



R2 Software

**Knihovna programů pro řízení dílčích procesů
automatizované tvorby topografických map**

Uživatelská příručka

Výzkumný ústav geodetický, topografický a kartografický, v.v.i.
T-MAPY spol. s r.o.

P R A H A 2 0 1 6

„Projekt č. TB04CUZK001 „Výzkum a vývoj metod pro kartografickou generalizaci státního mapového díla středních měřítek“ je řešen s finanční podporou TA ČR“.

Tato metodika je jedním z výstupů projektu TB04CUZK001.

T A
Č R

Program **Beta**

Obsah

Obsah dokumentu	1
Co je R2	2
Generalizační modul M1	3
Generalizační modul M2	5
Generalizační modul M3	7
Popis prostředí R2	10
Celková struktura	10
Struktura sdílených dat	11
Datové sady	13
Základní koncept	13
Implementace a funkce	13
Použití datových sad	14
Vizualizace dat	15
Příprava nové datové sady	16
Zdroj dalších informací	16
Ovládání software R2	17
Popis a nastavení modulu M1	18
Instalace	18
Struktura modulu	18
Technologie	18
Popis fungování modulu	18
Konfigurace	19
Příklady konfigurací	20
Popis a nastavení modulu M2	25
Instalace	25
Struktura modulu	25
Technologie	25
Popis fungování modulu	25
Konfigurace	26
Parametrizace procesních dávek	27
Parametrizace pravidel	27
Příklady konfigurací	29
Popis a nastavení modulu M3	30
Instalace	30
Struktura modulu	30
Technologie	30
Popis fungování modulu	30
Konfigurace	31

Obsah dokumentu

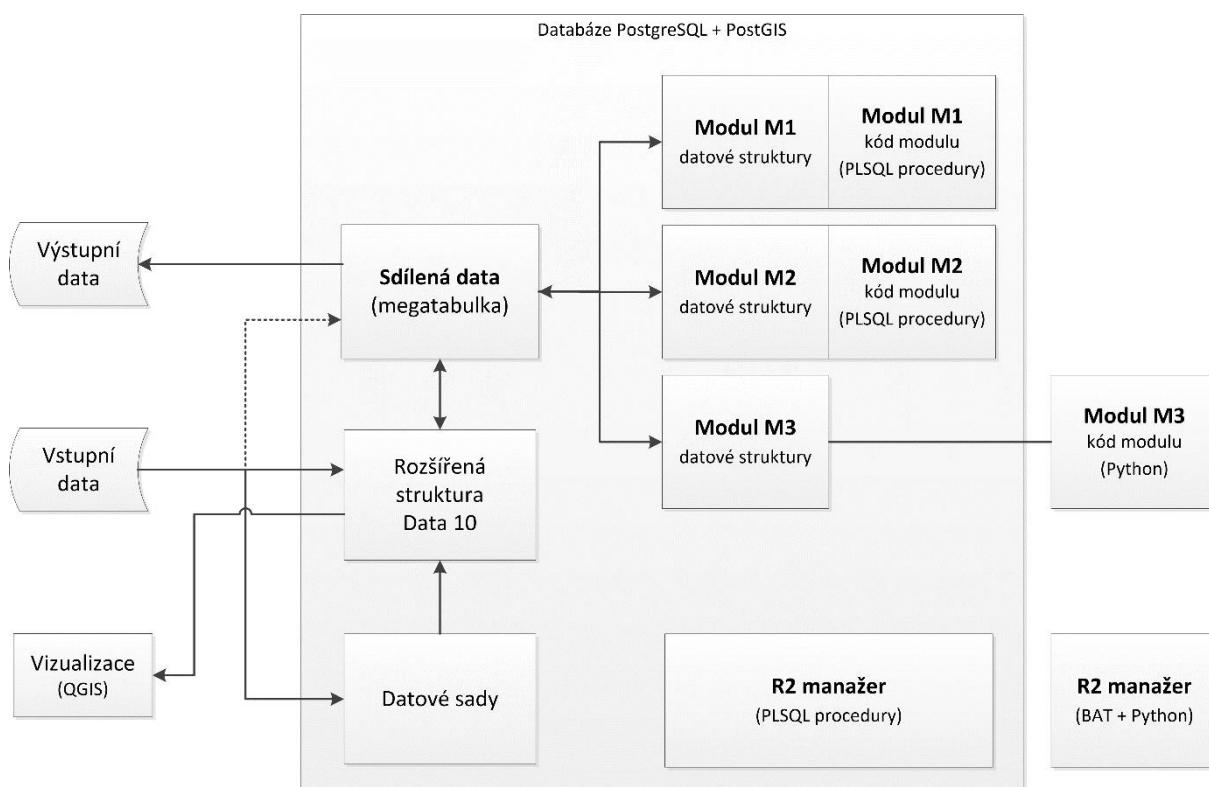
Dokument popisuje koncept R2 Software, jeho architekturu a jednotlivé moduly. Zejména se zaměřuje na popis prostředí, v němž jsou jednotlivé moduly implementovány, a způsob řízení generalizace. Důraz je kladen na seznámení s jednotlivými moduly, jejich fungováním a nastavením. Pro usnadnění nastavení jsou doplněny příklady konfigurací.

Co je R2

Architektura software R2 byla navržena na základě zkušeností z testování provedeného v rámci vývoje software R1 a s těmito cíli:

- **Modularita:** Jednotlivé funkční moduly jsou v maximální možné míře nezávislé a plní jednoznačně definované funkce. To zajišťuje flexibilitu jak při vývoji a testování, tak otevřenost pro budoucí úpravy a vývoj.
- **Minimalizace závislosti na struktuře dat:** Software by měl být, v míře dané celkovými podmínkami, nezávislý na struktuře vstupních dat. Tím je zajištěno, že případná změna struktury vstupních dat nebude mít za následek (nebo pouze omezené důsledky) porušení funkčnosti modulů.
- **Sjednocení použitých technologií:** Snížení počtu použitých technologií zjednodušuje instalaci software, používání a odstraňování případných chyb. Klade také menší nároky na obsluhu software.

Implementace architektury R2 vychází ze zvolené platformy a technologií, tedy databáze PostgreSQL s rozšířením PostGIS a jazyka Python:



Obrázek č. 1 Schéma architektury R2

Architekturu R2 lze rozdělit do následujících funkčních celků:

- **Úložiště dat:** Definovaná datová struktura, ke které přistupují moduly M1, M2 a M3 a která slouží jako centrální sklad zpracovávaných dat. Tento celek je implementován jako DB tabulka, více informací o této tabulce je k dispozici v kapitole *Popis prostředí R2, část Struktura sdílených dat*.
- **Generalizační moduly:** Každý modul (M1, M2, M3) je samostatným celkem, který je z hlediska instalace, konfigurace a používání nezávislý na ostatních modulech. Každý modul v databázi využívá jedno nebo více schémat a je

implementován buď jako sada PLSQL funkcí nebo Python moduly. Podrobnosti o jednotlivých modulech jsou uvedeny v samostatných kapitolách.

- **Řídící struktury a funkce (R2 manažer):** Obslužné funkce zajišťující spouštění software R2, případně jednotlivých modulů, a manipulaci s daty. Funkce jsou implementované jako PLSQL funkce, doplněné pro jednodušší používání dávkovým souborem, a jako Python modul.
- **Úložiště připravených datových sad:** Toto úložiště nemá přímou vazbu na funkčnost software R2, zjednodušuje ale jeho používání a konfiguraci jednotlivých modulů i software jako celku.

Generalizační modul M1

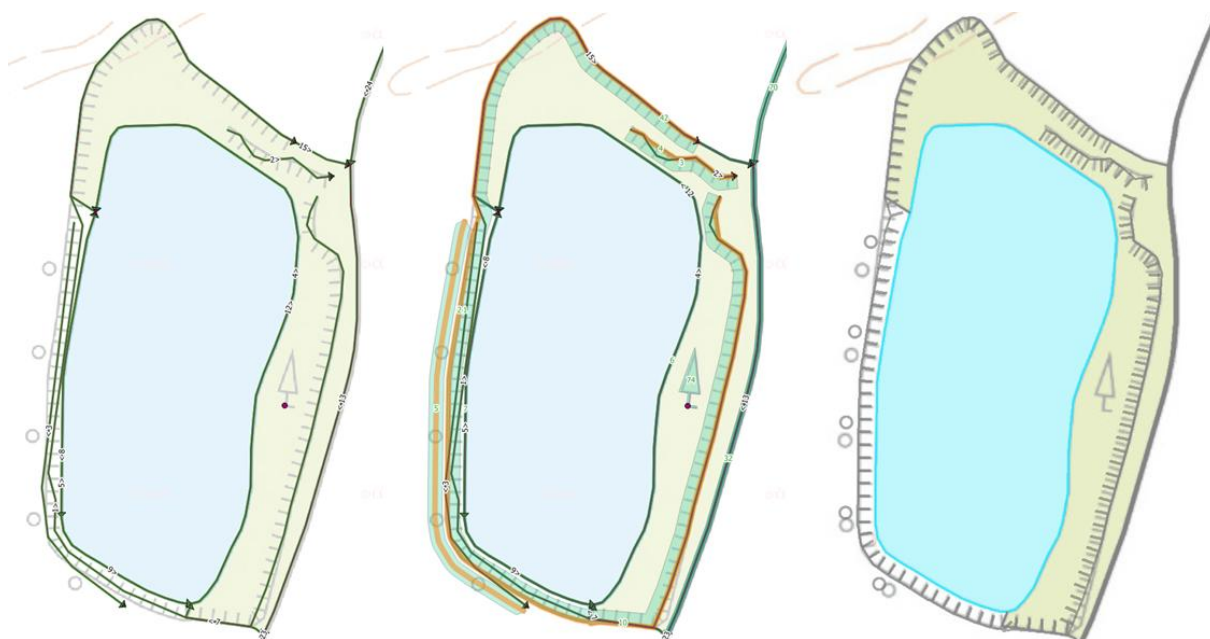
Generalizační modul M1 je navržen jako specifický modul pro řešení generalizace metodou odsunu prvků v izolovaných a méně komplexních generalizačních situacích. Pro realizaci je navržen a vyvinut komplexní operátor odsunu.

Modul M1 využívá generalizačního přístupu, ve kterém se uplatňují prvky Rule Based Generalization a Constraint Based Modelling, přičemž jednotlivá omezení a podmínky (constraints) jsou omezeny na množinu tzv. hard constraints a vyhodnocení jejich porušení. U tohoto typu constraint je vždy možné jednoznačně rozhodnout, zda v daném kontextu bylo splněno či nikoli. Modul M1 při vyhodnocení akceptuje jen taková řešení, kdy definovaná constraints nejsou v řešeném kontextu porušena.

Cílem řešení je eliminovat konflikty zejména liniových objektů v kartografickém modelu, který bude použit k sestavení mapy. Objekty jsou v kartografickém modelu symbolizovány signaturami (mapovými značkami, mapovými znaky). Signatury bodových, liniových i plošných prvků zařazených do kartografického modelu vytvářejí obraz mapy. V kartografickém modelu lze jednotlivé prvky nahradit stopou (tzv. footprintem) vymezenou jejich signaturou. Tento footprint má pro bodové, liniové i plošné prvky vždy areálový charakter, který lze vyjádřit objektem geometrického typu polygon. Pro účely identifikace konfliktů objektů v kartografickém modelu je každému prvku přiřazen footprint. Ten může věrně kopírovat stopu signatury nebo může být do určité míry zjednodušen. To platí zejména u složených signatur, kdy např. liniový prvek je symbolizován linií určité tloušťky v kombinaci s bodovými signaturami pravidelně rozmístěnými na definiční linii prvku. V takovém případě je možné nahradit přesný footprint zjednodušeným footprintem tvořeným bufferem kolem definiční linie prvku o šíři velikosti bodové signatury. Rovněž pro signatury bodových prvků může být footprint geometricky zjednodušen (např. kruhový oblouk bude nahrazen lomenou čarou), zejména z výpočetních důvodů. Míru zjednodušení footprintu je vhodné volit tak, aby zbytečně nedocházelo k identifikaci falešných konfliktů nebo naopak aby nezůstaly některé skutečné konflikty opomenuty.

