New York, Texas, ...) a tuto skutečnost nazýváme jako divizi první úrovně. Jednotlivé státy jsou dále děleny na další, menší správní celky, nazývané kraje. Tuto skutečnost opět nazveme divizí, ale tentokrát druhé úrovně. Stejným principem jsme schopni pojmenovat všechny jednotlivé vrstvy dělení územních částí. Převážně u menších států jako je Grenada či Vatikán se setkáme pouze s jednou úrovní rozdělení. Naopak u velmi rozlehlých států jako je Čína, Kanada nebo Ukrajina můžeme nalézt i tři nebo čtyři takové úrovně. Místní administrativní jednotky lze mimo jiné považovat za sociálně konstruované entity, které slouží pro prostorové scénáře a ekonomické procesy [11]. Ty se mohou projevit například jako rozlišení cen u poskytovatele veřejné dopravy na krajské úrovni nebo příslušností voliče k určitému volebnímu okrsku na úrovni města. Různorodost správních jednotek mezi zeměmi a vnitřními územími je velmi široká a nejasná. Každá země si velikost a způsob dělení určuje sama, přestože existuje několik standardů, které se snaží tuto skutečnost řešit, jak je zmíněno níže. Hranice území se obvykle vytváří podle několika různých veličin, které se dle [14] dají rozdělit do tří faktorů:

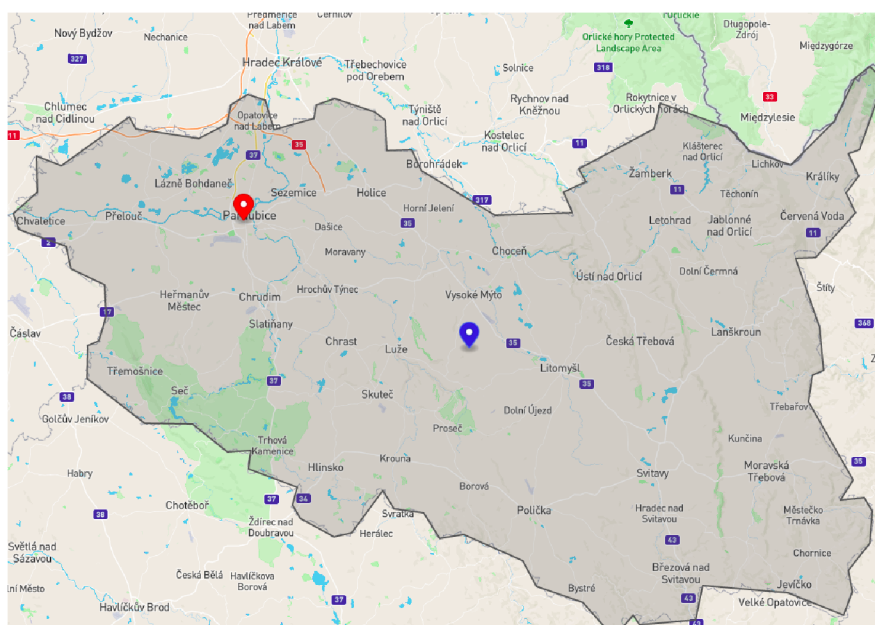
- velikost regionu,
- tvar (kompaktnost) regionu,
- poloha centra regionu.

Tvar, tedy kompaktnost regionu, může být vytvořen uměle pomocí rovných čar či oblouků. Za příklad použití takového přístupu můžeme považovat jednotlivé státy USA, kde se setkáme s hranicí v podobě přímky i několik stovek kilometrů dlouhé. Opačným přístupem je tvoření hranice správního celku na základě geografických vlastností území. Linie tak může mít i stovky různých zákrutů či klíčků, a to obzvlášť pokud se vine korytem řeky, na hranici oceánu nebo po hřebenu pohoří.

Ve středu takových celků se většinou nacházejí velká města, které plní správní úlohu nad přiděleným územím a tvoří přirozené centrum takového administrativního celku. Je zde ale potřeba rozlišovat pojmy geometrický střed území a významový střed území. Dle českého statistického úřadu [1] je geometrický střed statistického obvodu shodný se středem

jeho polygonu. Zároveň ale uvádí definiční vymezení takovýchto bodů jako bezpředmětné, jelikož na území měst se jedná o obvykle velmi malá území a ve venkovských lokalitách se naopak často shoduje s územím základní sídelní jednotky. Takový geometrický střed pak nijak neovlivňuje významový střed, popsán dále, v nadřazených územních prvcích a územně evidenčních jednotkách. Významový střed pak český statistický úřad definuje na svých webových stránkách jako střed centra osídlení na určité části území České republiky. Jeho vytvoření vychází z metodiky tvorby významových středů základních sídelních jednotek a z referenčního definičního bodu základního registru územní identifikace, adres a nemovitostí.

V geografických vizualizacích se setkáme s použitím obou přístupů, a to převážně na bodové mapě. Použití geometrického středu může být pro čtenáře mnohem více názornější, protože se bod nachází právě ve středu objektu, kde data chceme zobrazovat. Získání souřadnic lze provést také matematickým výpočtem středu plochy území a nemusí být definován příslušnými úřady jako významový střed, jehož umístění může být občas velmi matoucí. To zvláště pokud je zasazen poblíž hranic svého administrativního celku. Jako příklad uveďme Pardubický kraj v obrázku 4.1, jeho geometrický střed, označen modrým bodem, je umístěn na jihovýchod od svého významového středu vyznačeného červeným bodem.



Obrázek 4.1: Ukázka rozdílu mezi významovým a geometrickým středem.¹

4.1 Dělení administrativních celků

Administrativní celky si lze představit jako hierarchicky uspořádaný strom objektů s různými úrovněmi. V minulosti tak došlo k několika pokusům formalizovat jednotlivé úrovně a o pojmenování takových objektů. V této práci se budu zabývat pouze dvěma formalizacemi, které budou nadále využity i v implementaci nástroje.

¹Vytvořeno pomocí nástroje Geojson.io. <https://geojson.io/>

4.1.1 ISO 3166

Dle [27] je ISO 3166 mezinárodní standard pro kódování názvů zemí a jejich subdivizí. Jeho součástí jsou tři různé díly. ISO 3166-1, poprvé publikována již v roce 1974, definuje kódy v dvoupísmenné, trojpísmenné a trojčiferné formě. A to nejen pro státy, ale i jejich závislá území společně s názvy v anglickém a francouzském jazyce. Na tuto normu pak navazuje řada dalších norem. Dalším dílem a zároveň navazující normou je ISO 3166-2. Ta definuje již i nižší vrstvu kódů, specifickou zvláště pro každý stát. Kód **CZ** tak označuje českou republiku a **CZ-72** již podružný Zlínský kraj. Tato norma, se svými dvoupísmennými definicemi státu, se například používá pro pojmenování nejvyšších úrovní internetových domén tak, jak jsou definovány společnostmi Internet Assigned Numbers Authority (IANA) a Internet Corporation for Assigned Names and Numbers (ICANN). Posledním dílem je ISO 3166-3. Tato norma určuje čtyřmístné kódy pro země, u kterých došlo ke spojení ze dvou a více států, rozdělení nebo změně v oficiálním názvu. Všechny díly jsou neustále aktualizovány agenturou pro údržbu této normy.²

Jak je popsáno v [27], jednotlivé díly jsou sice pojmová schémata sama o sobě, ale jejich pojmy na sebe i odkazují. Jednotlivé subdivize definované v ISO 3166-2 tak mohou být propojeny mezi sebou do hierarchické struktury. Například Francie je rozdělena do 100 různých celků, které jsou seskupeny do dvaadvaceti metropolitních a čtyř zámořských regionů. Definováno je i nesourodé seskupování, jako u Kanady, která se skládá z deseti provincií a dalších třech teritorií.

4.1.2 NUTS

Klasifikace NUTS, nebo také česky nomenklatura územních statistických jednotek, byla vytvořena již v první polovině sedmdesátých let minulého století statistickým úřadem Evropské unie. Jedná se o dělení členských států evropské unie do několika úrovní. Původně zamýšleným účelem bylo vytvořit jednotný systém pro rozdělení území EU pro potřeby produkce regionálních statistik. Postupem času bylo vydáno několik různých aktualizací, k dnešnímu dni se jedná o verzi NUTS 2021. Ta už neslouží pouze ke sběru statistik, ale své využití nachází i na poli geografických vizualizací či k zaměření regionálního rozvoje.

Základními kategoriemi dělení jsou tři úrovně klasifikace. V nejvyšší úrovni, označované jako NUTS1, nalezneme dle oficiálních webových stránek³ hlavní socio-ekonomické regiony. V úrovni NUTS2 se nachází menší správní celky pro aplikaci regionálních politik. A v nejnižší úrovni NUTS3 pak nejmenší geoprostorové celky. Jako příklad uvedu rozdělení České republiky. V nejvyšší úrovni NUTS1 je definován pouze jeden celek a to Česká republika. V druhé úrovni pak celky jako jihovýchod, střední morava nebo severozápad. Nejnižší úroveň NUTS3 pak odpovídá jednotlivým krajům České republiky.

Jelikož se jedná o dělení vytvořené pro statistickou analýzu, setkáme se na jednotlivých úrovních s různě velkými, zdánlivě nesourodými objekty. Jejich vytvoření totiž podléhá třem různým principům a charakteristikám. Jako první princip jsou uvedeny prahové hodnoty populační hustoty pro jednotlivé úrovně dělení. Můžeme se tak setkat s velice malými jednotkami v hustě obydlených oblastech, které jsou zařazeny do stejné úrovně s velkými jednotkami ve venkovských oblastech. S tím je svázán druhý princip, který umožňuje redefinici tohoto dělení. K němu ale smí dojít až po minimálně třech letech od poslední redefinice. Jako třetí princip je uvedeno, že dělení NUTS favorizuje administrativní dělení jednotlivých

²<https://www.iso.org/iso-3166-country-codes.html>

³<https://ec.europa.eu/eurostat/web/nuts/background>

členských států. A to z praktických důvodů jako je dostupnost statistických dat. Eurostat taktéž definuje lokální administrativní úroveň, zkráceně LAU. Ta je kompatibilní s vyšší úrovní NUTS3 a k změně definice dochází každý rok. Celky většinou reprezentují města, vesnice nebo jejich menší seskupení.

4.2 API poskytující definice objektů

V rámci této práce jsem již popsal jak datovou reprezentaci geoprostorových objektů, tak i jejich administrativní dělení. V této podkapitole se odráží tato dvě témata jako důležitá součást existujících API pro dotazování a stahování datových reprezentací geografických objektů. Existuje mnoho dalších volně dostupných, ale i placených API poskytujících definice geografických objektů. V následujících řádcích popíši pouze ty, které souvisí nejvíce s touto prací. Existují ale i jiné, jako je Geodata API od Bing Maps nebo MapBox.

