



Konwerter danych OSM na wejście programu Kraksim

Dokumentacja procesowa i deweloperska

Przedmiot:

Praktyki letnie 2016

Prowadzący:

dr inż. Jarosław Koźlak

Rok akademicki:

2015/2016

Zespół:

Daniel Ogiela

Krzysztof Węgrzyński

Spis treści

1 Dziennik projektu.....	2
2 Obserwacje dotyczące napotkanych problemów.....	6
3 Realizacja harmonogramu.....	10
4 Rozliczenie realizacji.....	11
4.1 Funkcjonalność programu.....	11
4.2 Realizacja modułów.....	11
5 Spis najistotniejszych problemów napotkanych w czasie realizacji projektu i sugestie ulepszeń. .	12
6 Uruchomienie programu.....	12
7 Repozytorium projektu.....	13

1 Dziennik projektu

04.07.2016

- uruchomienie systemu
- przegląd dokumentacji użytkownika
- częściowy przegląd dokumentacji programisty
- ustalenie tematu i wstępne określenie zadań do wykonania
 - konwersja różnych formatów map na format wejściowy systemu
 - wykonanie testów
 - praca nad modulem predykcji (bliżej nieokreślone wymagania)

05.07.2016

- przegląd dokumentacji programisty
- dokładne zaznajomienie się z formatem danych wejściowych do systemu
- uruchomienie przez wszystkich członków zespołu systemu
- zbadanie oferowanych usług przez różne serwisy mapowe
 - OpenStreetMap
 - Google API
 - Targeo API

- pobieżne zaznajomienie się z API OSM

06.07.2016

- przegląd dokumentacji Overpass API
- testowanie kwerend Overpass API
- przygotowanie query w Overpass API do pobrania interesujących parametrów dróg
- rozwiązanie problemu wykrywania skrzyżowań
- przegląd dokumentacji python-overpy
- rozwiązanie problemu parsowania pliku wyjściowego pobranego od Overpass API
- ustalenie sposobu implementacji wykrywania skrzyżowań
- konwersja geograficznych współrzędnych punktów na współrzędne systemowe oraz przeliczenie odległości na jednostki systemowe

07.07.2016

- stworzenie pseudokodu algorytmu konwersji
 - algorytm ConvertToBlock1InnerFormat(resultObject) - konwersja danych do struktury wewnętrznej pozwalającej na stworzenie bloku 1 danych wejściowych
 - algorytm ConvertToBlock2InnerFormat(resultObject) - konwersja danych do struktury wewnętrznej pozwalającej na stworzenie bloku 2 danych wejściowych
 - algorytm ConvertToBlock3InnerFormat(resultObject) - konwersja danych do struktury wewnętrznej pozwalającej na stworzenie bloku 3 danych wejściowych

08.07.2016

- stworzenie dokumentu projektowego, w szczególności harmonogramu

11.07.2016

- stworzenie diagramu klas
- implementacja wczytywania koordynat i wykonania kwerendy

12.07.2016

- założenie repozytorium
- implementacja zapisywania skrzyżowań i bram do struktury wewnętrznej - algorytm konwersji nr 1
 - brak odległości(x i y dla node'ów), trzeba ustalić punkt odniesienia
 - trzeba usunąć time.sleep - implementacja własnego odpytywania o brakujące node'y lub optymalizacja query (w tym momencie nie wiemy jak)

13.07.2016

- dokończenie implementacji zapisywania skrzyżowań i bram do struktury wewnętrznej - algorytm konwersji nr 1
 - wyznaczanie współrzędnych i punktu odniesienia (lewy dolny róg wybranego regionu)
 - optymalizacja kwerendy (zaciągane od razu wszystkie potrzebne node'y)
- implementacja zapisywania skrzyżowań i bram do struktury wewnętrznej - algorytm konwersji nr 2
 - dodanie pola oneway do klasy Way w celu możliwości oznaczenia dróg jednokierunkowych