Modul M1 pracuje s obecně definovanými entitami a vztahy mezi nimi. Proces generalizace je řízen parametry, které jsou definovány konfiguračně v XML souboru. Znamená to, že tentýž modul lze použít pro generalizaci libovolného kartografického modelu, který je popsán v sémantice generalizačního modulu M1. Základní entitou tohoto modelu je *element*. Jedná se o prvek kartografického modelu jednoznačně definovaných charakteristik a chování. Jednotlivé elementy jsou výskytem entity *element typ*, přičemž každý element typ má v rámci celého kartografického modelu jedinečné charakteristiky, chování a vztahy k ostatním entitám. Každý element typ je popsán jednoznačným identifikátorem, geometrickým typem, vahou (důležitostí), barvou (v případě areálových prvků barvou obrysové linie), původem (např. v případě vstupního modelu v Esri Geodatabase organizací prvku v rámci struktury Feature Data Set a Feature Class), definicí footprintu a příznakem, zda má topologické vztahy vůči dalším entitám. Popis element typu může být doplněn dalšími vlastnostmi, např. příslušností k sémantickým kategoriím kartografického modelu. Chování element typu je popsáno výčtem operátorů,

kteřé lze na element této entity aplikovat. Model je doplněn popisem vztahů mezi jednotlivými entitami, typicky mezi elementy téhož element typu, mezi elementy dvojice element typů či mezi elementy skupin element typů navzájem. Jde především o vyjádření, zda jsou relačně svázané entity při prostorovém překryvu signatur v konfliktu, případně jaká je požadovaná světlost (rozestup, mezera, clearance) mezi signaturami entit. Obecné (generické) vztahy mezi entitami jsou přepisovány specifickými vztahy popsány v konfiguraci modelu. Příkladem může být definice specifického footprintu prvku terénní stupeň v případě řešení konfliktu s prvkem stromořadí, kdy bodová signatura na linii stromořadí může zasahovat do spádových čárek signatury terénního stupně, ale nikoli do jeho koruny.



Obrázek č.2 Příklad řešení odsunů v rámci izolované situace modulem M1

Modul M1 využívá prostorového uložení dat v DB PostgreSQL s prostorovým rozšířením PostGIS. Vstupem jsou data výchozího topografického modelu transformované do modelu elementů kartografického modelu. V něm jsou elementy reprezentující bodové, liniové a plošné prvky uloženy společně bez ohledu na jejich geometrický typ. Tento způsob uložení umožňuje každému elementu přiřadit a uchovat nejen vstupní geometrii převzatou z topografického modelu, ale rovněž footprint signatury a cílovou geometrii získanou generalizačním procesem. V pracovní části databáze vymezené modulem M1 je vytvořen datový model dle kartografického modelu popsaného v konfiguraci a z výchozího úložiště jsou převzaty všechny elementy popsané v konfiguraci modelu. Do datového modelu jsou zapsány charakteristiky všech definovaných entit, jejich chování a vzájemných vztahů včetně topologických. Pro elementy zařazené do topologie je vytvořen topologický model, v němž jsou jednotlivé elementy vyjádřeny jejich topogeometriemi složenými ze základních konstrukčních entit node, edge a face. Do pracovního datového modelu pro generalizaci jsou zahrnuty topologické i netopologické elementy. Pro všechny elementy pracovního datového modelu jsou vypočteny jejich footprinty, přičemž jeden element může mít definováno v závislosti na konfiguraci modelu jeden či více footprintů.

Proces vlastní generalizace probíhá tak, že jsou postupně procházeny elementy seřazené sestupně podle váhy, která je jednotlivým elementům typům přiřazena. Pro každý řídicí element (vodící element, master element) je zjištěno výchozí kolizní místo. V konfliktní zóně jsou seřazeny všechny elementy dle vzdálenosti od řídicího elementu. Postupně jsou řešeny konflikty řídicího elementu a nejbližšího konfliktního elementu (závislého elementu, slave elementu), přičemž konflikty elementů jsou detekovány vždy vůči

aktuální geometrii elementů. Zároveň je zaručeno zachování pořadí závislých konfliktních elementů vůči řídicímu elementu v konfliktní zóně v průběhu řešení konfliktu. Rekursivně jsou procházeny vzdálenější konfliktní elementy dle principu, že pro následnou iteraci je původní závislý element zafixován, stává se řídicím elementem a přebírá váhu iniciačního řídicího elementu. Po každém elementárním odsunu je provedeno vyhodnocení, zda se podařilo konflikt vyřešit v souladu s definovanými podmínkami a omezeními (constraints). V případě nesouladu jsou neakceptované úpravy vráceny do předchozího stavu, což je umožněno transakčním způsobem řešení. Po vyřešení všech konfliktů výchozího řídicího elementu jsou iterativně řešeny konflikty elementů všech element typů s výjimkou nekonfliktních. Po projití všech elementů je provedena rekonstrukce topologických elementů a provedené změny jsou promítnuty do cílové geometrie modelu elementů kartografického modelu včetně naplnění stavových atributů.

Při řešení konfliktu elementů odsunem je na základě identifikace konfliktní zóny vypočtena distanční linie, která představuje v konfliktní zóně průběh geometrie závislého elementu, která nezpůsobí konflikt signatur řídicího a závislého elementu. Dále jsou vypočteny vektory posunu geometrie závislého elementu na distanční linii a provedena transformace geometrického průběhu závislého elementu.

Ve specifických vztazích element typů je podporován princip ztotožnění signatur řídicího a závislého elementu v celém jejich průběhu nebo jejich části. Z hlediska vzájemné polohy obou elementů jsou podporovány následující typy ztotožnění signatur: ztotožnění průběhu linie (distanční linie inciduje s geometrií řídicího elementu), ztotožnění hrany průběhu kresby (distanční linie inciduje s obrysovou linií signatury řídicího elementu), ztotožnění průběhu linie na dotyk (distanční linie ve vzdálenosti aritmetického průměru šířek signatur obou elementů od geometrie řídicího elementu) a ztotožnění průběhu linie s rozestupem (distanční linie ve vzdálenosti aritmetického průměru šířek signatur obou elementů od geometrie řídicího elementu zvětšené o hodnotu clearance).

Generalizační modul M1 je vhodný pro řešení konfliktů signatur elementů odsunem v izolovaných a méně komplexních generalizačních situacích. Užití vyvinutého operátoru odsunu není vázáno na předem definovaný kontext generalizační situace.

Generalizační modul M2

Generalizační modul M2 je navržen jako obecný rozšiřitelný modul pro řešení kartografické generalizace. Rozšiřitelný je ve smyslu možnosti přidávání pravidel, parametrů a algoritmů pro zpracování geometrie kartografických objektů. Jako obecný je označován vzhledem k ambici řešit všechny typy konfliktů kartografické reprezentace a možnosti definovat zpracování pro různé typy a měřítka map.

Modul je postaven na několika základních principech:

1. Znalostně omezený "útok hrubou silou". "Útok hrubou silou" je chápáno generování kombinace přípustných stavů geometrické a klasifikační charakteristiky kartografických objektů získané aplikací generalizačních algoritmů. Tato kombinace je testována na absenci konfliktu (s tím, že je uchováván stav s nejmenší mírou konfliktu) a při prvním takovém stavu ukončena. Znalostní omezení tohoto přístupu je dáno jednak určením přípustných operací nad kartografickými objekty jejich příslušností k třídě, jejím podskupinám a jejich lokální prostorovou konfigurací. Dalším omezením je uspořádání přípustných operací podle míry jejich dopadu a úspěšnosti v dosavadním běhu modulu.
2. Lokální omezení konfliktu k zamezení dominového efektu. Objekty v konfliktu jsou rozděleny do shluků určených aktuálním konfliktem a potenciálním konfliktem vzniklým manipulací dotčených objektů generalizačními operátory. Tento shluk je

ovšem omezen prostorovým dosahem hlavního konfliktu, který je určen arbitrárně s ohledem na měřítko.

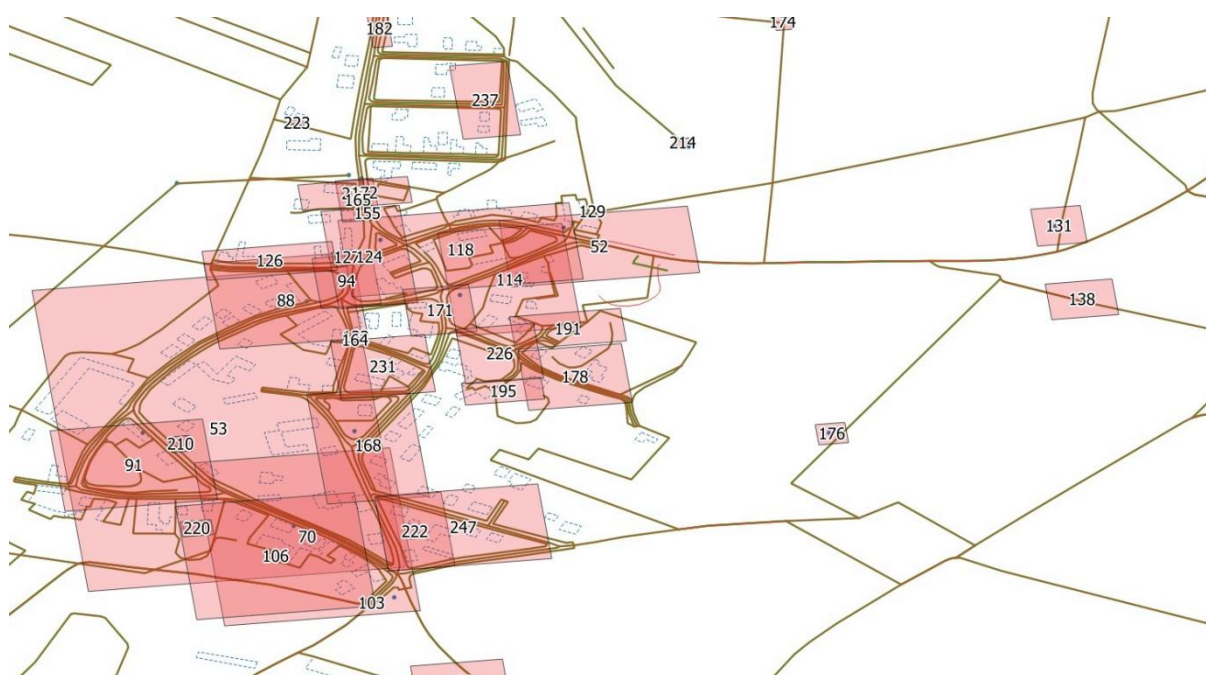
3. Topometrická měření pro identifikaci a konstrukci pravidel. V rámci prostorového rozsahu konfliktního shluku je provedena soustava topologických a metrických měření, která slouží ke kontrole topologických vazeb a stanovení odvozených pravidel.
4. Kódování znalostí do explicitně vyjádřené, perzistentní soustavy záznamů. Jádrem těchto záznamů jsou následující tabulky:
 - a. **maps** - popis obsahu mapy zahrnující identifikaci tříd a základní parametry jejich symbolizace. Tato tabulka slouží k zjištění parametrů symbolizace objektů a k odvození generických omezení.
 - b. **constraints** - podmínky čitelnosti mapy. Generické podmínky jsou odvozeny z parametrů tabulky maps. Další specifická omezení mohou být doplněna uživatelem.
 - c. **rules** - seznam přípustných generalizačních funkcí pro kartografický objekt (skupinu, třídu) v určité situaci. Pravidla mohou být uživatelská nebo odvozená z běhu modulu.
 - d. **contexts** - vztah kartografického objektu k jeho okolí a specifikace jeho vlastností vztahující se k pravidlu. Specifikace kontextů je součástí definice pravidel
 - e. **structures** - prostorové konfigurace kartografických objektů ovlivňující pravidla, která jdou nad rámec popisu obsahu mapy
 - f. **thesaurus** - popis agregovaných identifikací skupin kartografických tříd umožňující definici obecnějších pravidel.

