Západočeská univerzita v Plzni
Fakulta aplikovaných věd
Katedra informatiky a výpočetní techniky

Diplomová práce

ETL nástroj pro konverzi OpenStreetMap dat do datové struktury nástroje TrafficModeller

Plzeň 2022 Vaněk Jakub

Místo této strany bude zadání práce.

Prohlášení

Prohlašuji, že jsem diplomovou práci vypracoval samostatně a výhradně s použitím citovaných pramenů. V knize jsou použity názvy programových produktů firem apod, které mohou být ochrannými známkami nebo registrovanými ochrannými známkami příslušných vlastníků.

V Plzni dne 11. března 2022

Vaněk Jakub

Poděkování

Placeholder - k nalezení v souboru thesiskiv.cls

Abstract

Configuration of Software Components Generator from Feature Models. Goal of this thesis is to generate feature model from grammar of tesa language written in Xtext and to generate final source code from chosen features of this model. This thesis describes and uses knowledge about feature modeling and generative programming. In the opening part of the thesis reader is introduced with feature modeling, tools used for feature modeling, generative programming and Xtext framework. In the later part implementation design and final solution is described.

Abstrakt

Cílem této práce bude implementovat nástroj vykonávající obousměrný konverzní algoritmus mezi daty s datovou strukturou OpenStreetMap a nástrojem TrafficModeller. V úvodní části bude představen proces ETL, popsány datové struktury OpenStreetMap a nástroje TraMod. Dále bude popsána implementace a výsledné řešení.

Obsah

1 Úvod

Diplomová práce se bude zabývat sběrem a přípravou dat pro nástroj TrafficModeller, který byl vyvinut na Západočeské Univerzitě v Plzni. Tento nástroj umožňuje pomocí svého API jednoduše a rychle testovat různé scénáře dopravy na specifikovaném území.

K naplnění datové struktury tohoto nástroje budou použita data z OpenStreetMap (dále jen OSM), která jsou volně dostupná široké veřejnosti. K naplnění datové struktury nástroje TrafficModeller bude potřeba získat data z OSM, poté je transformovat do příslušné datové struktury nástroje TrafficModeller a následně je do něj nahrát.

V úvodní části této práce bude popsán proces ETL. Dále budou popsány přístupy k harmonizaci dat se zaměřením na geodata. Následně bude popsána datová struktura dat z OSM a budou detekovány části relevantní pro nástroj TrafficModeller.

V další části bude navržen a implementován nástroj vykonávající konverzní algoritmus, který data převede z datové struktury OSM do datové struktury TraMod. Součástí nástroje bude také zpětná konverze dat do datové struktury OSM obohacených o výstupy nástroje TraMod. Nástroj bude otestován a bude detailně okomentován.

V závěru práce budou popsány a kriticky zhodnoceny dosažené výsledky.

2 ETL

V současném obchodním světě se různé společnosti potýkají s roustoucím množstvím sbíraných a uchovávaných dat a s potřebou těmto datům co nejlépe porozumět. Tato data mohou být ve společnostech používána k omptimalizaci firemních procesů, sledování účinnosti firemních strategií, objevování nových nevyužitých příležitostí a mnohému dalšímu. Využití dat k rozhodovacím procesům ve společnosti nazýváme *Business intelligence*. Business intelligence chápeme jako soubor technologií a procesů, které uživatelům umožňují přístup k datům a analýzu dat za účelem podpory rozhodování [Howson].

Jednou ze základních komponent BI je tzv. *datový sklad*. William Inmon datový sklad definuje takto: "*A data warehouse is a subject-oriented, integrated, nonvolatile and time-variant collection of data in support of management's decision*". Jedná se o zvláštní typ databáze, která je používána pro datové analýzy. Inmon ve své definici používá 4 důležité charakteristiky takové databáze:

- *subject-oriented* - orientovaný na subjekt - Datový sklad je orientovaný na subjekt, protože poskytuje informace o konkrétním subjektu namísto probíhajících operací organizace. Těmito subjekty mohou být zákanící, dodavatelé, prodej, výnosy apod. Datový sklad se nezaměřuje na probíhající operace, ale zaměřuje se na modelování a analýzu dat za účelem rozhodování.