4.2.1 Overpass API

Overpass API je aplikační programové rozhraní spojené s projektem OpenStreetMap, který zajišťuje volně dostupná geografická data pro širokou veřejnost. Dle [3] je OpenStreetMap geografická databáze spravována uživateli internetu. Obsahuje obrovské množství dat, které lze zobrazit na mapě, a její databáze je aktualizována každých patnáct minut. K dnešnímu dni v ní nalezneme přes pět miliard GPS bodů a má více než 2,5 milionu uživatelů, kteří do projektu přispívají.

Všechna tato data jsou veřejně dostupná, nebo se na ně lze dotazovat právě pomocí Overpass API. Ta je určena, na rozdíl od hlavní OSM API, pouze pro čtení datových informací. Dotazování je pak možné přímo pomocí jazyka XML nebo specializovaného OverpassQL. Vzhledem k otevřenosti celého projektu vzniklo již několik „*wrapperů*“ abstrahujících celý proces dotazování, ať už v programovacím jazyku Python, nebo jako npm balíček pro prostředí Node.js. Dotazovací jazyk je velmi expresivní, dá se tedy jeho pomocí odeslat požadavek na velmi specifické informace. A to převážně díky možnosti zavedení prostorového omezení v rámci dotazu, pomocí definování vztahů mezi uzly, přidáním filtrů či ohraničujících rámečků. Služba posílá odpověď ve formátu OSM XML, což je nadstavba OpenStreetMap projektu nad značkovacím jazykem XML, sestávající převážně ze značek pro uzly (`nodes`), cesty (`ways`) a vztahy (`relations`).

Existuje několik přístupových bodů, na kterých běží instance Overpass API. Tyto body jsou obvykle udržovány třetí stranou a u některých je potřeba počítat i s omezením, jako je počet přístupů denně či dotazů za minutu. Z tohoto důvodu je umožněno nainstalovat a zprovoznit si osobní přístupový bod API, například pro použití v aplikaci, na vlastním webovém serveru. Služba není schopná dopředu určit velikost výsledku dotazu a z tohoto důvodu neumožňuje ani poskytnutí odhadovaného času do doby dokončení požadavku. API je velmi dobře zdokumentována a to na oficiálních stránkách projektu⁴, kde lze díky velké komunitě uživatelů vyzkoušet jednotlivé dotazy i na interaktivní mapě .

4.2.2 GeoNames

Podle [19] je GeoNames velmi známá geoprostorová datová sada poskytující geografická data a metadata přibližně 7 milionů unikátních míst z celého světa, shromážděna z několika dalších zdrojů. Na nejvyšší úrovni jsou místa kategorizována do 9 tříd jako například města,

⁴https://wiki.openstreetmap.org/wiki/Overpass_API

vodní plochy, cesty či přírodní prvky. Dále se dělí do dalších 663 kategorií podle přesnějšího určení typu místa. Existuje i speciální, takzvaná nultá třída, která obsahuje nezařazené entity. Mimo geografickou polohu, uchovanou v souřadnicovém systému WGS84, poskytuje GeoNames i další informace o místech. A to názvy v různých jazycích, údaje o nadmořské výšce nebo velikost populace.

API je používána především k nalezení míst pomocí full-textového vyhledávání, poštovních směrovacích čísel pro většinu států nebo k vyhledání nejbližších míst od zadaného `geonameID`, jedinečného identifikátoru přiděleného všem objektům v databázi. Samozřejmostí je i reverzní vyhledávání nebo vyhledávání pomocí odeslání požadavku se souřadnicemi. Tuto API ale zmiňuji především kvůli možnosti stažení hierarchie objektů. K tomuto slouží požadavek pomocí `geonameID` identifikující objekt na jehož hierarchii se dotazujeme. Odpovědí je nám XML nebo JSON soubor objektů seřazený od nejvyššího po nejnižší objekt. V následujícím obrázku je vizualizován příklad dotazu na město Curych.

Na oficiálních stránkách nalezneme i dlouhý výčet prémiových uživatelů a projektů, které GeoNames využívají. Nevýhodou API je nutnost autentizace pomocí uživatelského účtu, jelikož veřejný demonstrační účet je limitován na denní počet požadavků.

Kapitola 5

Analýza

V této části se zaměřím na analýzu požadavků uživatelů pro přívětivé zobrazení dat v tematických mapách při různých úrovních přiblížení. Základním předpokladem je umožnění přiblížení či oddálení mapy uživatelům, kteří chtějí zobrazovanou problematiku vidět z několika různých pohledů v závislosti na hierarchickém uspořádání objektů. Takový objekt je tedy možno v závislosti na úrovni pohledu určitým způsobem rozdělit na menší či spojit na větší celky. Tyto objekty reagují změnou zobrazovaných hodnot z datové sady, a to využitím různých agregačních funkcí, jako je suma nebo počet prvků.

5.1 Uživatelé

- **Autor** vizualizace může být kdokoli s potřebou vizualizovat data v tematické mapě. Ať už učitel, který chce vytvořit interaktivní pomůcku do výuky nebo zaměstnanec firmy, který chce získat širší pohled na zobrazovanou problematiku. Autor ale může disponovat datovou sadou, pro jejíž vyjádření je nutné znát kontext na různých úrovních. Jako příklad uvedu počty nakažených obyvatel určitou infekční nemocí. Při pohledu na malé administrativní celky, jako jsou třeba města, může autor předat velmi přesné informace o počtu nakažených v dané oblasti a s tím i dedukovat následné zatížení tamního zdravotnického zařízení. Při pohledu na data z perspektivy větších administrativních celků, jako jsou kraje či celé státy, pak může předat informace o celkové nákaze v populaci, kde se momentálně nejvíce šíří, nebo jakou souvislost má pohyb obyvatel mezi kraji na vývoj situace. U využití nástroje je tedy nutné předpokládat, že autor nemá široké dovednosti na úrovni pokročilého programování a je nutné i pro něj vytvořit dostatečně intuitivní způsob vytváření hierarchické struktury objektů, které chce zobrazit.
- **Čtenář** vizualizace je jakákoliv osoba, která se setká s geovizualizací na internetových stránkách či v nějaké aplikaci. Čtenář by měl být plně odstíněn od jakékoliv možnosti editace hierarchické struktury objektů, či celkově zobrazení s využitím tohoto nástroje vypnout. A to především, kvůli výrazné změně informací, které se budou zobrazovat, a pravděpodobnému pozměnění úmyslů autora. Naopak by mělo být čtenáři umožněno svobodně přibližovat a oddalovat vizualizaci s produkcí vždy stejných výsledků pro každou úroveň. U určitých typů vizualizací, jako je například bodová mapa, je zapotřebí myslet na nejednoznačnost zobrazovaného území. Vizualizace by tak měla jasně reflektovat přechod mezi jednotlivými úrovněmi geografických objektů tak, aby čtenář jasně pochopil, že došlo ke změně zobrazovaných dat jako reakce na základě

jeho akce. To může být zajištěno určitou animací, změnou barvy či jiným vizuálním indikátorem.

5.2 Vstupy

Nástroj bude pro svou plnou funkcionalitu potřebovat tři různé druhy vstupů.

- **Geografické objekty**

Prvním vstupem jsou definice geografických objektů v jejich datové reprezentaci. Různé úrovně objektů mohou být definovány ve vícero souborech. Nástroj by tak měl umožnit vstup z několika souborů nebo umožnit nahrání objektů více vrstev v takovém formátu, který dokáže v jednom souboru definovat i vícero různých polygonů či bodů a zároveň je později mezi sebou odlišit. Jednotlivé objekty by také měly být jasně identifikovatelné, ať už pomocí jména nebo unikátního identifikátoru.

- **Datová sada**

Datová sada by měla být upravena pro použití s hierarchií a to převážně z důvodu agregace jednotlivých datových domén. Jelikož se ale jedná o generická data, může disponovat hodnotami pro každou jednotlivou úroveň hierarchie, a nástroj by měl tento případ reflektovat zamezením agregace. Naopak v případě chybějících hodnot pro vyšší úrovně by měl být schopen nalézt nejnižšího potomka nebo potomky určitého objektu, vyextrahovat všechny hodnoty potomků a ty vhodně agregovat do jednoho vyššího celku.

- **Definice hierarchie**

Definice hierarchie musí být dostatečně jednoduchá na to, aby se dala efektivně měnit a vytvářet, zároveň ale musí být dostatečně flexibilní. A to zejména z důvodu poskytnutí co největší kontroly uživatele nad výslednou vizualizací. Vstup by měl být proveden buď pomocí interaktivního uživatelského prostředí, ve kterém je jasně viditelné s jakými potomky nebo rodičem je objekt svázán a při které úrovni přiblížení dojde ke změně pohledu, nebo pomocí textového či jiného datově orientovaného formátu souboru, jako je JSON nebo XML. Definování vztahů by mělo probíhat za použití jednoznačných identifikátorů geografických objektů, které si může uživatel sám zvolit, a to především kvůli vlastní přehlednosti. Výsledná definice musí dodržet všechna pravidla a omezení, která vyplývají z této problematiky a budou dále popsány v kapitole návrh6.

5.2.1 Chybějící vstupy a jejich stažení

Pokud uživatel nebude disponovat geografickými objekty či definicí hierarchie, nástroj by měl odkázat na důvěryhodné zdroje, nebo rovnou umožnit stažení a následné použití takových vstupů. U geografických objektů se jedná především o administrativní celky jednotlivých zemí, ze kterých by si měl uživatel mít možnost vybrat, jak hluboké zanoření a počet úrovní objektů potřebuje. Jako ideální způsob rozlišení jednotlivých úrovní se jeví využití některé z norem a definic pro dělení administrativních celků, které jsou popsány v kapitole 4. Nejenže je pravděpodobnější obeznámení uživatele s danou definicí, zároveň ale většina poskytovatelů geografických objektů sama takové dělení využívá. Jednotlivé tematické mapy pak mohou využívat různé typy geografických objektů. Nástroj by tak měl umožnit i výběr mezi jednotlivými typy, jako jsou polygony či body, a poskytnout pouze vybraný druh.

5.3 Současná řešení

V momentálním čase neexistuje mnoho řešení, které by umožnily uživateli vytvořit geografickou vizualizaci pomocí autorského systému s podporou hierarchického zobrazení objektů. Většinou je nutné, aby si uživatel naprogramoval vlastní řešení, což je v případě vytváření jedné jednoduché vizualizace poměrně hodně časově náročná činnost, která předpokládá, že uživatel má programovací znalosti v daném oboru a také, že autorský systém vůbec umožňuje modifikovat svůj zdrojový kód. V kapitole 3.2.2 jsem popsal některé programovací knihovny, které se pokoušejí tuto problematiku řešit, a to především u bodových objektů pomocí takzvaného shlukování (clustering). Jedná se spojení více bodů do jednoho celku na základě vzdálenosti těchto míst. Využití nalezne především u vizualizací hustě rozmístěných objektů, jako jsou například obchody v městských metropolích. Pokud by každý obchod byl označen určitou značkou, ze vzdáleného pohledu by uživatel neviděl nic jiného než změt neurčitých značek. Při využití shlukování by však viděl značku pouze jednu, ideálně i s číslem udávajícím počet obchodů spojených do jednoho celku. Tento přístup je ale většinou realizován pomocí výpočtu velikosti značky na obrazovce uživatele a vzdálenostech mezi nimi. Autor tak nemá žádnou kontrolu ani nad úrovní přiblížení, při kterém k shlukování dojde, ani nad pozicí výsledné značky. Může se objevit na naprosto nevhodném místě a zkreslit tím výsledek. Programovací knihovny většinou umožní programátorovi přechíst aktuální úroveň přiblížení a zároveň na její změnu vhodně reagovat.