Mimo tyto základní znalosti je udržován seznam dostupných algoritmů a záznam jejich úspěšnosti při řešení konfliktů.

5. Zatřídění a uspořádání konfliktů do 3 skupin:
 - a. **imperceptibilita** - nemožnost korektního zobrazení. Vedle celkové nepatrnosti objektu zahrnuje i problémy s vyjádřením průběhu tvaru. Je řešena jako první pomocí odstranění, kolapsu, zjednodušení a škálování.
 - b. **komplikace** - interakce skupiny objektů, kde existuje jednoznačně definovaný postup úpravy v rámci dané mapy. Vedle explicitně definovaného chování objektů zahrnuje i problémy nahloučení, souběhu a gestaltu. Řešena je vyžádanou úpravou průběhu, typifikací, agregací a vyčištěním.
 - c. **kolize** - symbolika kartografických objektů se překrývá. Jedná se o závěrečný běh řešený pomocí odsunů, odstranění, odřezání a deformace.
6. Postupné řešení od nejjednoduššího konfliktu. Modul se snaží primárně vyřešit s nejmenší četností. Předpokládá nenalezení bezkonfliktního stavu a pamatuje si nejlepší dosažený stav. Na výstupu je možné stanovit threshold pro akceptaci řešení. V modulu je také možné nastavit časový zámek pro dobu řešení konfliktního shluku.
7. Modul předpokládá běh ve dvou režimech:
 - a. striktní - běh je založen na uživatelských pravidlech
 - b. adaptivní - modul za běhu definuje nová pravidla na základě úspěšných řešení, která postupně nahrazují uživatelská pravidla na bázi podobnosti kontextů nově definovaných pravidel.
8. V rámci certifikace se předpokládá testování striktního režimu, adaptivní režim vyžaduje k efektivnímu použití rozsáhlejší testování a stabilní implementaci CBR rezolučního stroje.

V rámci výše zmíněného konceptu probíhá zpracování mapy následujícím způsobem:

1. vyhledání a řešení imperceptibility
2. vyhledání struktur zmíněných v pravidlech pro komplikace
3. řešení komplikací a případné sdružení vyřešených objektů
4. identifikace kolizních shluků a jejich uspořádání
5. topometrická měření
6. sestavení plánu řešení kolizních shluků (nalezení pravidla, algoritmu a parametrického rozsahu)
7. postupná aplikace algoritmů na objekty zohledňující minimalizaci změny
8. kontrola omezení po každé jednotlivé změně
9. zápis výsledku, odvozeného pravidla, statistiky průběhu zpracování.



Obrázek č. 3 Zóny konfliktních shluků identifikovaných kolizí

Generalizační modul M3

Generalizační modul M3 je navržen jako obecný modul pro modelování a řešení generalizace.

Požadavky na kartografické zpracování jsou reprezentovány využitím multi-agentního systému pomocí agentů, který není pro tento účel chápán jako striktně hierarchický (meso agents, micro agents, element agents).

Agent může být vázán na různé entity, například:

č.	Entita	Agent
1.	Grafický element	Prvek trigonometrický bod musí být zobrazen beze změny polohy
2.	Grafický element	Průběh silnice musí být zakreslen s přesností 0,8 mm v měřítku mapy

3.	Část elementu	Při souběhu dvou liniových kreseb (jednočará element, lemovka výplně, lemovka dvoučará kresby) v délce více než 2 cm, oscilaci méně než 0,3 mm a vzdálenosti menší než 5 mm v měřítku mapy musíme zabezpečit souběh průběhu.
4.	Část elementu	Kresba rohů budovy by měla být patrná a bodové značky by je neměly překrývat.
5.	Skupina elementů	Při generalizaci budov v rámci vnitrobloku musí být zachován jeho charakter. Tato charakteristika je modelována jako skupina agentů: - elementy patřící do vnitrobloku - rozeznané strukturální vzory
6.	Konfigurace objektů	Úhel napojení komunikací v křižovatkách by měl být zachován.

Tabulka č.1 Příklad agentů a jejich vazby na entitu

Základní vlastností každého agenta je schopnost stanovit číselně stanovit hodnotu splnění kritéria, které reprezentuje. Rozdíl hodnot před provedením nějakého kartografické úpravy (aplikováním algoritmu operátoru generalizace) a po něm reprezentuje úspěšnost této úpravy.

Každá z těchto operací může mít vliv na splnění jiných požadavků na stejný element či na elementy okolní. Za účelem komplexního stanovení úspěšnosti nějaké úpravy hodnotu splnění kritéria převádíme do normalizovaného tvaru tak, aby je bylo možno mezi sebou porovnávat v intervalu $[0, 1]$. Hodnota 0 označuje stav kdy je požadavek splněn, hodnota 1 první akceptovatelnou hodnotu. Průběh mezi nimi musí být patřičným způsobem linearizován.

Pro průběh kresby silnice v předchozí tabulce v mapě s požadovanou přesností zákresu 0,2 mm v měřítku mapy tedy platí:

2. Přesnost průběhu kresby silnice	Vyhodnocení	Normalizováno
0,1 mm	Spokojen	0
0,2 mm	Spokojen	0
0,5 mm	Akceptovatelné	0,5
0,8 mm	Nespokojen	1
1,0 mm	Nespokojen	1

Tabulka č.2 Příklad vyčíslení požadavku reprezentovaného agentem

Druhou vlastností agenta je seznam algoritmů generalizačních operátorů (odsun, vypuštění apod.), které potenciálně mohou směřovat ke zlepšení splnění kritéria, které agent reprezentuje. Seznam je seřazen tak, aby preferované metody řešení byly první. Vypuštění elementu tedy bývá až na konci seznamu, pokud žádné řešení nevede k akceptovatelnému zlepšení.

Seznam povolených algoritmů je patrný z následující tabulky:

č.	Agent	Povolené algoritmy
1.	Prvek trigonometrický bod musí být zobrazen beze změny polohy	Vrácení do skutečné polohy
2.	Průběh silnice musí být zakreslen s přesností 0,8 mm v měřítku mapy	Vrácení do skutečné polohy Vypuštění elementu
3.	Při souběhu dvou liniových kreseb (jednočará element, lemovka výplně,	Ztotožnění průběhu linií (paralelizace) Vypuštění elementu

	lemovka dvoučaré kresby) v délce více než 2 cm, oscilaci méně než 0,3 mm a vzdálenosti menší než 5 mm v měřítku mapy musíme zabezpečit souběh průběhu.	
4.	Kresba rohů budovy by měla být patrná a bodové značky by je neměly překrývat.	Vypuštění budovy
5.	Při generalizaci budov v rámci vnitrobloku musí být zachován jeho charakter. Tato charakteristika je modelována jako skupina agentů: - elementy patřící do vnitrobloku - rozeznané strukturální vzory	Viz agenti ve skupině
6.	Úhel napojení komunikací v křižovatkách by měl být zachován.	Modifikace vrcholů do požadovaného rozsahu Vypuštění elementu

Tabulka č.3 Příklad vazby operátorů generalizace na agenty

Řešení požadavků na kartografickou kresbu je možné mnoha různými způsoby. Navržený model je otevřený jak multiagentnímu řešení, metodě nejmenších čtverců, simulated annealing i gradient descent search podle potřeby, kontextu a počtu prvků.

Pro mapy ZM 10 se ukázal jako vhodný následující přístup:

1. Normalizujeme databázi do modelu grafika+symbol
2. Postupně umísťujeme jednotlivé elementy kresby podle priorit
3. Při umísťování elementů vytvoříme model agentů vztažený k tomuto elementu, včetně případných strukturálních vzorů
4. případnou nespokojenost s vyhodnocením řešíme postupně využitím dostupných algoritmů v plném rozsahu
5. při aplikaci operátorů okamžitě reagujeme na zhoršení ostatních agentů v kontextu
6. pokud žádná z dostupných metod nevedla k akceptovatelnému řešení, provedeme totéž iterativně s krokem 5 % optimálních hodnot.

Je zřejmé, že tento postup lze v dalším projektu optimalizovat, například zabránit opakovanému počítání jednotlivých hodnot a strukturálních vzorů jejich podržením v paměti či ukládáním do pomocné databáze, volit další umístěný prvek co nejbližší stávajícímu apod. Je také patrné, že uvedený postup bude vyhovovat kontextům v určité složitosti konfigurace. Jelikož postup kopíruje základní postup kartografa při ručním sestavení mapy a generalizaci, je pravděpodobné, že jejich množství bude velké. V ostatních případech je vhodné doplnit vytvořený model dalšími, účelovými strategiemi řešení. Zatímco výčet algoritmů řešení a normalizace hodnot "spokojenosti" je konečný je pravděpodobné, že při aplikaci v menších měřítcích ZM bude potřeba doplňovat algoritmy strukturálních vzorů a stanovovat parametry pro vyčíslení. Strategie řešení je navržena tak, že aplikuje operaci pouze tak, aby nedošlo ke zhoršení jiného kritéria.

Popis prostředí R2

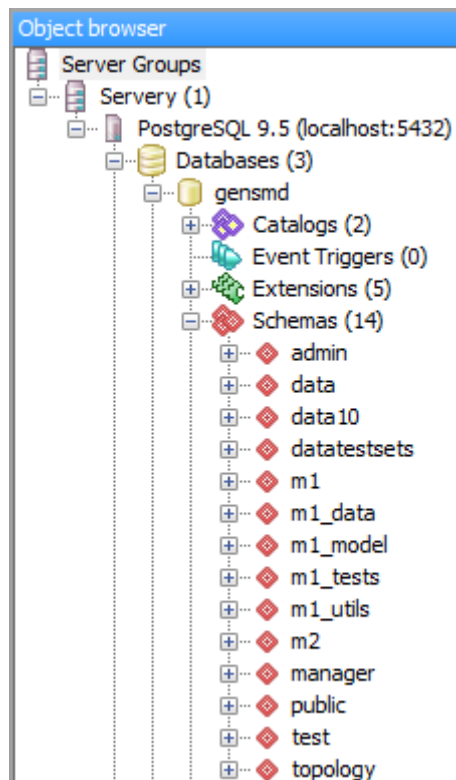
Tato kapitola poskytuje podrobnější informace o technickém řešení (prostředí) software R2.

Celková struktura

Software R2 se skládá z databázové a souborové části.

Databáze

Jednotlivé funkční moduly R2 jsou implementované jako jedno nebo více DB schémat obsahujících tabulky, funkce a data.