- *integrated* - integrovaný - Ze všech charakteristik je právě tato tou nejdůležitější. Data jsou do datového skladu integrována z více různých zdrojů, které mohou mít rozdílnou strukturu, názvosloví, jednotky apod.. Taková data tedy musí být očištěna a transformována tak, aby se do datového skladu nahrála v jedné konzistentní podobě a byla tak umožněna jejich analýza.

- *non-volatile* - stálost - Tato charakteristika značí skutečnost, že data v datovém skladu jsou neměnná. Do datového skladu data pouze nahráváme a následně k nim přistupujeme, nikdy je však neaktualizujeme. Data jsou do datového skladu nahrávána v podobě statického záznamu, který reflektuje stav v daném čase. Pokud se tedy takový stav změní, není v databázi aktualizován, ale je nahrán nový záznam opět reflektující stav v novém čase a starý záznam je pro analytické účely v datovém skladu zachován.

- *time-variant* - časová variabilita - Data jsou do datového skladu nahrána tak, že reflektují stav v přesně daném čase. Tyto stavy jsou tedy v datovém skladu zachovány a tím je umožněno získat přesný stav systému v jakémkoliv okamžiku.

K analýze dat je tedy nejdříve třeba vytvořit datový sklad. Datový sklad je tvořen procesem zvaným *ETL*. Zkratka ETL reprezentuje populární třífázový proces, při kterém jsou data z jednoho či více heterogenních zdrojů nahrána do datového skladu. Těmito třemi fázemi jsou fáze extrakce (z angl. *extraction*), transformace (z angl. *transform*) a nahrání dat (z angl. *load*). Běžným označením pro prostředky ETL je rovněž datová pumpa. V následujících částech budou popsány jednotlivé fáze procesu ETL.

2.1 Extrakce dat

Prvním procesem, který je používán při výstavbě datového skladbu je proces zvaný *extrakce*. Poté co určíme cíl datového skladu a stanovíme jeho strukturu, je třeba identifikovat zdrojové systémy, ze kterých budou data do datového skladu extrahována. Tyto zdrojové systémy se od sebe navzájem liší. Mohou se lišit například ve své struktuře, či formátu uložení. Běžnými formáty, ve kterých jsou data uložena mohou být například relační databáze, formát XML či JSON, ale může se jednat o jakékoliv jiné systémy pro uložení dat.

Klíčovým požadavkem v této fázi je, aby byla všechna data ze zdrojových systémů nahrána v požadovaném čase. S tím se pojí hned několik problémů. Zdrojové systémy mohou být například dočasně nedostupné. Může se také stát, že zdrojový systém není uzpůsoben k tako rozsáhlé extrakci dat a požadovaná zátěž pro něj může být z různých důvodů nepřípustná a je třeba hledat náhradní řešení (zálohy systému, čtení pouze části dat apod.). Dalším problémem, na který je možné při extrakci dat narazit může být například příliš velký objem dat, kdy u těchto procesů není výjimkou objem dat v řádu několika GB denně.

Extrakce probíhá ve dvou fázích [https://www.cs.colostate.edu/etl/papers/Thesis.pdf] . V první fázi probíhá tzv. úplná extrakce. Při této fázi jsou data extrahována poprvé a je tedy nutné je extrahovat kompletně celá. Druhá fáze je tzv. inkrementální. Tato fáze nastává ve chvíli, kdy se ve zdrojových systémech objeví nová nebo modifikovaná data. Nová nebo modifikovaná data je třeba identifikovat a odlišit od takových, které již procesem extrakce prošla dříve [Kimball]. Identifikace nových dat může být problematická. Můžeme k ní využít tři přístupy.

- Logy v databázi - V této technice mohou být použity logy DBMS. Tyto logy jsou použity pro nalezení přidání nebo změny dat ve zdrojové databázi.
- Triggery - Na každé tabulce ve zdrojové databázi jsou vytvořeny triggery, které jsou automaticky spuštěny při přidání či změně dat ve zdrojové databázi pomocí DML (Data Manipulation Language).
- Časová razítka - Některé databáze používají sloupce pro časová razítka, která specifikují čas ve kterém byl daný řádek naposledy modifikován. Pomocí těchto sloupců lze jednoduše identifikovat změnu ve zdrojovém systému.