14.07.2016

- implementacja zapisywania skrzyżowań i bram do struktury wewnętrznej - algorytm konwersji nr 3

15.07.2016

- częściowa implementacja algorytmu konwersji struktury wewnętrznej (blok 1) do pliku xml
- sporządzenie raportu z pierwszych dwóch tygodni praktyk

18.07.2016

- dodanie flagi flag_from do wyeliminowania błędu niepoprawnego formatu danych z OSM
- dokończenie implementacji algorytmu konwersji struktury wewnętrznej (blok 1) do pliku xml

19.07.2016

- implementacja algorytmu konwersji struktury wewnętrznej (blok 2) do pliku xml
- rozwiązanie problemów z kodowaniem przy zapisywaniu do pliku xml, słowem kluczowym 'from', który utrudniał zapisanie do pliku xml słowa 'from' oraz z kolejnością wpisywanych atrybutów do drzewa xml

20.07.2016

- implementacja algorytmu konwersji struktury wewnętrznej (blok 3) do pliku xml
- implementacja generowania konfiguracji do generatora ruchu z domyślnymi parametrami
- kraksim nie przyjmuje pliku wejściowego, debugowanie

21.07.2016

- naprawienie błędnego wpisywania numerów pasów oraz id węzłów do pliku xml - blok nr 3
- naprawienie błędnego wpisywania akcji dla ulic jednokierunkowych

22.07.2016

- znalezienie błędu w formacie pliku wejściowego generowanego przez nasz program
- znalezienie potencjalnego błędu nadmiernej liczby reguł nadrzędności. Przypuszczenie, że reguła nadrzędności dla danej akcji może wystąpić tylko wtedy, gdy dotyczy drogi, która w zbiorze swoich akcji ma jedną, której exit jest taki sam jak badanej wyjściowej akcji

25.07.2016

- poprawienie obydwu usterek znalezionych 16.09 - brak określenia pierwszeństwa dla "armów" o tym samym "exicie" i nadmierna liczba reguł nadrzędnych
- założenie, że dla dróg o jednakowym priorytecie pierwszeństwo ustalane jest w sposób losowy
- **wyświetlenie pierwszej mapy**

26.07.2016

- wyświetlanie mapy bez ingerowania w kod kraksima
 - odległości pomnożone przez stałą 20 w celu wygenerowania odpowiednio dużej mapy (kraksim odrzuca małe odległości między skrzyżowaniami z przyczyn nam nieznanych)
 - poprawianie wykrywania bram i skrzyżowań - ostateczne założenie, że co najmniej dwie a nie trzy drogi krzyżujące się tworzą skrzyżowanie
 - zapewnienie dwukierunkowości wszystkich dróg prowadzących lub wychodzących z bram (wbrew rzeczywistym danym - kraksim wymusza takie założenie)

27.07.2016

- refaktoryzacja kodu
- poprawienie wczytywania mapy poprzez poprawienie współrzędnych granicznych badanego obszaru
- uruchomienie symulacji dla ręcznie stworzonego pliku generacyjnego
- poprawienie orientacji mapy

28.07.2016

- refaktoryzacja kodu
- poprawa wyświetlenia mapy:
 - usunięcie pojedynczych bram
 - usunięcie krótkich dróg

2 Obserwacje dotyczące napotkanych problemów

04.07.2016

- System nie uruchamia się na systemach linuksowych:
 - pojawia się wyjątek FileNotFoundException
- Błędy w dokumentacji:
 - literówki
 - zapis 1GB MB