Význam jednotlivých schémat uvádí tabulka:

Schéma	Popis
admin	Funkce používané především při vývoji a instalaci některých modulů.
mX_*	Schémata obsahující kód, tabulky nebo data jednotlivých generalizačních modulů M1, M2 a M3.
data	Schéma obsahující sdílená data v tabulce elements (viz. <i>Struktura sdílených dat</i>).
manager	Funkce sloužící k manipulaci s daty (konverze z a do tabulky data.elements, příprava sad atd.).
test	Datová struktura vycházející z data10 zjednodušující konverzi do a z tabulky data.elements.
datatestsets	Připravené sady testovacích dat.
data10	Import vstupních dat.

Složky a soubory

Struktura složek software R2 je identická se strukturou repozitory TB04CUZK001_R2. Struktura obsahuje jak soubory nutné pro instalaci R2, tak pro jeho používání.

Význam jednotlivých složek uvádí tabulka:

Složka	Popis
admin	Instalační soubory schématu admin a data
data	Datové soubory (připravené testovací sady) a dávkové soubory pro manipulaci s nimi.
install	Instalační soubory pro základní software
m1	Instalační soubory modulu M1
m2	Instalační soubory modulu M2
m3	Programové moduly M3
manager	Dávkové soubory pro spouštění a další operace v R2.

V některých případech je obsah složek podrobněji popsán v souborech README.MD

Struktura sdílených dat

Jak bylo uvedeno v popisu architektury R2 (kapitola Co je R2), využívají moduly M1, M2 a M3 centralizované úložiště zpracovávaných dat. Toto uložení má podoby tabulky data.elements (tj. tabulka elements ve schématu data) s touto strukturou:

Sloupec	Typ Formát / zdrojová data	Popis Příklad
elm_id	seriál ×	Jednoznačný identifikátor (sérové číslo) záznamu v tabulce Příklad: 123
source_id	character varying(100) zdrojová_tabulka+"."+objectid	Identifikace zdrojového prvku ve struktuře data10. Příklad: Z_Budova_P:86447
source_elt_id	character varying(50) typ_geometrie+znacka (typ geometrie = B,L,P)	Typ geometrie a identifikátor značky (z data10). Příklad: P1010100
source_rotation	numeric(7,4) sloupec „natoceni“ nebo 0.0, pokud sloupec neexistuje	Natočení prvku. Pokud ve zdrojové tabulce informace o natočení neexistuje, je natočení 0.
source_geom	geometry(Geometry,5514) wkb_geometry	Vstupní (původní) geometrie prvku, převzatá z data10 a převedená z konkrétní geometrie (bod, linie, polygon) na obecný geometrický typ.
target_elt_id	character varying(50) typ_geometrie+znacka (typ geometrie = B,L,P)	Identifikace výsledného prvku, typicky identické se zdrojovým prvkem.
target_rotation	numeric(7,4)	Natočení prvku. Pokud pro daný prvek natočení neexistuje, pak 0.
target_geom	geometry(Geometry,5514)	Výstupní (nová) geometrie prvku.

Sloupec	Typ Formát / zdrojová data	Popis Příklad
state	integer 0..4	Poslední úprava prvku, hodnoty: 0 ... bez změny 1 ... změna geometrie (průběhu) 2 ... prvek byl odstraněn 3 ... změna typu prvku (např. náhrada plošného prvku značkou) 4 ... byl vytvořen nový prvek
log	text	Nestrukturovaný záznam jednotlivých modulů

Vzhledem k tomu, že tabulka obsahuje všechna generalizovaná data bez ohledu na jejich geometrii a zdrojovou tabulku, označuje se jako megatabulka.

Datové sady

Tato kapitola shrnuje informace o používání připravených datových sad, jejich struktuře, přípravě datových sad a jejich vizualizaci.

Základní koncept

Koncept datových sad není součástí řešené problematiky, je ale podpůrným nástrojem, který zjednodušuje vývoj modulů, jejich testování a také případnou úpravu parametrů jednotlivých modulů.

Vlastnosti

Datová sada je připravená skupina dat, kterou si lze představit jako vybranou skupinu tabulek ve struktuře data10, které jsou naplněny konkrétními daty.

Datová sada je identifikována svým jednoznačným identifikátorem (libovolný text); datová sada může obsahovat také popis.

Příklady datových sad vytvořených v rámci implementace R2:

- Data vybraných testovacích situací (např. sada 2 s popisem *Postupný odsun stromořadí a zářezů od silnice.*).
- Doplnková data, používaná programátory při vývoji modulů (např. sada *tmapy1* s prvky z vybrané oblasti).
- Mapové listy (např. sada *ms2* obsahující prvky z mapového listu ZM10 02-41-06).

Výhody

K nejdůležitějším výhodám implementace datových sad do software R2 patří:

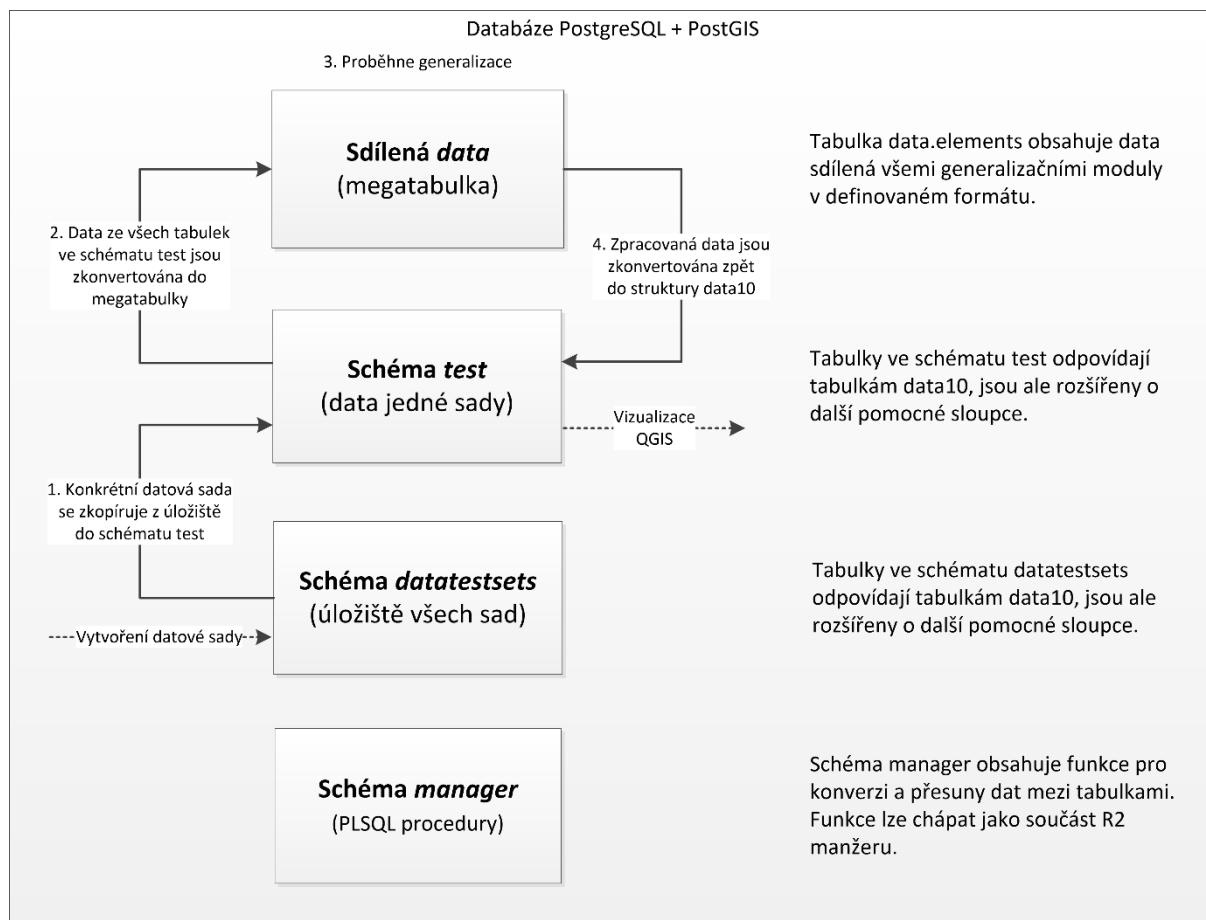
- Zavádí se pojem „datová sada“ (případně „test set“) ve smyslu připravených dat o definované struktuře.
- Není vyžadován časově náročný import GDB data10 do databáze.
- Není vyžadován jiný způsob importu dat (např. pomocí ogr2ogr), který by vyžadoval specifické znalosti.
- Umožňuje velmi jednoduše a rychle vracet se k vybraným datům a tyto kopírovat do tabulky sdílených dat, což umožňuje operativní testování např. rozdílů mezi moduly nebo vlivu úpravy parametrů na konkrétní situaci.
- Je zajištěna nezávislost na případných změnách vstupních dat.
- Jde o rozšíření a automatizaci standardních postupů, které je zcela nezávislé na ostatních částech software R2.

Implementace a funkce

Datové sady jsou implementovány takto:

- **Schéma test:** Obsahuje vždy jednu datovou sadu a představuje rozhraní mezi úložištěm datových sad a megatabulkou. Schéma test obsahuje 41 tabulek, které odpovídají datovému modelu data10, tyto jsou ale rozšířeny o další pomocné sloupce.
- **Schéma datatestsets:** Představuje úložiště všech připravených datových sad. Obsahuje 42 tabulek, z nichž 41 odpovídá datovému modelu data10, ale s nezbytným rozšířením o další sloupce, 1 tabulka představuje číselník datových sad a zajišťuje datovou integritu (vazba klíčů mezi číselníkem a jednotlivými záznamy).
- **PLSQL funkce** ve schématu manager zajišťují přesuny a konverze dat a jsou považovány jako součást skupiny funkcí R2 manažer.

Schéma funkce datových sad:



Převod dat ze schématu test do megatabulky je implementován s ohledem na skutečnost, že některé tabulky v datovém modelu data10 neobsahují sloupec znacka (např. tabulka m_ostatni10_b) a přímý převod proto není možný a bez údaje o značce nemusí algoritmy v generalizačních modulech fungovat správně. Proto je součástí převodu také číselník (jde o tabulky manager.ruleid2znacka), podle které se podle jména tabulky a hodnoty ruleid dohledává odpovídající hodnota údaje znacka.

Použití datových sad

Pro použití datových sad slouží PLSQL funkce ve schématu manager. Nejdůležitější pro koncového uživatele jsou:

- **manager.load_testcase('<id situace>')**: Funkce z úložiště (schéma datatestsets) vybere všechna data náležející k zadané situaci a zkopíruje je do schématu test. Následně tato data zkonvertuje do formátu megatabulky a zkopíruje do data.elements. Tato funkce se používá před spuštěním generalizace.
- **manager.get_results_from_megatable()**: Funkce zkonvertuje data z megatabulky zpět do odpovídajících tabulek. Tato funkce se používá po skončení generalizace.

Poznámka: Konverzní funkce manager.get_results_from_megatable() neřeší, vzhledem k velmi specifické povaze, převod objektů, u kterých došlo ke změně typu geometrie (např. náhrada plošného objektu bodovou značkou). Pro vizualizaci takto generalizovaných objektů je potřeba použít data přímo z tabulky data.elements.

Součástí software R2 jsou také dávkové soubory, pomocí který je možné:

- Obsah tabulek ve schématu test a tabulku data.elements vyexportovat do externích souborů.
- Obsah externích souborů naimportovat zpět do původních tabulek.

To umožňuje:

- Přenášet snadno a standardizovaným způsobem výsledky generalizace mezi více počítači.
- Uchovávat mezivýsledky zpracování dat moduly M1, M2 a M3.