Pokud však zdrojovým systémem není relační databáze, není možné takové přístupy použít. Je tedy třeba manuálně nalézt způsob, jak identifkovat přidaná či změněná data a ty následně extrahovat.

Cílem této fáze je tedy identifikovat relevantní informace ve zdrojových systémech a nahrát je do jediné struktury či formátu, která bude vhodná pro fázi transformace.

2.2 Transformace dat

2.3 Nahrání dat

2.4 Konkrétní případ ETL

3 Konverze Dat

V této části budou popsány jednotlivé struktury a způsob konverze mezi nimi.

3.1 Datové struktury

Každý nástroj využívá svou vlastní strukturu dat. Tyto struktury jsou vytvořeny pro účely jednotlivých nástrojů. Zdrojovými daty budou data ve formátu OpenStreetMap. Tato data bude potřeba transformovat do struktury dat cílového nástroje Traffic Modeller.

3.1.1 Traffic Modeller

Nástroj Traffic Modeller je serverové řešení pro modelování a simulaci různých dopravních situací téměř v reálném čase. K takovému účelu využívá geografická data týkající se silnic. Cílem práce je taková data připravit a nahrát do nástroje.

Data tohoto nástroje jsou uložena v databázi, která obsahuje několik tabulek.

- *edge* - tabulka hran
- *node* - tabulka vrcholů
- *odm* - origin-destinatiom matrix
- *zones* - zóny reprezentující generátory dopravy
- *turn_restriction* - tabulka zákazů odbočení

Tabulka vrcholů v nástroji označuje veškeré křižovatky. Na každém místě, kde se komunikace fyzicky kříží musí existovat v datech nástroje Traffic Modeller jeden vrchol. Vrchol obsahuje svůj identifikátor a také hodnotu geometrie, která v sobě nese geografické údaje o umístění vrcholu. Vrchol také existuje v bodě konce vozovky (např. na konci slepé ulice).

Tabulka hran obsahuje všechna spojení mezi vrcholy. Každá komunikace od křižovatky ke křižovatce je zaznamenána jednou hranou. Každá hrana obsahuje svůj identifikátor, zdrojový vrchol, cílový vrchol a geometrii. Tvoří tedy jednosměrnou orientovanou hranu. Pokud mezi dvěma vrcholy A a B

existuje obousměrná komunikace, musí v tabulce existovat dva záznamy s unikátními identifikátory. První záznam bude mít uvedený vrchol A jako zdrojový a vrchol B jako cílový a druhý záznam bude mít uvedený vrchol B jako zdrojový a vrchol A jako cílový. Pokud je silnice mezi vrcholy $A.B$ jednosměrná, existuje pouze jedna hrana, která obsahuje vrchol A jako zdrojový a B jako cílový. Hodnota geometrie určuje polohu a tvar křivky, která příslušnou komunikaci reprezentuje. Pokud vozovka obsahuje více pruhů, je stále reprezentována pouze jednou hranou.

Tabulka zákazů odbočení (turn_restriction) obsahuje záznamy pro všechny zákazy odbočení na mapě. Obsahuje identifikátor křižovatky (vrcholu), ke kterému náleží, identifikátor zdrojové hrany a identifikátor cílové hrany. Pokud tedy nelze z komunikace (hrany) A na křižovatce (vrcholu) X odbočit na komunikaci B, bude v tabulce uveden záznam s identifikátorem vrcholu X, zdrojovou hranou A a cílovou hranou B. Pokud je na na jedné křižovatce zákazů více, bude pro každý zákaz v tabulce jeden záznam. Nástroj Traffic Modeller neobsahuje způsob pro uchování příkazů odbočení. Pokud se na nějaké křižovatce vyskytuje přikázané odbočení, budou zakázány všechny ostatní možnosti odbočení a tyto zákazy budou zaneseny do tabulky.

Jak již první dvě tabulky databáze napovídají, silnice jsou v nástroji Traffic Modeller reprezentovány grafovou strukturou. Vrcholy tohoto grafu označují křižovatky . Taková reprezentace umožňuje nástroji modelovat dopravu pomocí algoritmů užívaných pro různé grafové problémy.