5.4 Proces použití

Proces použití se dá rozdělit do dvou případů, a to proces použití čtenáře a autora vizualizace. U čtenáře se jedná pouze o akci přiblížení či oddálení zobrazované problematiky, na jejímž základě dojde k reakci ve formě změny zobrazovaných objektů. Žádná další přímá interakce se z jeho strany s nástrojem neděje. U autora je ale proces použití již o něco složitější, a proto jej popíši v následujícím seznamu.

1. **Shromáždění geografických objektů** – je zapotřebí, aby uživatel znal již na začátku práce geografické objekty, se kterými chce dále pracovat. Pokud jimi nedisponuje, může si je sám vytvořit, a to například nástrojem `geojson.io` popsaným v kapitole 3.2.3 nebo pomocí nabídnutého rozhraní stáhnout z jiného zdroje.
2. **Definice hierarchické struktury** – určení propojení mezi množinou zobrazovaných objektů pomocí jednoduchých nebo mnohonásobných vazeb, jako je rodič-potomek nebo rodič-potomci. A to za dodržení všech pravidel a omezení, které jsou nutné pro validní definici.
3. **Určení úrovní přiblížení** – na základě zvoleného typu geografické vizualizace určit, ve kterých úrovních přiblížení dojde ke změně zobrazovaných objektů a s tím spojeným přesunem výše či níže v hierarchické struktuře. Uživatel by měl využít buď předem připravených hodnot, nebo alespoň znát minimum a maximum rozsahu, ze kterého hodnoty vybere.
4. **Propojení hierarchie s objekty** – uživatel propojí definici hierarchie s datovou reprezentací objektů. K tomu využije interaktivní uživatelské prostředí, nebo při vytváření uvede jednoznačné identifikátory v obou vstupech a k propojení dojde automaticky v rámci nástroje.

5. **Vložení datové sady** – vstup datové sady je záměrně až posledním krokem procesu použití. Teprve na základě aktuálně zobrazovaných objektů se dají případně hodnoty agregovat či spočítat informace o minimální, respektive maximální zobrazované hodnotě.

5.5 Shrnutí

Z analýzy vyplývá, že problematika geografických vizualizací s použitím hierarchicky řazených geoprostorových objektů má velmi specifické využití a existuje několik různých přístupů, jak tuto problematiku řešit. Rozdělením uživatelů na dva typy, tedy autora respektive čtenáře, získáváme povědomí o dvojitém využití nástroje. A to jak pro vytvoření požadovaného výsledku, tak i pro následné zobrazení vizualizace, při kterém je nutné měnit zobrazené celky dynamicky na základě čtenářova pohybu po mapě.

Díky analýze vstupů si můžeme povšimnout, že nástroj bude muset zkombinovat hned tři různé sady dat dohromady pro úspěšnou funkčnost. Formát geografické a datové sady bude pravděpodobně specifikován vhodnou notací, u definice hierarchie objektů však bude nutné vytvořit vlastní formát zápisu tak, aby co nejvíce vyhovoval uživateli, ale i potřebám nástroje. Autor vizualizace by taktéž měl mít možnost pro stažení některých chybějících vstupů, které je možno získat z veřejně dostupných webových zdrojů, a k tomuto účelu poskytnout i vhodné rozhraní.

Na trhu neexistuje mnoho nástrojů, které nabízí dostatečně komplexní řešení pro zobrazení objektů v hierarchii, jak vychází ze sekce s analýzou současných řešeních 5.3. Pokud ano, musí autor výrazně zasáhnout do zdrojových kódů, nebo nemá dostatečnou kontrolu nad výsledkem a možností využít vlastní geografické objekty. Proces použití poté určuje jednotlivé kroky, které musí uživatel splnit, aby získal požadovaný výsledek, a naznačuje myšlenkové pochody, které by mělo použití nástroje odrážet. Závěry z analýzy poslouží pro navržení nástroje, který bude součástí autorského systému Geovisto a pokusí se co největší měrou vyřešit problémy, které z analýzy vyplývají.

Kapitola 6

Návrh

Výsledkem práce bude implementace funkcionality pro hierarchické zobrazení objektů a rozhraní pro stažení geografických objektů v systému Geovisto. Jelikož si tento systém zakládá na své modularitě a možnosti nástroje libovolně přidávat či odstraňovat, bude se i můj návrh a následné naprogramování nástroje držet tohoto přístupu. Předpokládá se, že autor vizualizace provede stažení datové reprezentace geografických objektů a definování hierarchie pouze ojedinele, a to v době jejího vytváření.

Vizualizace pravděpodobně bude poté putovat k širokému spektru čtenářů, kteří tuto funkcionalitu již nevyužijí a autor tak nemusí nástroj dále využívat. Stále ale bude nutná správa objektů, reagování na přiblížení, respektive oddálení mapy čtenářem, nebo agregace hodnot z datové sady. Z tohoto důvodu nebude výsledkem pouze jeden nástroj, ale nástroje dva. A to nástroj pro správu hierarchie, načtení konfigurace a všeho dalšího, co je spojené se samotnou geografickou vizualizací. Druhý nástroj bude sloužit pro stažení datové reprezentace geografických objektů, případně pro vytvoření hierarchie. Oba budou implementovány v jazyce TypeScript, ve kterém je naprogramován i zbytek knihovny Geovisto.

6.1 Definice hierarchie geografických objektů

Jak již bylo zmíněno v popisu Geovista a analýzy, jedním ze vstupů je definice hierarchie geografických objektů. Do systému bude definice vstupovat jako součást konfiguračního souboru, který je zapsán ve formátu JSON a pomocí kterého se k jádru připojují a nastavují i ostatní moduly či nástroje. Tento přístup zajistí znovupoužitelnost, jednoduchou editaci a navíc se bude předpokládat že uživatel definici nahrává jen v případě, kdy nástroj hierarchie chce opravdu využít. Alternativou může být definice přímo v GeoJSON souboru s geografickými objekty, a to s využitím atributu `properties`, jak je popsáno v sekci 2.2.1. Ty ale obsahují poměrně dlouhé pole se souřadnicemi, a to především pokud jsou v souboru velmi složité polygony. Orientace v takovém souboru by byla velmi složitá a časově náročná bez použití specializovaných programů. Hierarchie bude definována jako pole objektů, kde každý objekt bude obsahovat tři různé atributy.

1. **id** – jednoznačný identifikátor geografického objektu ve formátu textového řetězce. Pomocí něj bude možné určit, který polygon, bod či jiný objekt je přiřazen k tomuto nastavení.
2. **parent** – jednoznačný identifikátor rodiče, opět v textovém formátu. Určuje objekt, který je v hierarchii nadřazený tomuto.

3. **zoomLevel** – číselná hodnota určující úroveň přiblížení, ve kterém se stane objekt aktivním a zobrazí se ve vizualizaci.

V celém poli se nesmí vyskytnout více objektů se stejným ID, jelikož by definice nebyla jednoznačná. Díky tomuto formátu také zajistím, že každý objekt má maximálně jednoho rodiče a datové hodnoty se nemohou štěpit do více nadřazených objektů. Pokud není ID rodiče přítomné, jedná se o objekt v nejvyšší úrovni. Každá definice musí obsahovat minimálně jeden takový, jinak by se jednalo o uzavřenou posloupnost objektů a ty by dohromady tvořily kružnici. Nebylo by tak možné určit, který objekt je rodičem a který potomkem, tudíž by nevznikla žádná hierarchická vazba.

Protože Geovisto umí a pracuje s více geografickými doménami současně, může tedy obsahovat definice několika vstupů geografických objektů v jeden moment, je nutné toto vše ještě zaobalit do dalšího pole, které bude obsahovat jednotlivé definice pro různé domény. Každá doména pak bude identifikována svým jménem a bude obsahovat i atribut logické proměnné **aggregation**, které umožní uživateli rozhodnout, zda chce datové hodnoty agregovat z nejnižší úrovně objektů nebo zda poskytl v datové sadě hodnoty i pro vyšší úrovně.

V ukázce 6.1 je uveden příklad definice dvou velmi jednoduchých hierarchií. Doména s názvem *Hierarchy_czech* je validní, má povolenou agregaci a skládá se ze dvou objektů, kde objekt s identifikátorem *KRAJ_ZLK* je potomkem objektu *CZ_REP*, který je v nejvyšší úrovni hierarchie a do kterého by se hodnoty jeho případných potomků agregovaly. Druhá doména s názvem *Hierarchy_UK* validní **není**, jelikož neobsahuje definici pro nejvyšší objekt v hierarchii.

```
1   "HierarchyConfig": [{
2     "name" : "Hierarchy_czech",
3     "aggregation" : true,
4     "hierarchy": [{
5       "id": "KRAJ_ZLK",
6       "parent": "CR_REP",
7       "zoomLevel": 10
8     }, {
9       "id": "CZ_REP",
10      "parent": "",
11      "zoomLevel": 3
12    }]
13  }, {
14    "name" : "Hierarchy_UK",
15    "aggregation" : false,
16    "hierarchy": [{
17      "id": "Ireland",
18      "parent": "UK",
19      "zoomLevel": 8
20    }, {
21      "id": "Dublin",
22      "parent": "Ireland",
23      "zoomLevel": 18
24    }]}]
```

Výpis 6.1: Ukázka navrhnuté JSON reprezentace hierarchie.

6.1.1 Úroveň přiblížení

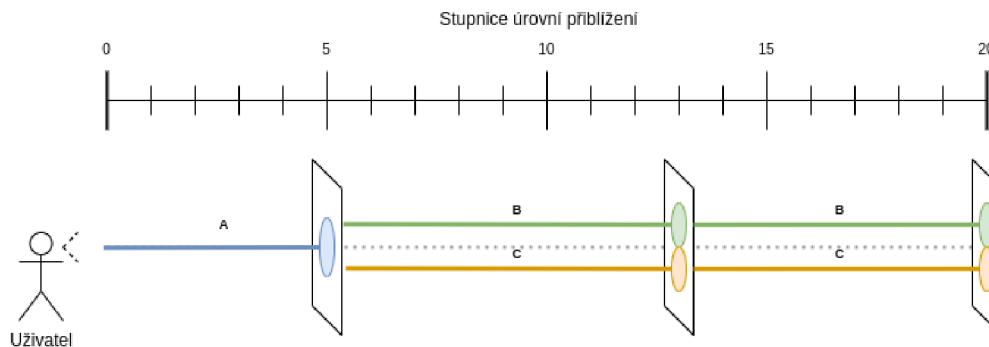
Jak je popsáno výše, pro určení úrovně přiblížení bylo zvoleno číslo, ve kterém se stane objekt aktivním a zobrazí se ve vizualizaci. Tento přístup nebyl zvolen náhodně, ale vychází z implementačních detailů jádra Geovista. Jelikož jeho použití a význam nemusí být na první pohled jasný, je mu věnována tato subsekce.