Vizualizace dat

Pro vizualizaci dat je připraven QGIS projekt, pomocí kterého je možné zobrazit data ve schématu test. Zobrazení dat v prostředí QGIS slouží k rychlému základnímu posouzení výsledků generalizace, nejedná se o detailní a kartograficky věrné zobrazení všech dat s využitím úplného značkového klíče SMD 10.

Zobrazené vrstvy

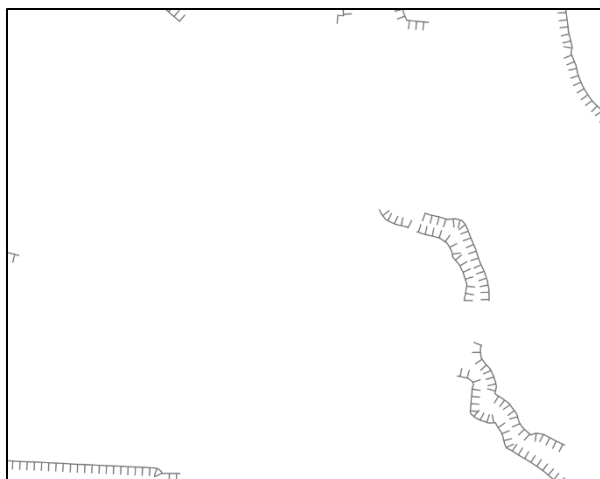
QGIS projekt zobrazuje obsah 32 tabulek, nejde tedy o úplné zobrazení všech tabulek ze schématu test (to obsahuje celkem 41, klasifikovaných v souboru FC_properties.xls jako vstup do generalizace). Každá vrstva v QGISu představuje jednu tabulku v datové struktuře data10 a tedy jeden typ objektu.

Organizace vrstev v QGIS

Vrstvy v projektu jsou rozděleny do následujících skupin:

- **ZM 10 Source Data:** Objekty vstupující do projektu generalizace (sloupec wkb_geometry).
- **ZM 10 Changes Data:** Objekty, jejichž geometrie byla změněna (state=1 a geometrie ve sloupci zm10_geom).
- **ZM 10 Generalized:** Zobrazují se objekty se změněnou, případně původní geometrií (pokud geometrie nebyla při generalizace upravena), nezobrazují se odstraněné prvky (state=2).
- **ZM 10 Removed:** Zobrazují se pouze odstraněné prvky.

Ukázka zobrazení dat v prostředí QGIS:



Zobrazení pouze prvků, u kterých byla při generalizaci upravena geometrie.



Zobrazení výsledku, tj. generalizovaných a původních prvků bez odstraněných.

Příprava nové datové sady

Příprava nové datové sady sestává z kroků:

- **Zajištění vstupních dat:** Součástí instalace je import dat ve struktuře data10 do schématu data10. Tato data mohou být jednoduše použita jako zdroj vstupních dat, alternativně lze ale využít i jakékoliv jiné externí zdroje.
- **Vytvoření záznam v číselníku sad:** Každá sada je identifikována jedinečným identifikátorem v tabulce `datatestsets.testsetslist`, volitelně může mít také popis.
- **Identifikace**, které objekty ze zdrojových dat se mají do sady vložit. Pomocí připravených funkcí je možná identifikace podle:
 - `zdroj_id` (výchozí identifikace)
 - `objectid` nebo `featureid` (pro tabulky, které `zdroj_id` neobsahují)
 - rozsahem souřadnic (pomocí PostGIS funkce `ST_MakeEnvelope`)
 - kladem mapových listů
- **Vytvoření SQL skriptu**

Zdroj dalších informací

Jako doplňující zdroj informací lze využít:

- Dávkové soubory pro pouštění generalizačních modulů.
- Skripty, které byly použity pro vytvoření připravených datových sad. Všechny vytvořené skripty jsou součástí repository ve složce `data`.
- Dokumentace funkcí ve schématu `manager`, které jsou součástí skriptu `\TB04CUZK001_R2\manager\sql\create_functions.sql`

Ovládání software R2

Pro ovládání software R2 je připravena sada dávkových souborů a souvisejících PLSQL funkcí v PostgreSQL databázi shrnuté pod označení „R2 manažer“.

Základní soubory pro práci s R2 jsou:

Soubor	Funkce
r2run.bat	Spustí úplnou sekvenci, zahrnující všechny dále uvedené dávky.
loadset.bat <sada>	Nahraje do tabulky sdílených dat určenou datovou sadu.
m1run.bat	Spustí modul M1.
m2run.bat	Spustí modul M2.
m3run.bat	Spustí modul M3.
getresults.bat	Zkopíruje výsledky ze sdílené tabulky zpět do tabulek ve schématu test.

Soubory jsou umístěny v repository ve složce manager, kde je také soubor README.MD s podrobnějšími informacemi.

Popis a nastavení modulu M1

Instalace

Modul M1 je instalován a oživen společně v rámci software R2 - knihovna programů pro řízení dílčích procesů automatizované tvorby topografických map.

Struktura modulu

Modul se skládá z několika částí, které jsou uloženy v adresáři m1 v samostatných podadresářích:

- manager – obsahuje skripty zajišťující řízení vlastního procesu generalizace
- misc – obsahuje různé doplňující součásti (např. vizualizační projekt v QGIS)
- model – obsahuje šablonu konfigurace modulu, odkud se distribuuje do příslušného datového balíku v adresáři data a nástroje pro vývoj (nepoužívá se ke generalizaci)
- postgresql – obsahuje skripty potřebné k nastavení prostředí (databáze), k založení a naplnění příslušných tabulek a prostorové funkce sloužící k měření, manipulaci a testování prostorových dat

Na úrovni podadresářů jsou uloženy skripty pro spouštění vybraných operací (např. skript pro obnovu dat ve schématu m1 do výchozího stavu či skript pro spouštění testů).

Další skripty, konfigurace, nástroje a utility jsou uloženy v části, která je společná všem modulům:

- admin – obsahuje administrační nástroje a utility
- data – obsahuje adresářovou strukturu pro jednotlivé testovací datové sady; adresář datové sada obsahuje skripty pro spouštění jednotlivých operací (importu modelu m1, importu dat do schématu m1, promítnutí provedených změn), šablonu konfigurace a konfiguraci modelu m1
- manager – obsahuje skripty zajišťující proces generalizace v rámci software R2 řízením modulů M1, M2 a M3 včetně logování

Technologie

Modul M1 se opírá o tyto technologie:

- postgresql – relační databázový systém pro efektivní správu dat
- postgis – prostorové rozšíření postgresql systému nabízející funkce pro snadnou manipulaci s geometriemi
- plpgsql – procedurální jazyk nad postgresql databází, který umožňuje vytvářet vlastní funkce a dotazy
- python – dynamický programovací jazyk, jehož funkcionalitu je možné výrazně rozšířit dostupnými knihovnami

Popis fungování modulu

Modul M1 je navržen jako specifický modul pro řešení generalizace metodou odsunu prvků v izolovaných a méně komplexních generalizačních situacích. Pro realizaci je navržen a vyvinut komplexní operátor odsunu. Modul M1 pracuje s obecně definovanými entitami a vztahy mezi nimi. Ty jsou vyjádřeny v modelu m1, který definovanou sémantikou popisuje kartografický model včetně footprintů signatur jednotlivých entit, chování entit a vztahy mezi entitami. Proces generalizace je řízen parametry, které jsou definovány konfiguračně v XML souboru (podrobněji viz kapitola Konfigurace).

Proces generalizace sestává z několika dílčích kroků:

- **pre-processing** – načtení modelu m1, načtení dat z data.elements do modelu m1, zohlednění topologických a netopologických elementů, vytvoření footprintů
- **processing** – vlastní generalizace – rekurzivní procházení řetězem konfliktních elementů a řešení konfliktů odsunem jejich průběhu na distanční linii, vyhodnocení provedených elementárních odsunů proti definovaným podmínkám a omezením (constraints)
- **post-processing** – rekonstrukce topologických elementů, promítnutí změn do data.elements včetně naplnění stavových atributů

Konfigurace

Modul M1 je začleněn do software R2 - knihovna programů pro řízení dílčích procesů automatizované tvorby topografických map. Z něho přebírá parametry společné s ostatními moduly. Tato kapitola se zabývá specifickými nastaveními modulu M1.

Model m1 je popsán konfiguračně v souboru m1-model.xml a je vázán na šablonu genmodel.xsd. Doporučená struktura konfigurace je vystavěna z těchto částí:

- obecné a společné vlastnosti a vztahy přebírané dle znalostní databáze kartografa
- specifické vlastnosti a vztahy přebírané dle znalostní databáze kartografa
- vlastnosti přebírané z tabulky FC_properties.xls listu ZM10_elt (pro model ZM10)

Základní entitou modelu m1 je *element*. Jedná se o prvek kartografického modelu jednoznačně definovaných charakteristik a chování. Jednotlivé elementy jsou výskytem entity *element typ*, přičemž každý *element typ* má v rámci celého kartografického modelu jedinečné charakteristiky, chování a vztahy k ostatním entitám.

Každý *element typ* je popsán:

- jednoznačným identifikátorem *id*
- geometrickým typem *geomType*
- vahou (důležitostí) *weight*
- barvou signatury (v případě složené symboliky barvou obrysové linie) *color*
- původem (např. v případě vstupního modelu v Esri Geodatabase organizací prvku v rámci struktury Feature Data Set a Feature Class)
- definicí footprintu signatury (*Buffer, Geometry*)
- příznakem, zda má topologické vztahy vůči dalším entitám *topology*
- příslušností k sémantickým kategoriím kartografického modelu *category*.

Chování *element typu* je popsáno:

- výčtem operátorů, které lze na element *element typu* aplikovat *Operation*.

Model je doplněn popisem vztahů mezi jednotlivými entitami:

- mezi elementy téhož *element typu*
- mezi elementy dvojice *element typů*.

Jde především o vyjádření:

- zda jsou relačně svázané entity při překryvu signatur v konfliktu *conflict*
- jaká je požadovaná světlost (rozestup, mezera) mezi signaturami entit *clearance*.

Obecné (generické) vlastnosti entit a vztahy mezi entitami jsou přepisovány specifickými vlastnostmi a vztahy popsány v konfiguraci modelu. V konfiguraci modelu mohou být vlastnosti a vztahy nastaveny hromadně pro skupiny element typů, v implementaci modelu jsou vlastnosti přiřazeny jednotlivým element typům a vztahy dvojicím element typů.

K popisu modelu m1 v souladu s šablonou slouží následující syntaktické elementy:

- *ElementTypeList* – uvozuje a ukončuje oddíl popisující vlastnosti *element typů*
- *ElementType* – uvozuje a ukončuje popis vlastností *element typu*
- *TagDict* – slovník tagů (Tag Dictionary), kde *Tag* je tvořen klíčovým slovem *key* a hodnotou *value*
- *TagList* – seznam tagů, kdy tag je tvořen textovým řetězcem
- *Footprint* – uvozuje a ukončuje definici footprintu signatury (*Buffer, Geometry*)
- *Buffer* – definuje footprint signatury *element typu* obálkou s parametry šířky obálky *radius*, typu zakončení *cap*, *capStart*, *capEnd*, zkrácení *crop* a náhradu nesymetrického bufferu odsunutým symetrickým bufferem *offset*
- *Geometry* – definuje footprint signatury *element typu* geometrií ve formátu WKT *Wkt* a vztahným bodem *Origin* s parametry *x* a *y*
- *SelectorList* – uvozuje a ukončuje oddíl popisující vlastnosti a vztahy entit pomocí selektorů na základě výběrových podmínek
- *Selector* – selektor představuje rámec výběrové podmínky *cond* nebo *join*, pro který jsou definovány vlastnosti a vztahy entit (není-li podmínka v selektoru vyjádřena, platí definované vlastnosti a vztahy pro všechny entity)
- *OperationDict* – slovník operací (Operation Dictionary), kde operace *Operation* je tvořena klíčovým slovem *key* (operátor) a hodnotou *value* (určující, zda je operátor v daném kontextu povolen)
- *RelationList* – uvozuje a ukončuje oddíl popisující vztahy entit pomocí relačních selektorů na základě výběrových podmínek
- *RelationSelector* – relační selektor představuje rámec výběrové podmínky *cond* nebo *join*, pro který jsou definovány vlastnosti a vztahy vůči selektoru *Selector*
- *Relation* – definice vztahu (*clearance, conflict*) mezi entitami omezenými společným rámcem *Selector* a *RelationSelector*
- *DisplacementList* – uvozuje a ukončuje seznam specifických (nesymetrických) nastavení odsunu *Displacement*
- *Displacement* – definice specifického (nesymetrického) nastavení odsunu s parametry stranové orientace řídicího elementu *selector*, stranové orientace závislého elementu *relation*, vzdálenosti distanční linie (v tomto případě ekvidistanty) od geometrie řídicího elementu *distance* a limitu přichycení k distanční linii *snap*.