- literówka w kodzie - Discretiser -> Discretizer

05.07.2016

- nie ma realistycznego odwzorowania pomiędzy skrzyżowaniami w Kraksimie i rzeczywistymi skrzyżowaniami w Krakowie. (dane wejściowe są niedokładne)
- OpenStreetMap oferuje większość parametrów istotnych dla danych wejściowych do systemu - ronda, skrzyżowania, liczbę pasów, porządki pierwszeństw
- OpenStreetMap nie oferuje jedynie informacji o organizacji ruchu na drogach oraz informacji o działaniu sygnalizacji świetlnej
- według pobieżnego sprawdzenia API Google nie oferuje interesujących nas parametrów mapy
- API Targeo nie oferuje ogólnodostępnych map (potrzebny wysłany wniosek o udostępnienie danych)
- mapy wyeksportowane z OpenStreetMap są dużego rozmiaru
- większość skrzyżowań powinno być zaznaczone - część skrzyżowań trzeba by było zanaczać oddzielnie przez sprawdzenie czy odpowiednie drogi się przecinają
- należy sprawdzić jak w OSM oznaczane są skrzyżowaniem z drogami równorzędnymi
- należy ograniczyć ilość informacji zaciąganych do pliku wyjściowego
- Overpass API pozwala na wyciągnięcie interesujących nas cech z mapy (SQL bazy danych) i tym samym ograniczenie rozmiaru pliku
- przystanki autobusowe generują dużą ilość danych (zaciągają rozkład jazdy autobusów)

06.07.2016

- podporządkowanie typów dróg na OSM(od najważniejszych):
 - motorway
 - trunk
 - primary
 - secondary
 - tertiary
 - unclassified/residential/service(zakładamy ich równość)

- wykrywanie skrzyżowań, pomocne linki:
 - <http://wiki.openstreetmap.org/wiki/Pl:Junctions>
 - <http://wiki.openstreetmap.org/wiki/Pl:Key:junction>
- zakładamy uproszczenia (być może tymczasowe)
 - pomiędzy każdymi dwoma skrzyżowaniami jest punkt wlotowy i wylotowy dla samochodów (fioletowy prostokąt w symulacji kraksima)
 - z racji braku dokładnych danych na temat dróg (np. jakie są znaki, organizacja ruchu) będziemy w uproszczony sposób generować zasady ruchu na skrzyżowaniach
 - głównie będzie się opierać na poziome "ważności" dróg
 - czy na skrzyżowaniu są światła (uruchomienie zarządzania skrzyżowaniem przez algorytm)
 - w.p.p wszystkie światła na skrzyżowaniu są *zielone*

07.07.2016

- dla dróg równorzędnych założenie, że moduł sterowania ruchem kraksima obsługuje ruch

11.07.2016

- overpy API działa poprawnie (możliwe uzyskanie dowolnych tagów)

12.07.2016

- punkt odniesienia jest wyznaczany niezależnie dla różnych danych, w przykładowych plikach wejściowych współrzędne są dodatnie co wskazuje na fakt umieszczenia mapy w dodatniej ćwiartce układu współrzędnych

13.07.2016

- program nie ma stałego punktu odniesienia, lecz dla każdego punktu odniesienia wyznaczany niezależnie
- do przyspieszenia działania programu wystarczyła optymalizacja kwerendy - dzięki niej zaciągane są wszystkie potrzebne node'y od razu
- w drogach jednokierunkowych droga przebiega w kierunku node[0] -> node[-1], gdzie w pliku node[0] jest u samej góry node'ów danej drogi, a node[-1] na samym dole

14.07.2016

- Overpass API pobiera część danych, które są poza określonym obszarem

18.07.2016

- podstawowy typ danych OSM - relacja - nie zawsze zawiera oczekiwane informacje. Interesujący nas typ relacji restrykcji nie zawsze zawiera wszystkie pola "from", "via" i "to"

20.07.2016

- przy tworzeniu konfiguracji generatora ruchu zakładamy pewne domyślne parametry, w późniejszych fazach rozwoju projektu można umożliwić zmianę tych parametrów przez GUI
- arm, exit, entrance są albo gatewayem albo intersection
 - arm jest początkowym punktem drogi(skrzyżowanie lub brama, z których dojeżdżamy)
 - exit jest ostatnim punktem drogi, w którą wjeżdżamy(skrzyżowanie lub brama, do których jedziemy)
 - entrance jest początkowym punktem drogi(skrzyżowanie lub brama, z których dojeżdżamy)