Mapový podklad je v aplikaci Geovisto implementován pomocí knihovny Leaflet.js. V ní přiblížení neprobíhá posunem po určitém spojitém spektru, jak by uživatel mohl na první pohled očekávat, ale po stupnici diskretních hodnot. Uživatel se tedy vždy nachází v určité úrovni přiblížení, která je definovaná jeho číslem. Jako nultou úroveň označuje Leaflet vždy pozici největšího oddálení mapy. Programátor má poté možnost definovat číslo označující největší přiblížení, v Geovistu je hodnota implicitně nastavená na číslo dvacet. Toto číslo se většinou odvíjí od použité podkladové mapy, a stejné stupnice se tak používají i v jiných knihovnách. Ukázka jednotlivých úrovní a jejich čísel je naznačena v příložené grafice 6.1.



Obrázek 6.1: Ukázka úrovní přiblížení a jejich korespondujících číselných označení za použití knihovny leaflet.js

V definici hierarchie tak hodnota atributu `zoomLevel` označuje moment, při kterém dojde k zobrazení objektu a tedy překrytí jeho potomků. Pokud má objekt sám přiřazeného rodiče, je z vizualizace skryt v momentě, kdy uživatel přejde do úrovně stejného nebo nižšího čísla úrovně. U objektu v nejvyšší vrstvě definované hierarchie, tedy takového, který nemá žádného rodiče, `zoomLevel` označuje úroveň, od níž se po oddálení již vizualizace nijak nemění. Naopak význam tohoto atributu je u nejnižších objektů, tedy takových, které nefigurují jako rodiče jiného a v teorii grafů bychom je označili pojmem listy, bezvýznamný. Při překročení hranice totiž již není žádný jiný potomek, který by se mohl zobrazit, a proto tento objekt zůstane ve vizualizaci aktivním. Demonstrační příklad popisuje infografika 6.2, která zobrazuje, na kterých úrovních přiblížení budou viditelné geografické objekty *A*, *B* a *C*. Definice hierarchie korespondující s příkladem je viditelná v ukázce kódu 6.2.



Obrázek 6.2: Graf popisující právě aktivní zobrazované objekty podle úrovně přiblížení a jejich definované hierarchie. Naznačuje souvislost atributu `zoomLevel` s určením úrovně, při které dojde k rozštěpení nebo spojení zobrazovaných objektů.

```

1 {
2   "id": "A",
3   "parent": "",
4   "zoomLevel": 5
5 },{
6   "id": "B",
7   "parent": "A",
8   "zoomLevel": 13
9 },{
10  "id": "C",
11  "parent": "A",
12  "zoomLevel": 13
13 }

```

Výpis 6.2: Definice hierarchie odpovídající demonstračnímu příkladu 6.2

6.2 Nástroj hierarchie

Nástroj bude fungovat jako modul připojitelný k jádru aplikace Geovisto. Uživatel nástroje získá možnost dynamicky měnit zobrazované geografické objekty v jednotlivých mapových vrstvách na základě přiblížení či oddálení mapového podkladu čtenářem. Geografické objekty jsou navíc díky nástroji mezi sebou propojeny určitými vztahy, které poskytnou možnost upravovat datové hodnoty zobrazovaných objektů na základě hodnot ostatních objektů v hierarchii, a to pomocí agregačních funkcí, jež jsou definovány mapovými vrstvami. Uživatel je také schopen přesně určit jednotlivé úrovně, při kterých dojde k přeměně objektů. Nástroj poskytne rozhraní ostatním modulům jádra nástroje Geovisto pro zjištění těchto úrovní nebo právě aktivních objektů a umožní jim tak pracovat s těmito informacemi pro vlastní použití.

Nástroj také nebude mít žádné uživatelské prostředí pro vstup. Jeho nastavení a chování bude ovlivnitelné pouze pomocí vstupního textového souboru a na jeho výstup navazují pouze ostatní mapové vrstvy, které se starají o samotnou vizualizaci. Těmto vrstvám nástroj také umožní získat nejnižší objekty v hierarchii, a to v případě, že je povolena agregace dat a vrstva musí sčítat hodnoty do vyšších celků.

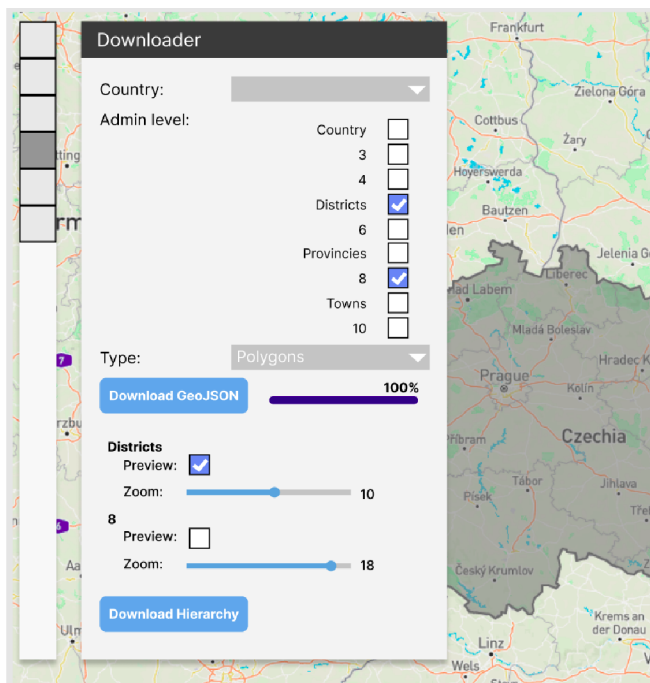
6.3 Nástroj pro stažení geografických objektů

Nástroj umožní uživateli vybrat určitou zemi, pro kterou následně stáhne geografické objekty z API přístupového bodu. Uživatel bude mít na výběr z jednotlivých druhů geografických objektů, jako jsou polygony či body, a stejně tak bude mít na výběr i z různých úrovní administrativních celků, jak jsou popsány v kapitole 4. Objekty budou po stažení z API nehladě na svou administrativní úroveň spojeny do jednoho souboru a uživatel si je bude moci stáhnout jako jeden soubor formátu GeoJSON. Objekty taktéž budou obsahovat všechny náležitosti pro následné použití při vizualizacích nástroje Geovisto a to především povinný jednoznačný atribut ID. Součástí nástroje bude i automatické generování definice hierarchie. K tomu bude vytvořeno uživatelské rozhraní, ve kterém bude možné definovat úrovně přiblížení jednotlivých administrativních úrovní a také tyto úrovně zobrazit. Vygenerovanou hierarchii bude opět možné stáhnout, a to v takovém formátu, který bude kompatibilní s nástrojem hierarchie popsáným v předešlé sekci.

Návrh UI

Navržené UI na obrázku 6.3 reflektuje prostředí systému Geovisto, který poskytuje možnost všem nástrojům vytvořit i prvek vlastního uživatelského rozhraní, a to přidáním záložky do vestavěného panelu nástrojů na levém okraji mapy. Návrh tak pracuje pouze s plochou, kterou mu aplikace poskytuje a jednotlivé ovládací prvky se budou řadit pod sebe.

Jedním z dalších nástrojů, které jsou pro systém Geovisto dostupné, je i modul pro změnu stylu. Ten změní barvy a design všech uživatelských prvků, například pro použití s nočním módem zobrazení. Pro kompatibilitu je proto potřeba při implementaci využít předdefinovaných ovládacích prvků a ne klasických HTML značek pro uživatelský vstup.



Obrázek 6.3: Návrh uživatelského rozhraní pro stahování objektů.¹

¹Vytvořeno pomocí programu Figma.<https://www.figma.com/>

Kapitola 7

Implementace

V této kapitole popíší přístup k implementaci nástroje pro hierarchii *Hierarchy Tool* a nástroje pro stahování geografických objektů pomocí API *geoDownloader*. Jelikož je aplikace Geovisto implementována v jazyce TypeScript, budou i oba tyto nástroje naprogramovány ve stejném jazyce jako moduly, které je možné k aplikaci připojit.

Pro první zmíněný se bude jednat především o jeho architekturu, komunikaci se správcem geografických dat, použité algoritmy a řešení problému s udržení kontextu hierarchie. Dále také o modifikaci mapové vrstvy kartogramu, bodové mapy a mapy spojení pro využití pouze aktivních objektů a následnou agregaci jejich hodnot.

U nástroje pro stahování geografických objektů popíší převážně využití technologie, úskalí pro přístup k Overpass API a generování definice hierarchie.

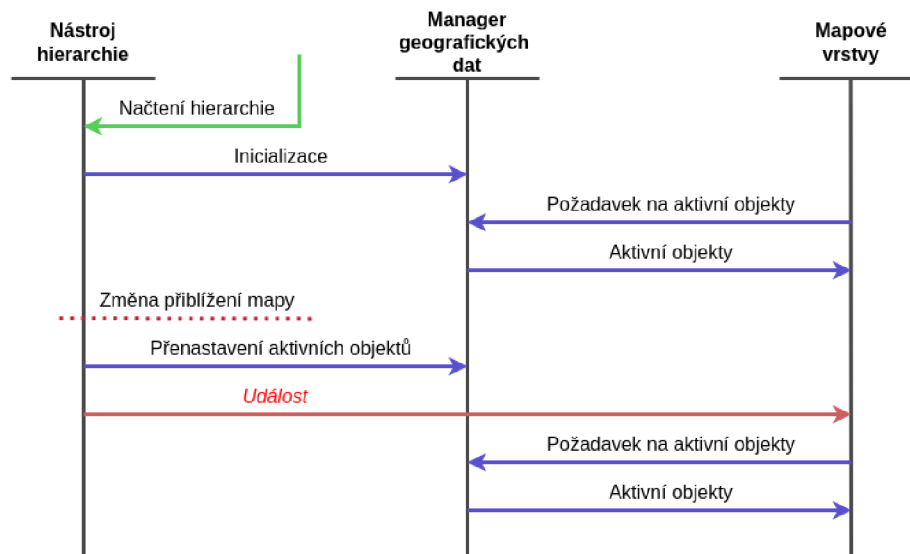
7.1 Nástroj hierarchie

Vzhledem k použití jazyka TypeScript bylo při implementaci možné využít základních principů objektově orientovaného programování. Nástroj je tak navržen jako třída dědicí vlastnosti od nadřazené generické třídy `MapTool`, definující základní rozhraní pro všechny nástroje jádra.