Všechny hodnoty parametrů velikosti či vzdálenosti jsou vyjádřeny reálnou mírou v terénu v jednotkách metry [m] ve zvoleném rovinném souřadnicovém systému a z hodnot vztažených k mapě jsou přepočteny referenčním měřítkem kartografického modelu (mapy).

Příklady konfigurací

Následující příklady částí konfigurace popisují:

- obecné a společné vlastnosti a vztahy přebírané dle znalostní databáze kartografa
- specifické vlastnosti a vztahy přebírané dle znalostní databáze kartografa
- vlastnosti přebírané z tabulky FC_properties.xls listu ZM10_elt (pro model ZM10).

Poznámka: Část konfigurace modelu z tabulky FC_properties.xls se pro vybrané *element typy* získává exportem (copy paste) sloupce *ElementType_XML* listu ZM10_elt s nastavením filtru na sloupci *Constraint* vylučující hodnoty *duplicita*, *nejasnost*, *nezobrazuje se*. Pro export je nezbytné použít nastavení desetinného oddělovače v nastavení prostředí na „.“. Po exportu se v editoru XML nahradí znak uvozovek

prázdným znakem. Sloupce Category1, Category2 a Category3 jsou ve výchozím nastavení listu tabulky skryté. Při exportu je možné použít filtry i na další sloupce.

Příklad č. 1: Nastavení výchozích (generických) vlastností a vztahů všech element typů resp. skupin element typů

- geometrický typ *geomType*
- váha *weight*
- topologie *topology*
- povolené operátory *Operation*
- světlost footprintů signatur *clearance*

Zápis v konfiguraci:

```
<Selector>
  <ElementType weight="10" geomType='line' topology='false'>
    <OperationDict>
      <Operation key="displacement" value="false" />
    </OperationDict>
  </ElementType>
  <RelationList>
    <RelationSelector>
      <Relation conflict="true" />
    </RelationSelector>
    <RelationSelector join="selector.tagdict->'color' =
relation.tagdict->'color'">
      <Relation clearance="2" />
    </RelationSelector>
  </RelationList>
</Selector>
```

Příklad č. 2: Řešení s použitím/bez použití topologie

- použití topologie – nastavením parametru *topology* na true (do topologie budou zahrnuty element typy označené tagem topology (<Tag>topology</Tag>))
- bez použití topologie - nastavením parametru *topology* na false

Zápis v konfiguraci:

```
<Selector cond="taglist ? 'topology'">
  <ElementType topology='true' />
  <RelationList>
    <RelationSelector cond="taglist ? 'topology'">
      <Relation conflict="false" />
    </RelationSelector>
  </RelationList>
</Selector>
<Selector cond="topology = 'true'">
  <ElementType>
    <OperationDict>
      <Operation key="displacement" value="false" />
    </OperationDict>
  </ElementType>
</Selector>
```

Příklad č. 3: Nastavení obecných vlastností a vztahů pro skupiny entit

- vyjmutí bodových prvků z množiny konfliktních prvků

Zápis v konfiguraci:

```
<Selector cond="geomtype = 'point'">
  <ElementType>
    <OperationDict>
      <Operation key="displacement" value="false" />
    </OperationDict>
  </ElementType>
  <RelationList>
    <RelationSelector>
      <Relation conflict="false" />
    </RelationSelector>
  </RelationList>
</Selector>
```

Příklad č. 4: Převzetí obecných vlastností prvku železnice dle znalostní databáze kartografa

 **Železnice**

[Znalosti kartografa](#) | [Mapové produkty](#) | [ZM 1:10 000](#) | [Komunikace](#) | [Liniové komunikace](#) | [Železnice](#) [Zaregistruj diskuzi](#)

- Železnice ve stavbě se vyjadřují příslušnou značkou, která je v daném úseku přerušovaná (207)

Vlastnost	Hodnota	Zděděno od	Typ
Category	line	Liniové komunikace	Category
Weight	100		Integer
Color	black	Komunikace	Color
Symbols	201, 202, 203, 204, 205		List
Třída přesnosti	B	Liniové komunikace	Precision
Minimální velikost	40 m		Area
Operátory	odsun, vypuštění, paralelizace, nahrazení sdruženou značkou, zjednodušení průběhu,		List





10.05.2016 07:32:54

Zápis v konfiguraci:

```
<!-- vlastnosti přebírané dle znalostní databáze kartografa -->
<SelectorList>
  <Selector cond="tagdict->'category1' = 'Komunikace' AND tagdict->'category2' = 'Liniové komunikace' AND tagdict->'category3' = 'Železnice'">
    <ElementType weight="100">
      <TagDict>
        <Tag key="color" value="black" />
      </TagDict>
    </ElementType>
  </Selector>
</SelectorList>
```

Příslušnost jednotlivých *element typů* k sémantickým kategoriím kartografického modelu *category1*, *category2* a *category3* je čerpána z tabulky FC_properties.xls, jak je ukázáno níže.

Poznámka: Vlastnost Weight má ve znalostní databázi kartografa pro prvky s největší vahou přiřazenu nejnižší hodnotu. Implementace modulu M1 naopak předpokládá přiřazení nejvyšší hodnoty pro prvky s nejvyšší vahou. Hodnota *weight* je pro účely konfigurace vypočtena odečtením hodnoty Weight ve znalostní databázi kartografa od hodnoty 1000. Ve výše uvedeném případě je ponechána původní hodnota z důvodu vizuální shody.

Příklad č. 5: Převzetí vlastnosti prvku železnice z tabulky FC_properties.xls listu ZM10_elt (pro model ZM10)

DATA10 - FDS	DATA10 - FC	Prvek	Elt_id	Category1	Category2	Category3	Footprint	Radius	Offset	LineCap	Crog	Wkt	Topologie
Z_KomZelezTrat	Z_KomZelezTrat_L	žel. norm. rozchodná neel 1kol.	L2010001	Komunikace	Liniové komunikace	Železnice	Buffer	0.35		flat			ANO
Z_KomZelezTrat	Z_KomZelezTrat_L	žel. norm. rozchodná neel 1kol.	L2010002	Komunikace	Liniové komunikace	Železnice	Buffer	0.35		flat			ANO
Z_KomZelezTrat	Z_KomZelezTrat_L	žel. norm. rozchodná neel vícekol.	L2020001	Komunikace	Liniové komunikace	Železnice	Buffer	0.35		flat			ANO
Z_KomZelezTrat	Z_KomZelezTrat_L	žel. norm. rozchodná neel vícekol.	L2020002	Komunikace	Liniové komunikace	Železnice	Buffer	0.35		flat			ANO
Z_KomZelezTrat	Z_KomZelezTrat_L	žel. norm. rozchodná el 1kol.	L2030001	Komunikace	Liniové komunikace	Železnice	Buffer	0.35		flat			ANO
Z_KomZelezTrat	Z_KomZelezTrat_L	žel. norm. rozchodná el 1kol.	L2030002	Komunikace	Liniové komunikace	Železnice	Buffer	0.35		flat			ANO
Z_KomZelezTrat	Z_KomZelezTrat_L	žel. norm. rozchodná el vícekol.	L2040001	Komunikace	Liniové komunikace	Železnice	Buffer	0.35		flat			ANO
Z_KomZelezTrat	Z_KomZelezTrat_L	žel. norm. rozchodná el vícekol.	L2040002	Komunikace	Liniové komunikace	Železnice	Buffer	0.35		flat			ANO
Z_KomZelezTrat	Z_KomZelezTrat_L	žel. úzkorozchodná	L2050000	Komunikace	Liniové komunikace	Železnice	Buffer	0.25		flat			ANO

Zápis v konfiguraci:

```

<!-- žel. norm. rozchodná neel 1kol. -->
<ElementType id='L2010001' geomType='line'>
  <TagDict>
    <Tag key='category1' value='Komunikace' />
    <Tag key='category2' value='Liniové komunikace' />
    <Tag key='category3' value='Železnice' />
  </TagDict>
  <TagList>
    <Tag>Z_KomZelezTrat</Tag>
    <Tag>Z_KomZelezTrat_L</Tag>
    <Tag>topology</Tag>
  </TagList>
  <Footprint>
    <Buffer radius='3.5' cap='flat' />
  </Footprint>
</ElementType>

```

Příklad č. 6: Nastavení specifického vztahu mezi terénním stupněm a stromořadím**Stromořadí, úzký pruh lesa**

Znalosti kartografa | Mapové produkty | ZM 1:10 000 | Porost a využití půdy | Aleje, ploty, průseky | **Stromořadí, úzký pruh lesa**

[Zaregistruj](#)
[diskuzi](#)

- stromořadí a úzký pruh lesa je nezobrazuje v lesní půdě (406), křovinatém porostu (408), kosodřevině(409) a v okrasné zahradě (415)
- **stromořadí maskuje příčné čárky terénního stupně**
- musí být zobrazeny alespoň dvě značky

Vlastnost	Hodnota	Zděděno od	Typ
Category	line	Aleje, ploty, průseky	Category
Weight	900		Integer
Color	black	Aleje, ploty, průseky	Color
Symbols	412		List
Třída přesnosti	C	Porost a využití půdy	Precision
Minimální velikost	100 m		Length
Operátory	odsun, vypuštění, vypuštění části		List



10.05.2016 07:32:54

Zápis v konfiguraci:

```

<!-- specifické vlastnosti a vztahy přebírané dle znalostní databáze kartografa -->
<Selector cond="id = 'L4120000'">
  <RelationList>
    <RelationSelector cond="id = 'L6060100'">
      <Relation>
        <Footprint>
          <Buffer radius="0.65" cap="flat" />
        </Footprint>
      </Relation>
    </RelationSelector>
  </RelationList>
</Selector>

```

Příklad č. 7: Nastavení specifického vztahu mezi terénním stupněm a železnici



Terénní stupeň

[Znalosti kartografa](#) | [Mapové produkty](#) | [ZM 1:10 000](#) | [Terénní reliéf](#) | [Terénní reliéf - liniové prvky](#) | [Terénní stupeň](#) | [Zaregistruj diskuzi](#)