Tabulka zón reprezentuje tzv. generátory dopravy. Jsou to vymezené oblasti na mapě, ze kterých je na základě budov odhadnut počet vozidel denně vyjíždějící z této oblasti. TODO

3.1.2 OpenStreetMap

OpenStreetMap je volně dostupná databáze s geografickými daty. Tato data poskytují jeho uživatelé, kteří mají možnost je libovolně upravovat a obohacovat. Projekt se takto podobá portálu wikipedie. Data mohou být užívána k libovolným účelům za podmínky uvedení autorství OpenStreetMap.

https://alga.win.tue.nl/tutorials/openstreetmap/

Data v OSM jsou uložena v jednoduché datové struktuře, která obsahuje nodes, ways a relations. Nodes reprezentují jednotlivé body na mapě. Tyto body obsahují souřadnice a unikátní identifikátor. *Way*, nebo také cesta, reprezentují křivky nebo uzavřené polygony na mapě. Záznamy o těchto cestách neobsahují žádnou lokaci, místo toho obsahují sled jednotlivých bodů (nodes). Ralations (spoje) mohou být chápany jako skupiny cest, či bodů, které mají nějaký společný význam. Tato data jsou tedy reprezentována

grafem.

Body (nodes) tedy mají v datové struktuře dvojí funkci. Mohou reprezentovat jednobodové elementy. Mohou reprezentovat například různé typy budov, sjezd z dálnice a mnoho dalšího. V druhém případě mohou být umíštěny jako součást cesty (way) k určení tvaru křivky či polygonu. Současně mohou plnit obě funkce. Například bod může být součástí křivky reprezentující silnici a současně reprezentovat sjezd z dálnice.

Cesty (ways) reprezentují nejednobodové elementy. Reprezentují veškeré křivky a uzavřené polygony. V datové struktuře se mezi křivkou a polygonem rozlišuje pouze použitím prvního vrcholu současně jako posledního, čímž se křivka uzavře a vznikne tedy polygon, který vymezuje. Křivky nereprezentují pouze silnice, ale také například železnice, řeky, elektrické vedení a mnoho dalšího. Polygony mohou reprezentovat plochy, jako jsou například lesy, vodní plochy a další.

K určení elementů, které cesty a jednotlivé body reprezentují se využívají tzv. značky (tags). Značky je možné přiřadit k jednotlivým bodům i celým cestám. Značky obsahují klíč a hodnotu. Klíč je užíván k definování názvu objektu a hodnota k definování jeho hodnoty. Některé značky hodnotu nepotřebují, užívá se potom klíčové slovo *yes*.

Příklady značek elementů:

- jednobodové - *shop=supermarket*
- neuzavřené cesty (křivky) - *highway=motorway*
- uzavřené cesty (polygony) - *building=yes*

Portál openstreetmap.org umožňuje zobrazit značky jednotlivých elementů. Značky jednotlivých elementů jsou uživateli zobrazeny po kliknutí na element pomocí funkce *Průzkum prvků*.

Krom bodů a cest mohou být v OSM také zaznamenány spoje (původně *Relations*). Tyto spoje jsou pouze sledem cest, bodů nebo jiných elementů a mohou mít své vlastní značky. Příkladem může být například dálnice, která bude mít své vlastní značení a atributy a bude spojovat několik menších cest, které jsou součástí této dálnice. Jiným příkladem může být například celá trasa nějaké linky městské hromadné dopravy, která bude opět spojena jednotlivými úseky, které jsou v OSM zaneseny jako jednotlivé menší cesty.

K využití těchto dat v nástroji Traffic Modeller je důležité prozkoumat způsob, jakým OSM reprezentuje data týkající se pozemních komunikací. Data v OSM jsou reprezentována následujícím způsobem. Každá pozemní komunikace je reprezentována jednou nebo více cestami (mohou být součástí

většího spoje). Pokud se dvě komunikace fyzicky kříží tak, že je možné dostat se z jedné na druhou, existuje v místě jejich překřížení vrchol. Pokud se však cesty fyzicky nekříží, například z důvodu existence mostu, který je druhou cestou podjížděn, v bodě překřížení těchto hran vrchol neexistuje. Pokud má pozemní komunikace více pruhů, které nejsou odděleny žádnou fyzickou bariérou, je v datech zanesena jako jedna cesta. Pokud mezi sebou jednotlivé pruhy mají fyzickou bariéru (dálnice), jsou v OSM zaneseny jako dvě jednotlivé cesty.