Jelikož nástroj komunikuje s dalšími mapovými vrstvami, které jsou ale také implementovány formou modulů, bylo nutné nalézt jakéhosi prostředníka, který je již součástí jádra aplikace, a přes který by mohl nástroj s jednotlivými vrstvami komunikovat. K tomuto účelu bylo nejvhodnější využít objekt `GeoDataManager` (dále správce geografických dat), který udržuje informace o přítomných doménách geografických objektů, a který využívají vrstvy tematických map k poskytnutí datových reprezentací polygonů či bodů. Celý nástroj pracuje pouze s jednoznačnými identifikátory geografických objektů. Jejich datovou reprezentaci nadále spravuje a poskytuje pouze správce geografických dat.

V následujících odstavcích jsou popsány jednotlivé kroky, které nástroj při svém použití absolvuje, a to v takovém pořadí, v jakém jdou za sebou. Posloupnost je zobrazena i v obrázku 7.1, který dává jednotlivé kroky a události do souvislosti se všemi použitými objekty.

¹Vytvořeno pomocí nástroje draw.io. <https://app.diagrams.net/>



Obrázek 7.1: Graf posloupnosti jednotlivých procesů při použití hierarchie. Po načtení hierarchie z konfiguračního souboru dochází k inicializaci správce geografických dat, který dále předává aktivní geografické objekty mapovým vrstvám. Nástroj hierarchie pokračuje sledováním událost přiblížení mapy. V reakci na ní posílá událost mapovým vrstvám, které reagují novým požadavkem na aktivní geografické objekty.¹

Načtení definice hierarchie

Geovisto je možné nastavit pomocí konfiguračního souboru, který obsahuje jak vlastní nastavení, tak i nastavení pro jednotlivé nástroje. Do tohoto konfiguračního souboru je možné umístit i definici hierarchie tak, jak byla definována v kapitole 6.1. Při vytváření nástroje je tedy tento soubor předán a následně zpracován.

Jako první dojde ke kontrole všech náležitostí. Tedy, zda je přítomen alespoň jeden objekt na nejvyšší úrovni, nebo zda je definován i rodič každého objektu. Chyba v definici by totiž mohla vést k mnoho nepříjemnostem ve výsledné vizualizaci a to zejména k nesmyslnému překrývání objektů nebo zacyklení celého programu.

Uvnitř nástroje je umístěna další struktura, která slouží jako správce jednotlivých domén. Jelikož každá tematická vrstva může pracovat s jinou doménou geografických objektů, musí i nástroj pracovat s několika definicemi hierarchie zároveň. K jednotlivým definicím je také uložen jejich agregační příznak.

Rozšíření správce geografických dat

Jak bylo zmíněno v úvodu této sekce, k implementaci bylo nutné upravit správce geografických dat. Došlo tedy k jeho rozšíření o implementaci rozhraní pro práci s hierarchií. Nástroj hierarchie inicializuje správce geografických dat touto definicí, načež je uložena do vnitřního objektu `HierarchyTree`. Toto řešení se může zdát na první pohled velmi neefektivní, jelikož by stačilo vždy předat pouze informaci o aktivních geografických objektech. Pokud je ale povolen agregační příznak, mapová vrstva potřebuje ke své funkci i informace o nejnižších potomcích aktivního geografického objektu, ze kterých následně agreguje hodnoty pro právě aktivní objekty.

Při využití nástroje hierarchie tak metoda správce `.getFeatures()`, kterou využívají mapové vrstvy pro získání geografických objektů v GeoJSON formátu, přistoupí nejdříve k objektu `HierarchyTree` a získá identifikátory aktivních objektů. Na jejich základě profiluje všechny geografické prvky a navrácí pouze ty momentálně aktivní.

Objekty `HierarchyTree` také dokáží vrátit identifikátory nejnižších potomků hierarchie. K tomu je využit upravený iterativní algoritmus prohledávání do hloubky.

Sledování změny přiblížení

Nástroj dále přejde do pasivního režimu, kdy již aktivně nic nevykonává a pouze sleduje změnu přiblížení mapy. Tuto událost lze sledovat pomocí specializovaných posluchačů (*angl. Event listeners*).

Knihovna Leaflet poskytuje základní metody pro práci s úrovněmi přiblížení a uživatelskou informací. Nástroj připojí vlastní *listener* na hlavní objekt mapy i s metodou nástroje `.zoomChanged`, pro zpracování události. Zde je nutné rozlišovat druhy událostí, které mapa odesílá. Pro implementaci je použita reakce na událost *zoom*, která se spustí při jakékoli změně přiblížení i například při použití „*fly over*“ animace. Existují ale i události, jako například *zoomend* nebo *zoomstart*, které se spouštějí pouze při ukončení nebo počátku změny přiblížení. Použitím těchto událostí by ale mohlo docházet k četným chybám, jelikož jiné moduly mohou přibližovat či oddalovat mapu o několik úrovní naráz. Nástroj by tak nedokázal udržovat momentálně aktivní objekty a další informace, které je nutné aktualizovat při přechodu mezi každou úrovní.

Aktuální úroveň přiblížení je získána metodou `.getZoom()` volanou nad hlavním objektem `Map`. Toto číslo je následně předáno správci geografických dat, který případně přenastaví své aktivní objekty ve stromu. Změna úrovně přiblížení však nutně neznamená, že došlo ke změně aktivních objektů v hierarchii. Z tohoto důvodu byla implementována struktura v nástroji, která po změně přiblížení zkontroluje, zda se od poslední změny změnila některá z hierarchie, tedy zda se změnil seznam aktivních objektů dané hierarchie. V případě, že došlo ke změně aktivních objektů, odesílá nástroj událost typu `GeoDataChangeEvent`, která notifikuje mapové vrstvy o změně aktivních objektů, a spouští tak jejich reakci. Pokud ke změně aktivních objektů nedošlo, událost se neodesílá. Toto je čistě z výkonnostního hlediska, jelikož mapové vrstvy reagují na událost přepočítáním dat, což může být poměrně náročná operace.

7.1.1 Podpora mapových vrstev

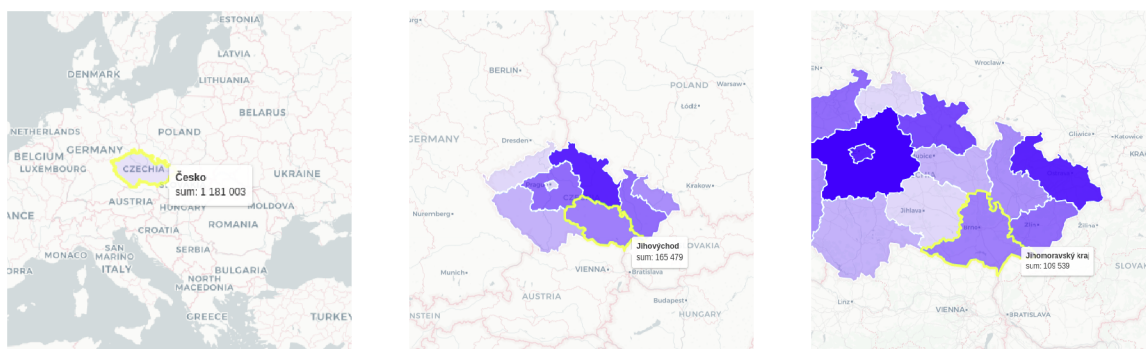
Součástí řešení byla nutnost modifikovat jednotlivé mapové vrstvy, které jsou již přítomné v jádře Geovista tak, aby správně využívaly nástroje hierarchie. Každá z vrstev je schopná pomocí metody `.handleEvent()` reagovat na různé zprávy zaslané objektu. V případě tohoto nástroje se jedná o správnou reakci na událost typu `GeoDataChangeEvent`. Reakce vrstev je tak vždy spuštěná právě touto zprávou, následné procesy se už liší vrstva od vrstvy podle jejího typu a jsou popsány v podkapitolách níže.

Kartogram

První modifikovanou vrstvou je vrstva kartogramu. V reakci na zprávu nejdříve zkontroluje, zda je hierarchie vůbec povolena. Pokud ano, tak zda je povolena pro uživatelem vybranou doménu. Na základě těchto informací získá nové aktivní geografické objekty pomocí přetížené metody geografického správce.

Kartogram následně projde celou datovou sadu a jednotlivé hodnoty sčítá pomocí zadaných agregačních funkcí jako je suma nebo počet prvků. Hodnoty jsou sčítány do takzvaných agregačních věder, objektů, které slouží pro uchování těchto hodnot a následnému předání k vizualizaci.

Pokud není povolen agregační příznak pro danou doménu, aktualizace dat tímto končí. Kartogram pouze promítne do získaných geografických prvků jejich hodnoty a vše zobrazí. V případě, že je povolen agregační příznak domény, kartogram se dotáže správce geografických dat na ID nejnižších potomků aktivních objektů. Následně proběhne druhé kolo agregace hodnot, při kterém se vytvoří nová agregační vědra pro aktivní objekty a do nich jsou agregovány hodnoty věder právě jeho nejnižších potomků. Není tak nutné procházet znovu celou datovou sadu a stačí pouze spojovat již agregované hodnoty. Výslednou vizualizaci v různých úrovních přiblížení si lze prohlédnout v obrázku 7.2.



Obrázek 7.2: Ukázka mapy kartogramu v různých úrovních přiblížení s použitím nástroje hierarchie. Lze si povšimnout i agregace hodnot pro jednotlivé administrativní úrovně.

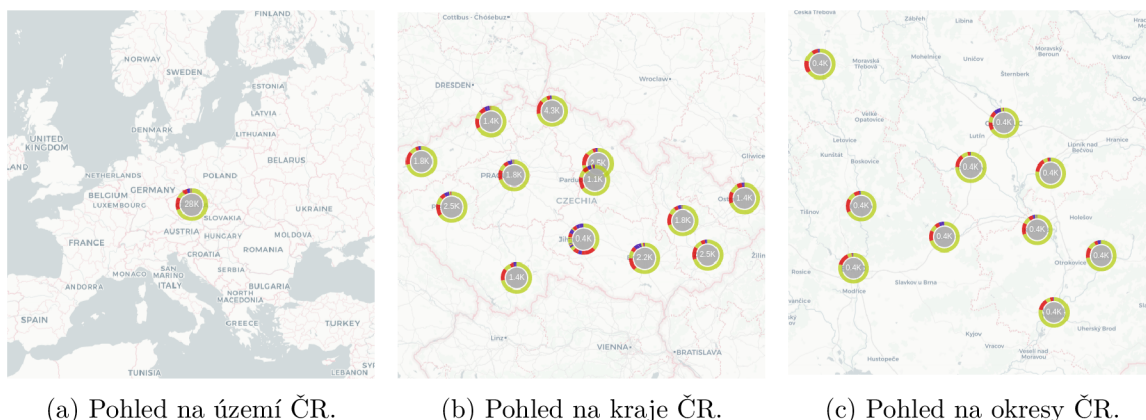
Bodová mapa

Bodová mapa využívá ke svému zobrazení geografické objekty typu bod. Neagreguje pouze jednu hodnotu, jako v případě kartogramu, ale několik hodnot podle jejich kategorie. Ty následně zobrazí formou *značek (markers)*, v tomto případě s podobou koláčového grafu zobrazující jednotlivé kategorie. Implementace tady opět využívá agregačních věder, v případě povoleného příznaku agregace tak sčítá hodnoty i přes jednotlivé kategorie dat. Ukázku výsledné vizualizace si lze prohlédnout na obrázku 7.3.