- při kolizi dvou souběžných terénních stupňů hranami k sobě je průběh nahrazen značkou 314a (vzdálenost hran do 10m), nebo značkou 314b (vzdálenost hran nad 10m)
- terénní stupně ve tvaru jámy do šířky 15m jsou nahrazeny bodovou značkou 607
- příčné čárky terénního stupně mohou být zkráceny, vymaskovány nebo vypuštěny, nesmí však být narušena linka (horní hrana)
- minimální vzdálenost terénního stupně od ostatních linií je 0,2 mm
- pokud je vzdálenost menší než 0,2 mm nebo při kolizi hranou k prvku slícovává se s hranicí užívání, železnici, dvoučarou komunikací, budovou a parkovou cestou, v opačném případě se paralelizuje
- pokud je vzdálenost menší než 0,2 mm nebo při kolizi s ostatními liniovými prvky se s nimi paralelizuje
- při kolizi s budovou se terénní stupeň odstraní
- při kolizi příčných čárek a budov jsou příčné čárky maskovány budovou
- v místech kolize příčných čárek s jiným terénním stupněm je ponechána pouze základna (příčné čárky se vypustí)
- v místech generalizace je terénní stupeň druhým prvkem (po stromořadí), který se vypouští
- při kolizi terénního stupně s objekty na komunikaci (propustek, most, lávka apod.), terénní stupeň se přeruší nebo zkrátí
- terénní stupeň se zaměřenou dolní hranou na komunikaci z obou stran mostu, nezkracují se jeho příčné čárky
- pokud terénní stupeň navazuje na rokli, stupeň se zkrátí a rokli se nalícuje dle potřeby
- pokud terénní stupeň se zaměřenou dolní hranou navazuje na obyčejný terénní stupeň, délka příčných dílků plynule přechází na základní délku
- vzájemná poloha terénních stupňů a ostatních prvků se zachovává

Zápis v konfiguraci:

```
<Selector cond="id = 'L6060100'">
  <RelationList>
    <RelationSelector cond="taglist ? 'Z_KomZelezTrat'">
      <Relation>
        <DisplacementList>
          <Displacement selector="right" relation="*" distance="12.5"
snap="4" />
          <Displacement selector="left" relation="*" distance="2.95"
snap="4" />
        </DisplacementList>
      </Relation>
    </RelationSelector>
  </RelationList>
</Selector>
```

Popis a nastavení modulu M2

Instalace

Modul M2 je instalován a oživen společně v rámci software R2 - knihovna programů pro řízení dílčích procesů automatizované tvorby topografických map.

Struktura modulu

Modul se skládá ze 4 hlavních součástí, každá se nachází v jednom adresáři:

- admin - obsahuje skripty potřebné k nastavení prostředí (databáze) a vstupní tabulky s pravidly generalizace
- docs - dokumentace postupů a hodnot použitých při samotné generalizaci.
- loader - samotný kód generalizace a její konfigurace.
- sql_scripts - prostorové funkce sloužící k měření, manipulaci a testování prostorových dat - geometrií.

Technologie

Modul M2 se opírá o tyto technologie:

- postgresql - relační databázový systém pro efektivní správu dat
- postgis - prostorové rozšíření postgresql systému nabízející funkce pro snadnou manipulaci s geometriemi
- plpgsql - procedurální jazyk nad postgresql databází, který umožňuje vytvářet vlastní funkce a dotazy
- python - dynamický programovací jazyk, jehož funkcionalitu je možné výrazně rozšířit dostupnými knihovnami

Popis fungování modulu

Inspirace/idea modulu M2 je tzv. útok hrubou silou. Principem takového postupu je snaha, provádět vždy trochu pozměněnou operaci tak dlouho, až bude řešení validní. Pro optimalizaci takového postupu jsou v modulu definované modifikovatelné tabulky:

- rules (pravidla) - definují akce, které se mají vykonávat
- contexts (kontexty) - určují podmínky, za kterých se "rule" aplikuje
- maps (mapy) - definují vizuální podobu prvků v daném měřítku/mapě
- constraints (omezení) - vycházejí z "maps" - vizuální podoby prvků a určují možnosti při bezkonfliktním umísťování geometrií.

Samotný proces generalizace je potom rozdělen na 4 části:

- pre-processing - v této části se hledají topologické vazby mezi prvky, zkoumá se struktura a řeší imperceptibilita
- clustering - úlohou tohoto kroku je, najít v datech všechny kolize a na jejich základě vytvořit skupiny geometrií, které spolu interagují a proto se budou řešit současně
- processing - hlavní část generalizace, která v rámci iterací postupně projde všechny nalezené clustery/shluky/skupiny geometrií a na základě tabulek rules, contexts, maps a constraints definuje operátory, které na tyto geometrie postupně aplikuje
- post-processing - ukládání změn, zápis úspěšnosti použitých operátorů do statistických tabulek.

Konfigurace

Proces generalizace obsahuje několik možností parametrizace. Některé modifikátory jsou definované v souboru loader/config/modifiers.json:

- `touches_limit` - definuje maximální vzdálenost mezi dvěma geometriemi, při které můžeme uvažovat o jejich vzájemném doteku (hodnota byla vytvořena kvůli nekonzistenci některých prvků). Parametr se udává v metrech a neměl by dosahovat velké hodnoty. Motivace existence tohoto parametru je například v případě nedotahu linií v rámci říční sítě. V případě neexistence takového jevu je možné použít 0.
- `cluster_range_limit` - vzdálenost, v rámci které bude vyhledávaná kolize ještě zařazená do aktuálního clusteru/shluku. Při navýšení tohoto čísla bude během generalizace vytvořený menší počet shluků, tím se může zabránit identifikaci některých konfliktů. Doporučená hodnota je zhruba 100 - 500, při přechodu na generalizaci menšího měřítka by se měla hodnota zvyšovat.
- `selection_iterations` - kolikrát se operace pokusí aplikovat pravidlo na každou geometrii ve výběru. Vyšší hodnota by měla zlepšit kvalitu výsledků na úkor času, modul byl testovaný s hodnotami 5 - 100.
- `processing_iterations` - počet opakování kroků "clustering" a "processing". Hodnota při testování se pohybovala na úrovni 2-10.
- `origin_table_path` - název zdrojové tabulky s geometriemi. Parametr by se měl měnit pouze při specifických nastaveních databáze a vstupu dat.
- `restore_elements` - v případě hodnoty 1 bude před začátkem procesu tabulka elementů obnovena do původního stavu (výstup obnovený do stavu vstupu) - vhodné pro testovací účely, pro standardní běh generalizace se doporučuje zachovat 0.
- `log_operations` - hodnota 1 umožňuje zapisovat vstupy a výstupy jednotlivých operací do složky loader/logs/operations - vhodné pro testovací účely, pro standardní běh generalizace se doporučuje zachovat 0.

Další možností konfigurace chodu modulu je využití tabulek popsaných v části Popis fungování modulu:

- `rules` (pravidla) - obsahuje kombinace operací, které budou aplikované v případě, že podmínky kontextu budou naplněné. Operátor odkazuje na konkrétní sql funkci, která bude postupně pouštěná na geometrii v konfliktu s parametry definovanými hodnotami ve sloupcích `minpar`, `maxpar` a `step`.
- `contexts` (kontexty) - mají za úlohu popisovat kritické situace na mapě. Tabulka obsahuje označení geometrie, které se tato situace týká a označení geometrie referenční (ke které se měří vztah). Samotný vztah je možné definovat pomocí operátoru a dané hodnoty (například vzdálenost geometrie od referenční je menší než 50m)
- `maps` (mapy) - tabulka obsahuje grafické omezení na mapě daného měřítka. Popisuje vždy označení daného prvku a pravidla pro něj určené symboliky. Na základě zvolené symbolizace se potom automaticky vyplňuje tabulka `constraints` - ta definuje vzájemné konflikty prvků na mapě.
- `thesaurus` (omezení) - výčet definovaných termínů pro označení prvků a skupin prvků, které je možné používat napříč ostatními tabulkami (`rules`, `maps`, `contexts`).

Tyto tabulky je možné najít ve složce *m2/admin/input_tables* ve formátu .csv. Po jejich editaci je třeba spustit proces *m2/admin/m2-import_structure.bat* (případně *m2/admin/m2-build.bat*), který tyto hodnoty naimportuje. Je důležité zachovávat validní formu tohoto souboru (csv se středníky jako oddělovači a hodnotami ve správném formátu), v opačném případě se proces importu zasekne. Při dalším puštění generalizace se tyto pravidla aplikují. Detailnější dokumentace jednotlivých sloupců a jejich domén je uložena v *m2/admin/input_tables_docs.md*.

Parametrizace procesních dávek

U operací, které jsou součástí předzpracování a mají jednoznačné řešení, jsou, v této verzi modulu, jejich parametry nastaveny json souborem uloženým v *m2/loader/config/modifiers.json*. Struktura záznamu parametru je následující:

```
"imperceptibility_min_building": {
  "value": 40,
  "description": "minimal area of a building"
}
```

Hodnotu lze tedy měnit v parametru "value". Pro řízení nepatrnosti mají význam následující klíče, které předpokládají hodnotu v m nebo m²:

- *imperceptibility_max_pit* - limit délky uzavřeného stupně pro kolizi na bod
- *imperceptibility_max_lot* - limit délky stromořadí pro odstranění
- *imperceptibility_max_terrain* - limit délky terénního stupně pro odstranění
- *imperceptibility_max_building* - limit velikosti solitérní budovy pro typifikaci
- *imperceptibility_max_area* - limit velikosti ploch užití pro sloučení
- *imperceptibility_max_forest* - limit velikosti lesní plochy pro odstranění

V souboru se nacházejí i interní klíče, jejich funkce je uvedena v položce "description".

Parametrizace pravidel

Pro uživatele jsou zpřístupněny tabulky *rules*, *contexts*, *maps* a *thesaurus*. Nacházejí se v adresáři modulu *m2/admin/input-tables*. Tabulky se nahrají do systému při inicializaci modulu. Jsou ve formátu .csv v kódování UTF-8 s oddělovačem ";" (";" se používá v definovaných hodnotách) a se strukturou tabulky v prvním řádku.

Záznam tabulky *rules* je jednosložkový a adresuje prvek, na který má být aplikován jednoduchý generalizační algoritmus:

id	jedinečné id pravidla
origin	pro uživatele pouze hodnota "user"
map	id mapy z tabulky maps
ffeature	typ prvku, který je generalizován
context	odkaz na situaci prvku podmiňující akci
operator	identifikátor algoritmu
minpar	minimální hodnota parametru v m
maxpar	maximální hodnota parametru v m
step	krok v m
priority	priorita pravidla pro prvek (1-99)

Na pravidlo je navázáno 1..n kontextových informací definujících podmínky za kterých se aplikuje. Tyto záznamy jsou uvedeny v tabulce *contexts*:

rule	id pravidla
origin	pro uživatele pouze hodnota "user"
ffeature	cíl pravidla
rfeature	prvek ve vztahu, self, any
relation	typ vztahu
predicate	srovnání s hodnotou
value	hodnota v m nebo m ² , booleovská hodnota

Pro možnost pracovat v pravidlech a kontextech s obecnějšími termíny (např. "nadcesta" je agregovaný termín pro "lávku" a "nadchod") je nutné tyto termíny definovat v tabulce *thesaurus*:

term	agregovaný výraz
part	část agregace z maps nebo thesauru
level	úroveň agregace (mimo jiné slouží jako vodítko, kde hledat část)

Základní kategorie a parametry kresby se nacházejí v tabulce *maps*:

pid	id mapy
name	název mapy
fid	id kategorie
feature	název kategorie jak je použit v pravidlech
fspec	doplňková informace o kategorii
type	typ kresby - point, polygon, simple_line, compound_line, marker_chain
color	styková barva kresby
signature	šířka tahu v mm
marker_size	velikost značky v mm
marker_offset	pozice značky vůči vodícímu prvku
marker_step	mezera mezi sérií značek
priority	priorita kategorie
displacement_limit	explicitní omezení posunu

Základní omezení jsou automaticky odvozena z přepočtu parametrů v tabulce *maps*.