Dále je ke správnému fungování důležité určit, jakým směrem cesty vedou. Cesty jsou reprezentovány sledem dvěma a více bodů. Pokud existuje komunikace mezi vrcholy A a B, bude existovat cesta *{A,B}* nebo cesta *{B,A}*, ale nikoliv obě. Takovou cestu můžeme v grafu poté chápat jako neorientovanou hranu mezi dvěma vrcholi. Pokud je však komunikace jednosměrná, cesta bude označena značkou *oneway=yes*. Takové označení využívají veškeré jednosměrné komunikace, včetně obousměrných komunikací oddělených fyzickou bariérou. Pokud cesta obsahuje tuto značku, její směr je poté dán pořadím uzlů, které cestu tvoří.

PLACEHOLDER: popis budov, jejich relations apod.

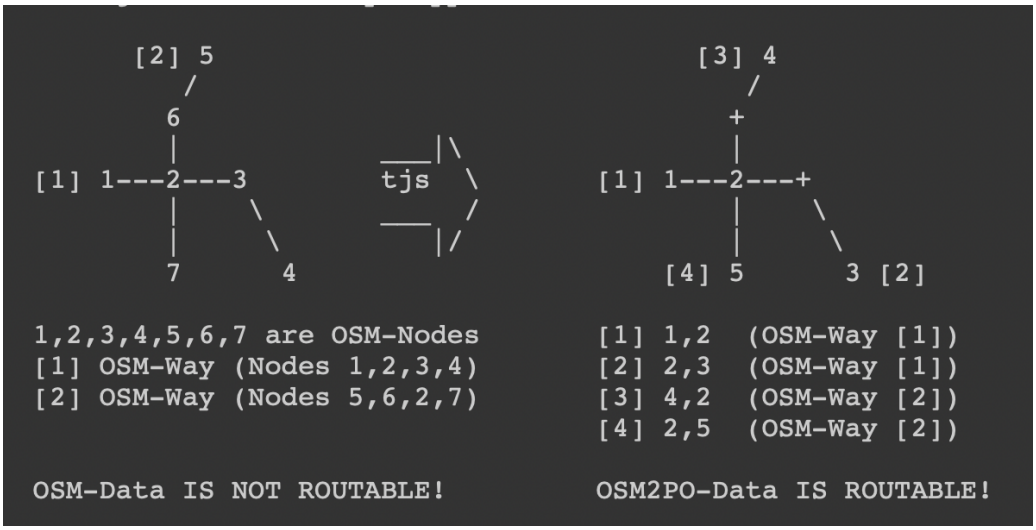
3.1.3 Odlišnosti

Z předchozích částí je jasné, že se od sebe struktury dat databáze OSM a nástroje Traffic Modeller navzájem liší. Tyto rozdíly je nutné identifikovat a určit, jakým způsobem musí být data z databáze OSM transformována tak, aby vyhovovala nástoji Traffic Modeller.

Silniční síť

Základním nedostatkem Open Street Map je, že silniční síť v těchto datech není směrovatelná (z angl. *routable*). Tento nedostatek je znázorněn na příkladě na obrázku:

Na obrázku je vidět 7 bodů: *{1, 2, 3, 4, 5, 6, 7}*. Tyto body reprezentují body na silnici. V datech jsou také zaneseny dvě cesty. Cesta *A: {1, 2, 3, 4}* a cesta *B: {5, 6, 2, 7}*. Takováto data tedy mohou reprezentovat jednoduchou křižovatku. Díky první cestě například víme, jakým způsobem se dostat z bodu *{1}* do bodů *{2, 3}* nebo *{4}*. Problém vzniká ve chvíli, kdy bychom se chtěli z bodu *{1}* dostat například do bodu *{7}*. Je zřejmé, že do bodu *{7}* je možńe se dostat přes bod *{2}* a cesta by tedy byla složená z bodů *{1, 2, 7}*. Bohužel však v datech Open Street Map taková cesta být nemusí a v datech tedy neexistuje způsob, jakým se z bodu *{1}* dostat do bodu *{7}*. Taková síť je tedy nesměrovatelná.



Základním předpokladem nístroje Traffic Modeller je však směrovatelná síť. Tedy síť, ve které bude možné najít cestu z libovolného zdrojového bodu, do libovolného cílového bodu. Pro tyto účely je možné využít nástroj *OSM2Po*.