Mapová vrstva taktéž využívá objektu `MarkerClusterGroup` z balíku `markercluster`². Ten původně při velkém oddálení zajišťoval spojení markerů do větší skupiny. Výsledný cluster se umístil do středu vypočteného ze souřadnic jeho bodů, což nemusí být vždy žádoucí. A to zvláště když uživatel chce využít různých druhů středů popsaných v kapitole 4, nebo vlastních geografických bodů. Z tohoto důvodu je tento balík při použití nástroje hierarchie zakázán a nepoužívá se.

Při implementaci bylo vyvinuto úsilí přetížít jednotlivé metody tohoto balíku a dosáhnout tak použití s tímto nástrojem a to kvůli animaci spojení respektive rozdělení skupin. Balík ale pracuje na zcela jiném principu než mnou navržený nástroj, a proto bylo od jeho dalšího použití upuštěno.

²<https://github.com/Leaflet/Leaflet.markercluster>



Obrázek 7.3: Ukázka bodové mapy v různých úrovních přiblížení s použitím nástroje hierarchie.

Mapa spojení

Poslední mapovou vrstvou, která byla v rámci této práce upravena, je mapa spojení. Ta pracuje na principu zobrazení dvou různých bodů, mezi kterými existuje určitý druh vazby, jako například dvě města, mezi kterými vede letecká linka.

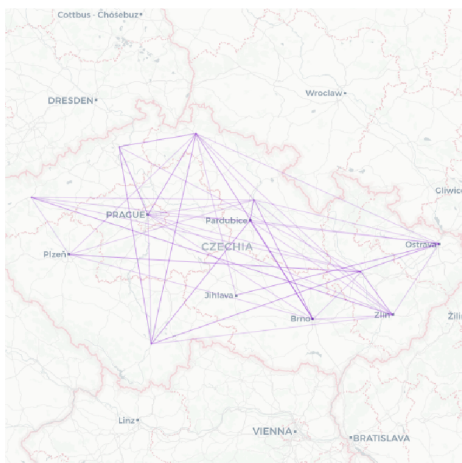
Při pozitivním agregačním příznaku nástroj musí projít všechna spojení znovu a kontrolovat, zda existuje spojení mezi alespoň dvěma potomky právě aktivních objektů. Pokud ale existovalo spojení mezi potomky jednoho právě aktivního objektu, je tato informace zahozena a autor s touto skutečností musí při použití nástroje počítat. Tímto je myšleno například spojení mezi dvěma okresy stejného kraje, které není v případě agregace do vyššího celku kraje nijak zohledněno.

Mapová vrstva taktéž využívá knihovny D3.js popsané v kapitole 3.2.2, respektive algoritmu z její D3-force knihovny³. Ten se při vytvoření vizualizace pokouší jednotlivé linky zahrnout do větších svazků. Pokud tedy existuje mnoho spojení vedoucích blízko sebe, je z nich na pohled vytvořena jedna silnější linka. Po změně aktivních objektů se tak tento algoritmus musí vždy zavolat znovu, aby se dosáhlo podobných výsledků vizualizace i v jiných vrstvách hierarchie. Ukázkové použití mapy spojení s hierarchií si lze prohlédnout na obrázku 7.4.

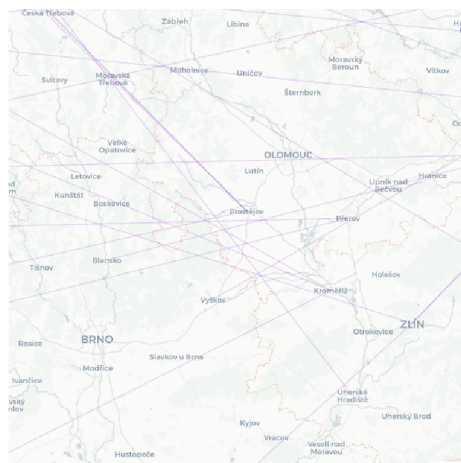
7.2 Nástroj pro stažení objektů

Druhým nástrojem, který byl implementován, je `geoDownloader` poskytující rozhraní pro stažení datové reprezentace geografických objektů pomocí API a také nástroj pro automatické vygenerování definice hierarchie pro tyto objekty. Nástroj je implementován jako modul pro aplikaci `Geovisto`, ze které využívá převážně rozhraní pro uživatelské prostředí nástroje. Práce částečně navazuje na bakalářskou práci [25], kde mimo nástroje pro editaci geografických objektů byla implementována i možnost stažení polygonů pro určitý stát. Výsledné uživatelské rozhraní, které uživatel používá při stahování objektů, si lze prohlédnout na obrázku 7.5.

³<https://github.com/d3/d3-force>



(a) Spojení mezi kraji ČR.



(b) Spojení mezi okresy ČR.

Obrázek 7.4: Ukázka geografické vizualizace na mapě spojení. Na prvním obrázku lze vidět spojení vytvořená pouze mezi krajskými městy, agregovaná z dat spojení mezi okresy. Na druhém, po přiblížení vizualizace, originální data spojení mezi okresy.

7.2.1 OverpassAPI

Ke stahování se využívá Overpass API, která byla popsána v kapitole 4.2.1. Umožňuje stahování různých geografických dat z databáze projektu OpenStreetMap. K dotazování existuje několik přístupových bodů, které se liší převážně výkonem a rychlostí odpovědi. Dotazovací jazyk je však pro všechny instance stejný. Já jsem ve své implementaci zvolil přístupový bod VK Maps Overpass API⁴, který je možné použít zdarma bez dalších větších omezení. V dotazovacím jazyce je možno specifikovat různé atributy. V rámci implementace jsou využity tyto tři:

1. **boundary** – atribut nastavený na hodnotu **administrative**. Určuje vyhledání pouze administrativních celků.
2. **admin_level** – číselná hodnota označující tzv. *administrativní úroveň*. Nejvyšší úroveň, tedy číslo dva, určuje hranice vybraného státu. Vyšší čísla, až po úroveň deset, označují nižší administrativní celky státu.
3. **country** – dvoumístný kód země podle normy ISO3166-1, která byla popsána v sekci 4.1.1. Určuje zemi pro kterou budou data stažena. V nástroji tedy nelze stahovat data pro vyšší územní celky, jako je Evropa nebo jiné kontinenty.

Ani OverpassAPI problematiku různých definic administrativních úrovní neřeší a nevyužívá žádné přesné normy na jejich dělení, jak byly popsány v kapitole 4. Z tohoto důvodu se může stát, že API na dotaz s určitou úrovní odešle zpět pouze prázdnou odpověď. Například při dotazu s úrovní šest získáme pro Českou republiku její kraje, ale pro Holandsko se nevrátí nic, jelikož tato úroveň není pro tento stát definována. Na oficiálních stránkách ale lze nalézt orientační tabulku⁵, která označuje definované úrovně pro většinu zemí. Tu

⁴<https://maps.mail.ru/osm/tools/overpass/api/interpreter>

⁵<https://wiki.openstreetmap.org/wiki/Tag:boundary%3Dadministrative>

jsem převedl z HTML reprezentace do formátu JSON pomocí nástroje pro konverzi⁶ a je dále využita v nástroji.

Jak je z použitých atributů zřejmé, pokud chce uživatel stáhnout několik administrativních úrovní, musí se požadavek na přístupový bod odeslat několikrát, pro každou úroveň zvlášť. Jednotlivé přístupové body API však různě omezují opakovaný přístup, například limitací dvou požadavků za minutu, nebo velikostí vrácených dat. Zvolený přístupový bod žádná omezení momentálně nemá, nelze však poslat několik požadavků v jeden moment, a proto se každý další odešle až po přijetí odpovědi z toho předešlého.

7.2.2 Použití nástroje

- **Výběr země**

Uživatel nejdříve vybere z nabídky zemí tu, pro kterou požaduje stažení geografických objektů. Seznam dostupných zemí je uložen ve statickém souboru. Ten byl již zpracován a použit v práci [25] a není tedy nutné využití jiného řešení.

- **Výběr administrativních úrovní**

Nástroj vytváří vícenásobnou selekci administrativních úrovní a to pro všechny, které poskytuje OverpassAPI, tedy od úrovně dva po úroveň deset. Pokud existuje záznam v tabulce, popisky jednotlivých úrovní jsou zobrazeny se jmény, které se z ní získají. I v tomto případě ale zůstanou přítomné i ostatní hodnoty úrovní, jelikož tabulka je importována ze statického souboru a mohlo tak dojít k její změně.

- **Stažení objektů**

Nástroj následně odesílá asynchronní požadavky na přístupový bod API. Doba mezi odesláním a příchodem odpovědi je poměrně dlouhá, z tohoto důvodu využívá nástroj ukazatele průběhu pro indikaci průběhu stahování. Po dobu stahování také není možná žádná jiná práce s tímto nástrojem, což je naznačeno „zašednutím“ tlačítek.

- **Konverze, filtrace a simplifikace**

Odpovědi API jsou data ve formátu OSM, prvním krokem po jejich stažení je jejich konverze do formátu GeoJSON. K tomu je využit npm balíček⁷, poskytující tuto funkcionalitu. Následně probíhá filtrace objektů, kde se na základě uživatelského výběru ponechají pouze objekty typu polygon nebo bod. V případě, že uživatel požaduje polygony, nabídne mu nástroj také možnost jejich simplifikace.

Stažené polygony jsou totiž velmi detailní a jejich následná vizualizace v této kvalitě je velmi náročná na zdroje uživatele. Navíc soubor s definicí několika takových objektů, dosahuje poměrně velké velikosti, což nemusí být pro uživatele žádoucí. Proto je využit nástroj `turf`⁸, který poskytuje metodu `.simplify()`. Tato metoda umožňuje použít i koeficientu tolerance, který určuje, k jak velkému zjednodušení dojde. Uživatel má proto pomocí posuvníku možnost tuto hodnotu také zvolit.