Příklady konfigurací

Na následujícím sejmutí obrazovky je příklad nastavení souboru *m2/loader/config/modifiers.json*. Význam jednotlivých položek je uveden v description.

```
"imperceptibility_max_pit": {
  "value": 200,
  "description": "limit délky uzavřeného stupně pro kolizi na bod (metry)"
},
"imperceptibility_max_lot": {
  "value": 100,
  "description": "limit délky stromořadí pro odstranění (metry)"
},
"imperceptibility_max_terrain": {
  "value": 50,
  "description": "limit délky terénního stupně pro odstranění (metry)"
},
"imperceptibility_max_area": {
  "value": 625,
  "description": "limit velikosti ploch užití pro sloučení (metry čtvereční). neproběhne, pokud je hodnota scale menší než 25000"
},
"imperceptibility_max_forest": {
  "value": 1000,
  "description": "limit velikosti lesní plochy pro odstranění (metry čtvereční). neproběhne, pokud je hodnota scale menší než 25000"
},
"imperceptibility_max_building": {
  "value": 50,
  "description": "limit velikosti soliterní budovy pro typifikaci (metry čtvereční)"
},
}
```

Následující příklad je definicí pravidla pro terénní prvky. Nejprve je třeba přidat řádky do souboru *thesaurus.csv* definující agregovaný výraz "sráz" pro vybrané terénní prvky.

```
sráz;jáma;0
sráz;terénní stupeň;0
sráz;násep;0
sráz;zářez;0
sráz;srázný břeh;0
```

Nula na konci znamená, že se jedná o elementární agregát.

Následně se definuje řádek v souboru *rules.csv*.

```
2;user;1;sráz;2;gn_parallelisation;0;15;1;25
```

V tomto případě se jedná o uživatelské pravidlo číslo 2, příslušející mapě číslo 1, které se aplikuje na kategorie agregované termínem "sráz" a je podmíněno prostorovým kontextem číslo 2. Řešení je provedeno algoritmem *gn_parallelisation* s počátečním parametrem 0 m a koncovým parametrem 15 m s krokem po 1 metru. V případě kolize pravidel má toto pravidlo prioritu 25.

Na závěr je potřeba definovat prostorový kontext, kdy je pravidlo aplikováno vložení zvolených podmínek do *contexts.csv*.

```
2;user;sráz;any;PARALLEL;=;1
2;user;sráz;self;LENGTH;<;500
2;user;sráz;any;DISTANCE_CLOSEST;<;20
```

Kontext s pořadovým číslem 2 obsahuje následující podmínky - je v relaci PARALLEL s libovolným geometrickým objektem, je menší než 500 m a paralelní objekt je blíže než 20 m.

Popis a nastavení modulu M3

Instalace

Modul M3 je instalován a oživen společně v rámci software R2 - knihovna programů pro řízení dílčích procesů automatizované tvorby topografických map.

Struktura modulu

Modul se skládá z několika částí, které jsou uloženy v adresáři m3 v samostatných podadresářích:

- m3 – specifická funkcionální modularita modulu m3 a nastavení pro generalizaci ZM
- constraints - definice dekorátorů a implementací jednotlivých constraints
- dataconnectors – obsahuje skripty zajišťující řízení vlastního procesu generalizace
- generalization – algoritmy a rutiny generalizace
- mapgraphic - definice mapových elementů
- model - knihovna pro modelování vztahů mezi entitami, definici objektového modelu a další
- sharedtools - knihovna sdílených funkcí pro logování a práci s nastaveními
- workflow - knihovna pro vytváření workflow pro umožnění vzdálené správy

V hlavním adresáři jsou uloženy skripty pro spuštění vybraných operací:

- preprocessing.py – workflow připravená pro spuštění v rámci software R2 před jednotlivými moduly M1, M2 a M3
- m3.py – skript modulu M3, spouštěný v rámci dávky software R2
- manager.py - skript umožňující vzdálenou správu modulu M3

Technologie

Modul M3 se opírá převážně o tyto technologie:

- postgresql - relační databázový systém pro efektivní správu dat
- postgis - prostorové rozšíření postgresql systému nabízející funkce pro snadnou manipulaci s geometriemi
- python - dynamický programovací jazyk, jehož funkcionální rozšíření je možné výrazně rozšířit dostupnými knihovnami

Popis fungování modulu

Modul M3 je navržen jako specifický modul pro řešení generalizace metodou postupného umísťování prvků do kresby mapy a řešení definovaných konfliktů, které v postupně zaplňované mapě vznikají. Postup je v tomto podobný klasickému sestavení mapy. Pořadí elementů jsou dány jejich důležitostí při řešení případných konfliktů mezi sebou, vycházejí z hodnoty váhy prvku stanovené v metodice Nmet2. Nejedná se tedy o vizuální pořadí vykreslování prvků jako při výsledném tisku kartografického obrazu mapy.

Proces generalizace sestává z několika dílčích kroků:

- **pre-processing** – načtení modelu m3, načtení dat z data.elements do modelu m3, přiřazení priorit vykreslení a barvy okraje jednotlivých symbolů
- processing – vlastní generalizace – postupné umísťování prvků do mapy a řešení konfliktů constraints pro ně definovaných (constraints)
- **post-processing** – promítnutí změn do společné tabulky data.elements včetně naplnění stavových atributů

Konfigurace

Modul M3 je začleněn do software R2 - knihovna programů pro řízení dílčích procesů automatizované tvorby topografických map. Z něho přebírá parametry společné s ostatními moduly. Tato kapitola se zabývá specifickými nastaveními modulu M3.

Model m3 je popsán konfiguračně v souboru *config.json* a je vázán na šablonu.

Definice datového modelu R2

Modul M3 pracuje v rámci software R2 pomocí výměnného datového modelu, který definuje vstupní tabulku dat, ze které jsou čerpány elementy ke zpracování včetně jejich stavu. Do stejné tabulky jsou elementy poté vráceny. Definice této tabulky je napevno umístěna ve skriptu *m3\megatable.py*.

Definice mapového díla

Mapové dílo je pro účely modulu M3 definováno jeho značkovým klíčem a dalšími parametry, které stanovují požadavky na sestavení a generalizaci mapy. Definice ZM je umístěna v adresáři *M3\mapseries* a jeho podadresářích.

Barvy prvků

Barva prvků ve skriptu *colors.py* je pro potřeby modulu M3 omezena na identifikátor barvy a jeho symbolické hodnoty.

```
class ZMColors:
    GRAY = Color("Tmavě šedá", "#707070", "ZM10_tmaveseda")
    BLUE = Color("Tmavě modrá", "#0061ca")
    GREEN = Color("Zelená", "#00ff04")
    VIOLET = Color("Fialová", "#784999")
    YELLOW = Color("Žlutá", "#ffff700", "ZM10_zluta")
```

Symbologie mapy

Zjednodušená symbologie kresby jednotlivých mapových značek je definována v souboru nastavení *symbology.py*.

Měřítko mapy

Měřítko generované mapy je definováno globální proměnnou

```
model.generalization.Generalization.scale = MapScale(10000)
```

Barva a síla okraje značky

Většina okrajů značek v Základní mapě je provedena v tmavě šedé barvě o tloušťce 0.13 mm, totéž platí i pro obrysy (lemovky) ploch. Příkazem *buildEmptySymbols* nastavíme pro všechny značky v *data.elements* tyto parametry. V tomto příkazu definujeme barvu pomocí identifikátoru barvy definovaného výše. Hodnota 0.13 je samozřejmě v milimetrech v měřítku mapy a je přepočtena dynamicky pomocí nastavení měřítka.

```
zm10Symbolization.buildEmptySymbols(ZMColors.GRAY, 0.13)
```

Přiřazení barvy okraje seznamu značek

Pro značky v jiných barvách okraje použijeme jednoduchý příkaz *setOutlineColor* k jejich "přebarvení".

```
zm10Symbolization.setOutlineColor(ZMColors.BLUE, ["P3330000", "L3020100", "L3020102", "L3030000", "L3030002", "L3040000", "L3040001", "L3040002"])
```

Přiřazení symetrické signatury značky

Mapové dílo jeví systémovou skladbu síly signatur liniových značek, například hlavní silniční síť je vyvedena v síle čáry 1.5 mm a jednotlivé typy komunikací jsou odlišeny

výplní, respektive středovou čarou. Obdobně lze snadno definovat jednotlivé kategorie kartomapového díla.

Dálnice, rychlostní silnice, silnice I. třídy:

```
zm10Symbolization.setWidth(1.50, ["L2400000", "L2410000", "L2420000"])
```

Silnice II a III. třídy, neevidovaná silnice:

```
zm10Symbolization.setWidth(1.20, ["L2430000", "L2440000", "L2450000"])
```

Přiřazení nesymetrického symbolu

```
zm10Symbolization.symbols["L6060100"].footPrintBuilder = LinearSymbolFootprint(0.7, 0.065)
```

Pravidla pro umisťování a generalizaci mapových značek

buildconstraints.py

Nepřekrývající se symboly

```
symbolSet.addConstraintsToSymbols(ClearanceConstraint(), [])
```

Rozestup mezi barvami

```
ClearanceConstraint.COLOR_TRESHOLDS.add(ZMColors.BLUE, ZMColors.BLUE, 0.5, 0.5)  
ClearanceConstraint.COLOR_TRESHOLDS.add(ZMColors.BLUE, ZMColors.GRAY, 0.0, 0.3)
```

Požadovaná přesnost vykreslení prvku

Základní přesnost mapy:

```
symbolSet.addConstraintsToSymbols(MapSymbolPositionConstraint(0.20, 0.8), [])
```

Prvky s nižším požadavkem přesnosti:

```
symbolSet.addConstraintsToSymbols(MapSymbolPositionConstraint(0.60, 1.3), ["L6060100", "L6100000",  
"L6250000"...])
```

Nejméně důležitá přesnost:

```
symbolSet.addConstraintsToSymbols(MapSymbolPositionConstraint(0.60, 0.8), ["P4010100", "P4020100",  
"P4020200", "P4020300", ...])
```

Přiřazení dalších pravidel

Další pravidla je možné k vybraným skupinám značek definovat pomocí příkazu `symbolSet.addConstraintsToSymbols`, stejně jako v předešlých příkladech. Jména constraints a případné parametry jsou definovány v příslušných dokumentačních řetězcích (docstrings) knihovny *m3\constraints*.

Slovník pravidel

Slovník pravidel používaný modulem je poté kompilovanou jednostupňovou databází, kde každému symbolu odpovídá jeden klíč ve slovníku, jak je patrné z následujícího obrázku. Je zřejmé, že v případě potřeby je možné tento slovník generovat přímo z dostupných digitálních informací, zejména ontologické databáze z metodiky Nmet2.

```
mapDefinition = {  
  "symbols": [  
    {  
      "outlineColorValue": "#707070",  
      "constraintIds": [  
        66794760,  
        66795720,  
        66795912  
      ],  
      "name": "4230200",  
      "outlineColorName": "Tmavě šedá",  
      "footPrintBuilderId": "66794312",  
      "footPrintBuilderTitle": "Obrys plochy 1.30&nbsp;",  
      "identifier": "P4230200",  
      "type": "Plocha"  
    },  
    {  
      "outlineColorValue": "#707070",  
      "constraintIds": [  
        66794760,  
        66795720,  
        66795912  
      ],  
      "name": "4200600",  
      "outlineColorName": "Tmavě šedá",  
      "footPrintBuilderId": "66777352",  
      "footPrintBuilderTitle": "Obrys plochy 1.30&nbsp;",  
      "identifier": "P4200600"  
    }  
  ]  
}
```