OSM2Po je volně dostupný nástroj. Tento nástroj je současně konvertorem a směrovacím enginem. Dokáže parsovat data z OSM a vytvořit z nich směrovatelnou síť. Jeho výstupem jsou sql soubory pro databázi PostGIS, které společně reprezentují graf. Těmito soubory jsou:

- uzly - v souboru *<název_území>_2po_vertex.sql*
- hrany - v souboru *<název_území>_2po_4pgr.sql*

Každý z těchto souborů tvoří příslušnou tabulku v databázi.
Tabulka uzlů:

- *id* - identifikátor uzlu
- *clazz* -
- *osm_id* - původní id elementu v OSM
- *osm_name* -
- *ref_count* -
- *restrictions* - zákazy, či příkazy odbočení
- *GeometryColumn* - kód geometrie elementu

Záznamy v tabulce uzlů reprezentují veškerá místa, kde se vozovka fyzicky kříží či končí. Reprezentují tedy veškeré křižovatky, sjezdy z dálnic, konce slepých ulic a další. Ve výstupním vygenerovaném souboru se vyskytuje méně záznamů o uzlech, než ve zdrojovém souboru ve formátu OSM. Prvním důvodem je vybírání pouze některých elementů. V OSM mohou jednotlivé body značit krom křižovatek i budovy, zastávky a jiné jednobodové elementy na mapě, zatímco tento nástroj vyexportuje pouze body, které se týkají silniční sítě. Druhým důvodem je způsob reprezentace hran. OSM reprezentuje hrany pouze jako sled několika bodů. Tvar této hrany (křivky) je tedy určen body, ze kterých se skládá. Výstup tohoto souboru však tvar hrany určuje pomocí samostatného pole s gemoetrií elementu a nevyužívá bodů k určení tvarů. Pokud se tedy žádné vozovky v bodě nekříží, bod v místě neexistuje.

Důležitou vlastností, kterou je možné najít v každém záznamu o uzlu je záznam o tzv. *Restrictions*, neboli zákazech. Tento záznam popisuje zákazy či příkazy odbočení, které se na této křižovatce vyskytují. Díky tomuto záznamu bude možné jednoduše naplnit tabulku *Turn_Restrictions* v cílovém nástroji Traffic Modeller.

Tabulka hran:

- *id* - identifikátor hrany
- *osm_id* - původní identifikátor elementu v OSM
- *osm_name*
- *osm_meta* -
- *osm_source_id* - původní identifikátor zdrojového uzlu hrany z OSM
- *osm_target_id* - původní identifikátor cílového uzlu hrany z OSM
- *clazz* - třída komunikace
- *flags*
- *source* - identifikátor zdrojového uzlu hrany odpovídající záznamům z tabulky vrcholů
- *target* - identifikátor cílového uzlu hrany odpovídající záznamům z tabulky vrcholů
- *km* - délka úseku v km
- *kmh* - maximální povolená rychlost na daným úseku

- *cost* - cena za přejetí tohoto úseku počítána jako kmkmh
- *reverse_cost* - cena za přejetí tohoto úseku v opačném směru
- *x1*
- *x2*
- *y1*
- *y2*
- *GeometryColumn* - kód geometrie elementu

Záznamy v tabulce hran reprezentují veškeré úseky na vozovce od jednoho bodu k druhému. Každý záznam o úseku z vozovky je definován svým identifikátorem, zdrojovým bodem a cílovým bodem. V seznamu jsou uvedeny další informace, které v sobě záznam uchovává. Důležitou informací je směr daného úseku. Pokud je vozovka jednosměrná, vede od zdrojového k cílovému uzlu, hodnota *cost* určuje čas potřebný k přejetí daného úseku a hodnota *reverse_cost* je nastavena na *1000000.0*. Pokud je vozovka obousměrná, má nastavené obě hodnoty *cost* i *reverse_cost*, kde hodnota cost značí potřebný čas k přejetí úseku od zdrojového k cílovému bodu a hodnota *reverse_cost* značí čas potřebný k přejetí vozovky od cílového bodu ke zdrojovému. Tvar křivky reprezentující vozovku není narozdíl od OSM reprezentován několika body, avšak samostatnou hodnotou geometrie.

4 Závěr

Literatura