- **Generování hierarchie**

OverpassAPI nenabízí žádnou možnost pro stažení informací o hierarchickém uspořádání administrativních celků. Nelze proto jasně určit vztahy mezi jednotlivými úrovněmi stažených objektů. Lze předpokládat, že střed polygonu nižší úrovně leží ve

⁶<https://www.convertjson.com/html-table-to-json.html>

⁷<https://www.npmjs.com/package/osmtojson>

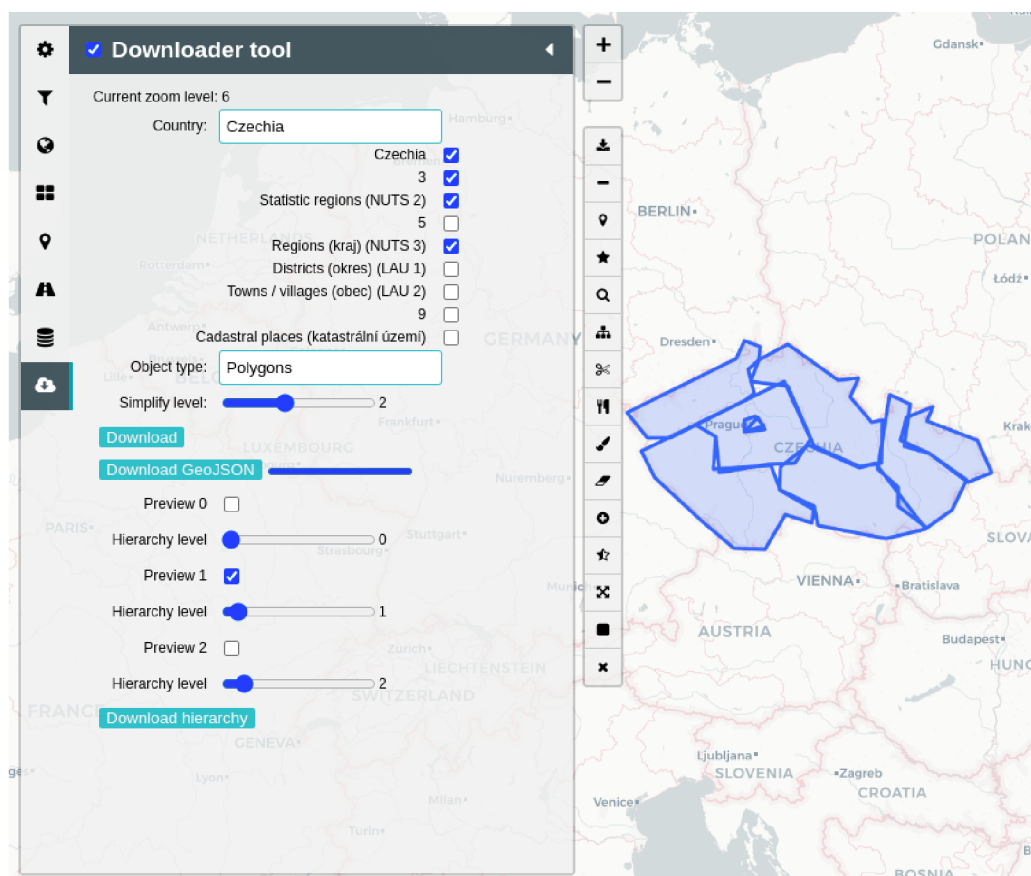
⁸<http://turfjs.org/>

vnitřní ploše polygonu vyšší úrovně, a tedy se jedná o jeho potomka. Na tomto předpokladu staví automatické generování definice hierarchie, kterou může následně uživatel v případě stažení polygonů použít. Geografické objekty bodů ale žádnou plochu nemají, při jejich stažení proto není možné hierarchii generovat.

Následně uživatel určí pomocí číselných posuvníků úroveň přiblížení (atribut `zoomLevel`) jednotlivých úrovní. Pro jeho uživatelský komfort je také přidána i možnost zobrazit každou úroveň zvlášť na mapovém podkladu, aby získal představu o velikosti a množství stažených polygonů. K tomu byl přidán i ukazatel momentálního přiblížení, uživatel tak nemusí tuto hodnotu odhadovat. Výslednou definici hierarchie je možné stáhnout ve formátu JSON a následně pouze přidat do konfiguračního souboru nástroje hierarchie.

- **Spojení objektů a jejich stažení**

Na závěr jsou jednotlivé objekty ze všech stažených administrativních úrovní spojeny dohromady do jednoho souboru formátu GeoJSON. K tomu je využito nástrojů knihovny Leaflet⁹ pro práci s nimi.



Obrázek 7.5: Ukázka vytvořeného uživatelského prostředí pro stahování geografických objektů. Obrázek zachycuje stav již po stažení, kde je možné i upravit a generovat definici hierarchie.

⁹<https://leafletjs.com/reference.html#geojson>

Kapitola 8

Testování

V této kapitole je popsáno testování implementovaných nástrojů. Je opět rozdělena do dvou různých sekcí podle testovaného nástroje.

8.1 Nástroj hierarchie

V první části testování tohoto nástroje byly prováděny zátěžové testy, ve kterých bylo cílem zjistit hranice implementovaného řešení a jeho případné chyby. Ve druhé pak uživatelské testování, ve kterém bylo cílem zjistit vliv a působení na čtenáře vizualizace při použití mapové vrstvy s podporou hierarchie.

8.1.1 Testování mapových vrstev

V první části jsem se zaměřil na otestování nástroje s různě velkými datovými vstupy. A to na všech třech mapových vrstvách, které byly v rámci práce upraveny, aby podporovaly tento nástroj.

K tomuto byly využity datové sady o různé velikosti, jež byly generovány pomocí jednoduchého skriptu v jazyce Python. Geografické prvky byly získány pomocí druhého implementovaného nástroje umožňujícího jejich stažení, a to pro území České republiky na všech čtyřech administrativních úrovních. Jednalo se tak přesně o 100 geografických objektů.

Při provádění testu jsem se zaměřil především na rychlost shlukování či rozpadnutí jednotlivých geografických prvků při oddálení či přiblížení mapového podkladu. K tomu byly použity výpisy do konzole jednotlivých fází vizualizace i s jejich časovými známkami, které indikovaly přibližnou dobu trvání jednotlivých operací. Postupně byly vkládány větší a větší datové sady, kde každá byla testována s povoleným agregačním příznakem. Jelikož se jedná o aplikaci, která běží ve webovém prohlížeči klienta, mohou se výsledky jednotlivých testů lišit podle typu a výkonnosti jejich zařízení. Výsledné časové hodnoty si lze prohlédnout v tabulce 8.1.

Velikost datové sady	10 000 prvků	100 000 prvků	1 000 000 prvků
Karotgram	6ms	49ms	320ms
Bodová mapa	8ms	58ms	574ms
Mapa spojení	60ms	246ms	1,617s

Tabulka 8.1: Tabulka obsahující časové prodlevy mezi rozpadnutím resp. sloučením geografických objektů při různých velikostech datové sady.

Mapa kartogramu

Výsledná vizualizace nevykazovala žádné vážné chyby. Mezi jednotlivými úrovněmi docházelo k správné přeměně objektů podle definice hierarchie. Přesto při větších datových sadách docházelo k nepatrnému zamrznutí celé vizualizace. Po bližším prozkoumání bylo možné přesně určit, že se jedná o agregaci hodnot v první iteraci, která je popsána v podsekcí 7.1.1 implementace pro danou mapu. Při testování kartogramu tak nebylo zjištěno žádných velkých nedostatků.

Bodová mapa

Testování bodové mapy probíhalo opět pomocí náhodných vstupních dat pro Českou republiku. Výsledná vizualizace nevykazovala žádné vážné chyby, co se týče zobrazovaných dat. Prodleva mezi přechody v jednotlivých vrstvách hierarchie byla již vyšší než u předešlé mapy Kartogramu. Za příčinu tohoto prodlení jsem určil vytváření jednotlivých koláčových grafů, které každý bod zobrazuje. Jelikož ale byla použita náhodně generovaná data, existovalo velké množství kategorií, podle kterých se grafy vytvářely, a náročnost tak byla značně vyšší než při použití běžných dat.

Mapa spojení

Při generování dat spojení byly náhodně vybírány geografické objekty a mezi nimi tato spojení navázaná. Prodlevy mezi jednotlivými úrovněmi přiblížení byly největší ze všech popsaných vrstev. Vizualizace vykazovala známky pomalejší reakce, a to převážně kvůli náročnému přepočítávání jednotlivých spojení pomocí d3-force algoritmu, jak byl popsán v kapitole 7.1.1. Ten svoji činnost přepočítávání ukončí až po několika sekundách od zobrazení, a uživatel tak musí čekat než dojde k ustálení celé vizualizace.

Shrnutí

Všechny mapové vrstvy se chovaly dle předpokládaného očekávání. Nebyla pozorována žádná závažná vizuální chyba, která by zapříčinila špatnou interpretaci zobrazovaných dat uživatelem.

Se zvyšující se velikostí datové sady dochází k delší prodlevě u přechodů mezi jednotlivými úrovněmi hierarchie. Tento jev se nejvíce projevuje při použití mapy spojení. Ta při velkých datových sadách vykazuje již delší zamrznutí mapové vrstvy a může negativně ovlivnit dojem čtenáře na celou vizualizaci.

Jako jedno z možných řešení se jeví ukládání dat do mezipaměti. Při každém přechodu mezi úrovněmi totiž dochází k opětovné agregaci datových hodnot. Tímto způsobem by tak došlo k agregaci pouze jednou, a to při zobrazení celé vizualizace. Tento přístup ale předpokládá, že se nastavení jednotlivých mapových vrstev po zobrazení již nemění.

8.1.2 Uživatelské testování

Uživatelské testování bylo koncipováno formou pozorování a následného rozhovoru se třemi uživateli. Ti neměli žádnou předešlou zkušenost s vytvářením geografických vizualizací a nejsou profesně zaměřeni na informační technologie. Běžně se ale pohybují po internetu a

navštěvují různé zpravodajské servery, díky čemuž se již mnohokrát s geografickou vizualizací setkali. K testování byly využity reálné datové sady o pandemii nemoci Covid-19, získané z oficiálních stránek ministerstva zdravotnictví České republiky¹.

Uživatelé byli vždy postaveni před hotovou vizualizaci kartogramu nebo bodové mapy na nejvyšší úrovni České republiky a bylo jim pouze řečeno, že se jedná o vizualizaci počtu nakažených. Dále byli požádáni, aby zjistili počet nakažených obyvatel v libovolném okrese.

Z pozorování a následných rozhovorů vyšlo najevo několik faktů. V případě použití kartogramu uživatelé instinktivně mapu přiblížili až po zobrazení jednotlivých okresů a nepřemýšleli nad tím, jak nalézt data pouze pro okres. V případě použití bodové mapy se naopak pokoušeli nejdříve klikat na marker a teprve poté se pokusili o přiblížení mapy. A to z důvodu neexistujícího výrazného indikačního prvku naznačujícího rozpad či shlukování jednotlivých markerů, kromě změny hodnoty ve středu koláčového grafu. Řešením může být použití kombinace mapových vrstev kartogramu a bodové mapy.

Uživatelé pak oceňují především menší vizuální chaos v jednotlivých vrstvách přiblížení. Hlavně při výrazném oddálení mapového podkladu nedochází k splývání jednotlivých objektů, jelikož na každé úrovni je jejich velikost přiměřeně velká pro pozorovatele.

8.2 Nástroj pro stažení objektů

V druhé části byl testován nástroj pro stahování geografických dat, a to se zaměřením na automatické generování hierarchie a použitelnost uživatelského prostředí.