21.07.2016

- zauważona nieścisłość w dokumentacji programisty przy opisie formatu pliku wejściowego. W bloku nr 3 w gałęzi "action", jak i gałęzi "rule" pole "lane" ma różne znaczenie w zależności od wartości tego pola. Jeśli pole ma wartość mniejszą od 0, to jest to pas do skrętu w lewo. Jeśli pole ma wartość większą od 0, to jest to pas do skrętu w prawo. W przeciwnym przypadku jest to zwykły pas
- node "intersection" w trzecim bloku pliku wejściowego musi być inny zarówno od węzła "arm", jak i od węzła "exit"

22.07.2016

- blok nr 3 pliku wejściowego potrzebuje informacje o tym, która z dwóch dróg ma pierwszeństwo na skrzyżowaniu przy wyjeździe ze skrzyżowania na tą samą drogę. Informacja musi być obecna dla każdej takiej sytuacji

25.07.2016

- kraksim nie przyjmuje “gateway’ów”, które kończą bądź zaczynają się drogą jednokierunkową
- problem z odnalezieniem pliku z logami kraksima - dotychczas nie znaleziono
- kraksim nie odpala się ze względu na wyjątek spowodowany niemożnością stworzenia modułu phys, który prawdopodobnie wynika ze zbyt krótkich długości niektórych dróg
- zakomentowanie 24-tej liniiki w pliku LinkRealExt.java pozwoliło na **wyświetlenie pierwszej mapy**

26.07.2016

- query wczytuje dużo większy obszar niż zaznaczony
 - tworzą się “wyspy” czyli drogi odseparowane od głównej sieci dróg typu brama - brama
 - potencjalny powód nie działania symulacji
- po odpaleniu programu, kraksim wczytuje i wizualizuje mapę, lecz po uruchomieniu symulacji rzucony jest wyjątek `pl.edu.agh.cs.kraksim.routing.NoRouteException`
- wizualizowana mapa jest obrócona, konkretnie zamieniona jest północ z południem

27.07.2016

- program działa tylko dla współrzędnych które wycinają obszar w północno-wschodniej ćwiartce kuli ziemskiej
- poprawienie granicznych współrzędnych było konieczne ze względu na pobieranie przez kwerendę węzłów, które są poza początkowo podanymi współrzędnymi

28.07.2016

- w OSM oznaczane są drogi o bardzo małej długości (wielokrotnie mniejsze od długości przeciętnego samochodu), co utrudnia uruchomienie wizualizacji kraksima

3 Realizacja harmonogramu

zadanie	czas potrzebny na wykonanie(w godzinach)	faktyczny czas realizacji(w godzinach)
Diagram klas	3	3,5
Wczytanie danych wejściowych (koordynat)	3	3
Zapisanie skrzyżowań i bram do struktury wewnętrznej - implementacja algorytmu konwersji do bloku nr 1	4	10
Zapisanie połączeń między skrzyżowaniami do str. wew. - implementacja algorytmu konwersji do bloku nr 2	3	2
Zapisanie zasad ruchu na skrzyżowaniach do str.wew. - algorytm konwersji do bloku nr 3	10	7
Konwersja str. wew.(bloku 1) do danych w formacie wejścia programu Kraksim	3	7
Konwersja str. wew.(bloku 2) do danych w formacie wejścia programu Kraksim	3	7
Konwersja str. wew.(bloku 3) do danych w formacie wejścia programu Kraksim	5	35
Testy aplikacji	7	7
Stworzenie podręcznika użytkownika	7	0.25
Implementacja GUI	7	nie zrealizowano

4 Rozliczenie realizacji

4.1 Funkcjonalność programu

Wykonany program generuje plik wejściowy dla programu kraksim. Wygenerowany plik pozwala na wyświetlenie mapy w programie kraksim zgodnie z koordynatami wyznaczonymi przez użytkownika. Program nie pozwala na automatyczne przeprowadzenie symulacji wykreowanej mapy, gdyż nie tworzy automatycznie pliku generującego ruch. Nie można stworzyć najprostszego pliku generującego ruch (każda brama do każdej) ze względu na fakt, iż wygenerowany graf z powstałej mapy nie jest spójny. Tworzą się "wyspy" bram, wśród których można przeprowadzić ruch każdy do każdego. W momencie ukończenia projektu, plik generujący ruch można tworzyć tylko manualnie, poprzez wyszukiwanie par bram, które pozwalają na ruch między sobą.