Generování hierarchie

Test probíhal vždy vybráním náhodné země, stažením polygonů a následným nastavením a vygenerováním definice hierarchie. Vše pak bylo vizualizováno pomocí mapové vrstvy kartogramu. Při testování jsem narazil na dva různé problémy. API totiž poskytuje některé geografické objekty, které by měly správně být ve společné administrativní vrstvě, rozděleny do dvou vrstev. A to například v případě, kdy je daná země ve válce, a tudíž není jasné, ke kterému státu určité části území patří, nebo když se jedná o uměle vytvořený správní celek pro různé vojenské oblasti. V takovém případě automatické generování hierarchie nenalezlo objekt ve vyšší vrstvě a ponechalo ho bez určení jeho rodiče. Proto bylo opraveno prohledávání vyšších vrstev a pokud dané území nenalezne vhodného rodiče v první vyšší vrstvě, pokusí se o jeho nalezení i v ostatních vyšších vrstvách.

Druhým problémem byly administrativní celky, jejichž střed byl mimo jejich území nebo je v jejich středu otvor. Jako příklad můžeme uvést Středočeský kraj, který má ve svém středu další administrativní oblast Prahy. Z tohoto důvodu byla přidána kontrola, zda vypočtený střed je stále uvnitř plochy původního polygonu.

Uživatelské testování

Tři různí uživatelé byli požádáni, aby pomocí nástroje získali definice geografických objektů. Následně s nimi byl veden krátký rozhovor o použití nástroje. Z těchto rozhovorů vzešel podnět na lepší zobrazení právě probíhajícího stahování. Původně byl implementován pouze indikátor průběhu, který zobrazoval počet již stažených administrativních vrstev. Ten se ale začal naplňovat až po ukončení čekání na první staženou úroveň. Uživatel tak nepoznal, zda stahování opravdu začalo probíhat. Jeden z nich dokonce stlačil tlačítko pro stažení

¹<https://onemocneni-aktualne.mzcr.cz/covid-19>

několikrát. Na základě tohoto podnětu byla přidána změna kurzoru do stavu čekání po dobu stahování.

Nástroj byl taktéž několikrát využit pro stažení dat a generování hierarchie. A to pro využití při testování jednotlivých vrstev. Při jeho používání jsem nezjistil žádné vážnější chyby.

Kapitola 9

Závěr

Cílem této práce bylo vytvořit nástroj, který umožní využít hierarchického uspořádání geografických objektů a který dokáže tyto objekty poskytnout pro určitý stát na jeho různých administrativních úrovních. Výsledkem jsou nakonec nástroje dva, jeden pro hierarchii a druhý pro stahování geografických objektů.

První zmíněný poskytuje uživateli prostředí pro uspořádání geografických dat do vlastní hierarchie. Autor vizualizace tak může použít nástroj v kombinaci s ostatními geografickými vizualizacemi a docílit tak lepšího pochopení problému čtenářem, měnit zobrazované objekty na základě jeho úrovně přiblížení, nebo agregovat vstupní data do větších celků.

Druhý nástroj pak poskytuje uživateli možnost stažení datové reprezentace různých geografických objektů přes Overpass API, které následně využije pro vytváření svých vizualizací. Nástroj poskytuje také možnost určit i několik různých administrativních úrovní, zajistit zjednodušení stažených polygonů, ale i automatické generování definice hierarchie, kterou autor může využít v prvním zmíněném nástroji.

Nástroje byly implementovány s cílem poskytnout autorovi geografických vizualizací nové možnosti pro jejich vytváření. Použitím je možné zvýšit míru předaných informací nebo doplnit vizualizace o nové prvky na základě uživatelských potřeb. Využití tak může nalézt v mnoha odvětvích, kde je potřeba vytvářet geografické vizualizace. Ať už například pro vizualizaci počtu osob nakažených určitou nemocí, tak i pro výsledky voleb v jednotlivých volebních okrscích, okresech či krajích, nebo pro hustotu zalidnění různých oblastí. Oba nástroje pak budou publikovány jako plnohodnotné moduly pro aplikaci Geovisto, a to ve formě *npm* balíčku.

Nástroj pro podporu hierarchie může být rozšířen o další vylepšení, jako je ukládání předpočítaných dat do mezipaměti, které by optimalizovalo rychlost vizualizací. Nástroj je také jednoduše upravitelný programátorem, který by chtěl využít uspořádání objektů v hierarchii pro své vlastní účely. Nově vzniklé mapové vrstvy pak mohou být stejně jako kartogram, bodová mapa a mapa spojení rozšířeny o podporu tohoto nástroje a tím rozšířit svou vlastní funkcionalitu.

Nástroj pro stahování objektů lze v budoucnu rozšířit o uživatelské prostředí, které by poskytovalo možnost vytvořit si vlastní definici hierarchie bez použití automatického generování.

Literatura

- [1] *Významový střed území* [online]. Český statistický úřad [cit. 2021-12-16]. Dostupné z: <https://www.czso.cz/csu/rso/vyznamovy-stred-uzemi>.
- [2] AGAFONKIN, V. *Documentation - leaflet - a JavaScript library for interactive maps* [online]. [cit. 2022-3-21]. Dostupné z: <https://leafletjs.com/reference.html>.
- [3] ANELLI, V. W., CALÌ, A., NOIA, T. D., PALMONARI, M. a RAGONE, A. Exposing open street map in the linked data cloud. In: Springer. *International Conference on Industrial, Engineering and Other Applications of Applied Intelligent Systems*. 2016, s. 344–355.
- [4] BLACK, M. A. a CARTWRIGHT, W. E. Web Cartography & Web-enabled Geographic Information Systems (GIS) New Possibilities, New Challenges. In: *The 22nd International Cartographic Conference, A Coruña, Spain*. 2005, s. 9–16.
- [5] BOSTOCK, M. a METCALF, C. *Topojson/Topojson-specification: An extension to geojson that encodes topology and allows compact, delta-encoded coordinates*. 2018. Dostupné z: <https://github.com/topojson/topojson-specification>.
- [6] BUCKLEY, A. R., GAHEGAN, M. a CLARKE, K. Geographic visualization. In: *A research agenda for geographic information science*. CRC Press, 2004, s. 313–333.
- [7] BUTLER, H., DALY, M., DOYLE, A., GILLIES, S., SCHAUB, T. et al. *The GeoJSON Format* [RFC 7946]. RFC Editor, srpen 2016. Dostupné z: <https://www.rfc-editor.org/info/rfc7946>.
- [8] CAMPBELL, J. a SHIN, M. *Geographic Information System Basics*. Dec 2012 [cit. 2021-01-13]. Dostupné z: <https://2012books.lardbucket.org/books/geographic-information-system-basics/>.
- [9] DUŠEK, R. a MIŘIJOVSKÝ, J. Vizualizace prostorových dat: chaos v dimenzích. *Geografie–Sborník ČGS*. 2009, sv. 144, č. 3, s. 169–178.
- [10] ESRI. Shapefile technical description. *An ESRI white paper*. 1998, sv. 4, s. 1.
- [11] FONTANELLI, O., MIRAMONTES, P., COCHO, G. a LI, W. Population patterns in World's administrative units. *Royal Society open science*. The Royal Society Publishing. 2017, sv. 4, č. 7, s. 170281.
- [12] GIRARD, P. *Khartis - How to simply create thematic maps in three steps* [online]. FOSDEM VZW, 2018 [cit. 2022-4-25]. Dostupné z: <https://doi.org/10.5446/42146>.

- [13] GISGEOGRAPHY. *Vector vs Raster: What's the Difference Between GIS Spatial Data Types?* [online]. [cit. 2021-12-28]. Dostupné z: <https://gisgeography.com/spatial-data-types-vector-raster/>.
- [14] HALÁS, M., KLAPKA, P., BAČÍK, V. a KLOBUČNÍK, M. The spatial equity principle in the administrative division of the Central European countries. *Plos one*. Public Library of Science San Francisco, CA USA. 2017, sv. 12, č. 11, s. e0187406.
- [15] HYNEK, J., KACHLÍK, J. a RUSNÁK, V. Geovisto: A Toolkit for Generic Geospatial Data Visualization. In: *VISIGRAPP (3: IVAPP)*. 2021, s. 101–111.
- [16] KHADEMI, M., HAGHSHENAS, M. a KABIR, H. A review on authoring tools. In: *Proceedings of the 5th International Conference on Distance Learning and Education, IPCSIT*. 2011, sv. 12, s. 40–44.
- [17] KIMERLING, A. J. Dotting the dot map, revisited. *Cartography and Geographic Information Science*. Taylor & Francis. 2009, sv. 36, č. 2, s. 165–182.
- [18] KUČEROVÁ, H. *Geografická informace* [online]. Praha: KTD: Česká terminologická databáze knihovnictví a informační vědy (TDKIV), 2003 [cit. 2021-12-04]. Dostupné z: https://aleph.nkp.cz/F/?func=direct&doc_number=000000397&local_base=KTD.
- [19] MALTESE, V. a FARAZI, F. A semantic schema for GeoNames. Università di Trento. 2013.
- [20] NEUMANN, A., WINTER, A. M. a ATLAS, T. Chapter 12 - Webmapping with Scalable Vector Graphics (SVG): Delivering the Promise of High Quality and Interactive Web Maps. In: PETERSON, M., ed. *Maps and the Internet*. Oxford: Elsevier Science, 2003, s. 197–220. International Cartographic Association. ISBN 978-0-08-044201-3. Dostupné z: <https://www.sciencedirect.com/science/article/pii/B9780080442013500141>.
- [21] PORTELE, C. OpenGIS® Geography Markup Language (GML) Encoding Standard. Version 3.2. 1. Open Geospatial Consortium. 2007.
- [22] PRAVDA, J. a KUSEDOVÁ, D. Počítačová tvorba tematických máp. Univerzita Komenského. 2004.
- [23] SMUTNÝ, J. *Geografické informační systémy*. CERM, 1998.
- [24] SŁOMSKA PRZECH, K. a GOŁĘBIOWSKA, I. M. Do Different Map Types Support Map Reading Equally? Comparing Choropleth, Graduated Symbols, and Isoline Maps for Map Use Tasks. *ISPRS International Journal of Geo-Information*. MDPI AG. Feb 2021, sv. 10, č. 2, s. 69. ISSN 2220-9964. Dostupné z: <http://dx.doi.org/10.3390/ijgi10020069>.
- [25] TLČINA, A. *Webová aplikácia pre definíciu grafických objektov na mape*. Brno, CZ, 2021. Bakalářská práce. Vysoké učení technické v Brně, Fakulta informačních technologií. Dostupné z: <https://www.fit.vut.cz/study/thesis/23887/>.
- [26] TRAME, J. a KESSLER, C. Exploring the lineage of volunteered geographic information with heat maps. *GeoViz, Hamburg, Germany*. 2011.

- [27] VOSS, J. Encoding changing country codes for the Semantic Web with ISO 3166 and SKOS. In: *Metadata and Semantics*. Springer, 2009, s. 211–221.