4.2 Realizacja modułów

Moduły ConverterReader, ConverterPrinter, QueryLoader i ConverterRunner zrealizowano zgodnie ze specyfikacją w ramach wykonanych funkcjonalności. Ze względu na ograniczony czas realizacji projektu nie wykonano modułu GUI.

5 Spis najistotniejszych problemów napotkanych w czasie realizacji projektu i sugestie ulepszeń

Najistotniejszym problemem było pobieranie przez Overpass API węzłów z mapy OSM, które wykraczają poza wyznaczony przez użytkownika obszar oraz dróg o bardzo małej długości. Z związku z faktem odrzucania przez kraksim dróg o krótkich długościach powstawał graf niespójny, dla którego nie można wygenerować najprostszego pliku opisującego ruch. W związku z ograniczonym czasem projektu nie zrealizowano wyznaczania spójnych składowych powstałego grafu w celu automatycznego tworzenia pliku generującego ruch. Opisaną czynność można wykonać w celu ulepszenia działania programu.

Inną problematyczną kwestią był sposób odrzucania dróg pobranych przez Overpass API oraz skalowanie mapy. W związku z opisanym wcześniej problemem odrzucania zbyt krótkich dróg przez kraksim, należało stworzyć długość graniczną odrzucanych dróg oraz ustalić jednostkę mapy. Ostatecznie opisane parametry wyznaczono drogą eksperymentalną. W celu ulepszenia programu można przemyśleć inny sposób wyznaczania parametrów. Ponadto parametr skalowania map można zmieniać tylko z poziomu kodu w pliku `converter_reader.py` w funkcji `measure(lat1, lon1, lat2, lon2)` (w wyrażeniu *return* ostatnia wartość przed nawiasem).

Zakładamy, że każda droga, którą otrzymamy w odpowiedzi na zapytanie do OSM prowadzi od skrzyżowania do skrzyżowania, co nie zawsze okazuje się być prawdą. Powoduje to, że pojawiają się bramy w środku drogi zamiast skrzyżowań. Jako usprawnienie programu można napisać algorytm, który będzie wykrywał takie węzły i odpowiednio poprawiał przyległe drogi oraz tworzył skrzyżowanie.

Inne napotkane problemy opisane są w rozdziale "Obserwacje". W celu dogłębnego zapoznania się z problemami sugeruje się porównywanie opisanych obserwacji i

czynności wykonanych danego dnia z kodem, który został umieszczony w repozytorium danego dnia.

6 Uruchomienie programu

Program uruchamiany jest z linii poleceń i przyjmuje pięć argumentów: współrzędną szerokości geograficznej ograniczającą obszar mapy od południa, współrzędną długości geograficznej ograniczającą obszar mapy od zachodu, współrzędną szerokości geograficznej ograniczającą obszar mapy od północy, współrzędną długości geograficznej ograniczającą obszar mapy od wschodu oraz nazwę wyjściowego pliku.

7 Repozytorium projektu

Utworzone zostało repozytorium z kodem naszego projektu, dostępne pod linkiem: <https://github.com/DanOgiela/OSMKraksimConverter>

8 Testy

Wykonano kilka prostych testów, aby sprawdzić jaki jest wpływ stałej skalującej mapę na średnią prędkość i czas przejazdu pojazdów. Stała skalująca nie wpływa na prędkość, natomiast liniowo zmienia czas przejazdu pojazdów. Współrzędne testowanej mapy wynosiły 50.04287 19.92274 50.07876 19.96077.