



Konwerter danych OSM na wejście programu Kraksim

Dokumentacja koncepcyjna

Przedmiot:

Praktyki letnie 2016

Prowadzący:

dr inż. Jarosław Koźlak

Rok akademicki:

2015/2016

Zespół:

Daniel Ogiela

Krzysztof Węgrzyński

Spis treści

1. Wizja systemu.....	3
1.1 Opis problemu.....	3
1.2 Opis rozwiązania.....	3
2. Koncepcja systemu.....	3
2.1 Lista modułów systemu.....	3
2.2 Diagram komunikacji.....	4
2.3 Opis diagramu komunikacji.....	5
3. Architektura systemu.....	5
3.1 Diagram klas.....	5
4. Opis algorytmów konwersji do obiektowej struktury danych.....	7
4.1 Algorytm wykrywania skrzyżowań.....	7
4.2 Algorytm wykrywania bram.....	7
4.3 Algorytm tworzenia akcji.....	7
4.4 Algorytm tworzenia reguł nadrzędności.....	8
5. Plan przedsięwzięcia.....	8
5.1 Wprowadzenie.....	8
5.2 Organizacja przedsięwzięcia.....	9
5.3 Analiza zagrożeń.....	9
5.4 Wykorzystane technologie i oprogramowanie.....	9
5.5 Harmonogram.....	9

1. Wizja systemu

1.1 Opis problemu

Klient potrzebuje narzędzia pozwalającego zautomatyzować mozolny proces tworzenia map do programu *Kraksim*. Mapy w systemie można tworzyć ręcznie, tzn. edytując surowy plik typu .xml lub po przez *city designer* będący częścią systemu. Tworzenie plików wejściowych dla systemu *Kraksim* tymi dostępnymi metodami jest czasochłonne.

1.2 Opis rozwiązania

Tworzona aplikacja będzie konwertować dane pobrane z OpenStreetMap na pliki wejściowe typu .xml programu *Kraksim*. Dane będą pobierane za pomocą specjalnie przygotowanego wcześniej zapytania które pobierze tylko istotne dla systemu informacje. Aplikacja nie będzie integrowana z systemem *Kraksim*.

Po uruchomieniu aplikacji użytkownik będzie miał dostęp do pól umożliwiających wprowadzenie koordynatów regionu który chce przekonwertować na wejście programu *Kraksim*. Przez koordynaty regionu rozumiemy cztery pola:

- szerokość geograficzna
- długość geograficzna
- szerokość geograficzna
- długość geograficzna

Razem koordynaty te powinny tworzyć prostokąt określający region który zostanie przekonwertowany.

Aplikacja udostępni też możliwość podania ścieżki i nazwy pliku wynikowego tj. pliku wejścia do programu *Kraksim* typu .xml oraz przycisk rozpoczynający konwersję, jeżeli tylko wszystkie niezbędne dane zostały podane.

2. Koncepcja systemu

2.1 Lista modułów systemu

1. GUI

Moduł odpowiedzialny za wyświetlenie prostego interfejsu graficznego, pobranie danych, sprawdzenie poprawności pobranych danych oraz przesłanie poprawnych danych do modułu *Konwertera*.

2. ConverterReader

Moduł odpowiedzialny za wczytanie pobranych danych z Open Street Map do wewnętrznej struktury obiektowej.

3. ConverterPrinter

Moduł tworzący plik wejściowy do programu kraksim._

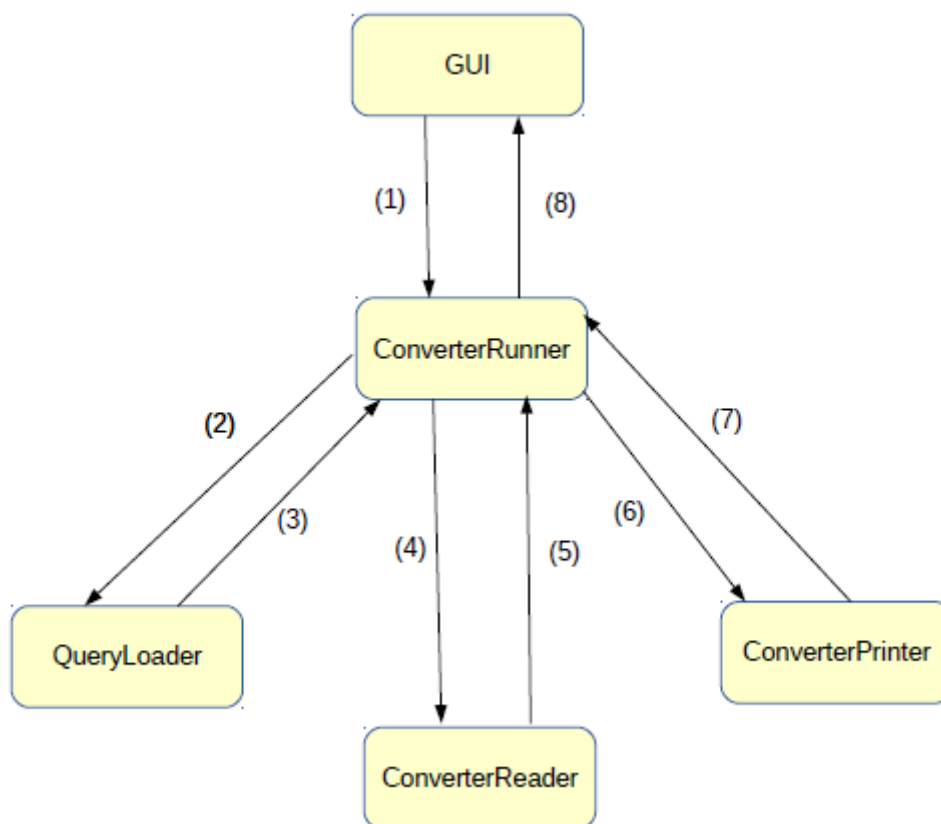
4. QueryLoader

Moduł przygotowujący i wykonujący kwerendę do pobrania odpowiednich danych z Open Street Map. Wykorzystuje Overpass API.

5. ConverterRunner

Moduł koordynujący działanie pozostałych modułów.

2.2 Diagram komunikacji



2.3 Opis diagramu komunikacji

1. Przekazanie współrzędnych interesującego użytkownika obszaru

Moduł GUI po otrzymaniu wszystkich niezbędnych danych sprawdza ich poprawność a następnie przesyła je do modułu ConverterRunner.

2. Przekazanie współrzędnych do modułu QueryLoader

3. Zwrócenie obiektu overpy.Result z odpowiednimi danymi

Do zwrócenia obiektu wykorzystano Overpass API i Python Overpass Wrapper

4. Przekazanie do modułu ConverterReader obiektu overpy.Result

5. Zwrócenie referencji obiektu ConverterReader z uzupełnionymi danymi w wewnętrznej strukturze programu

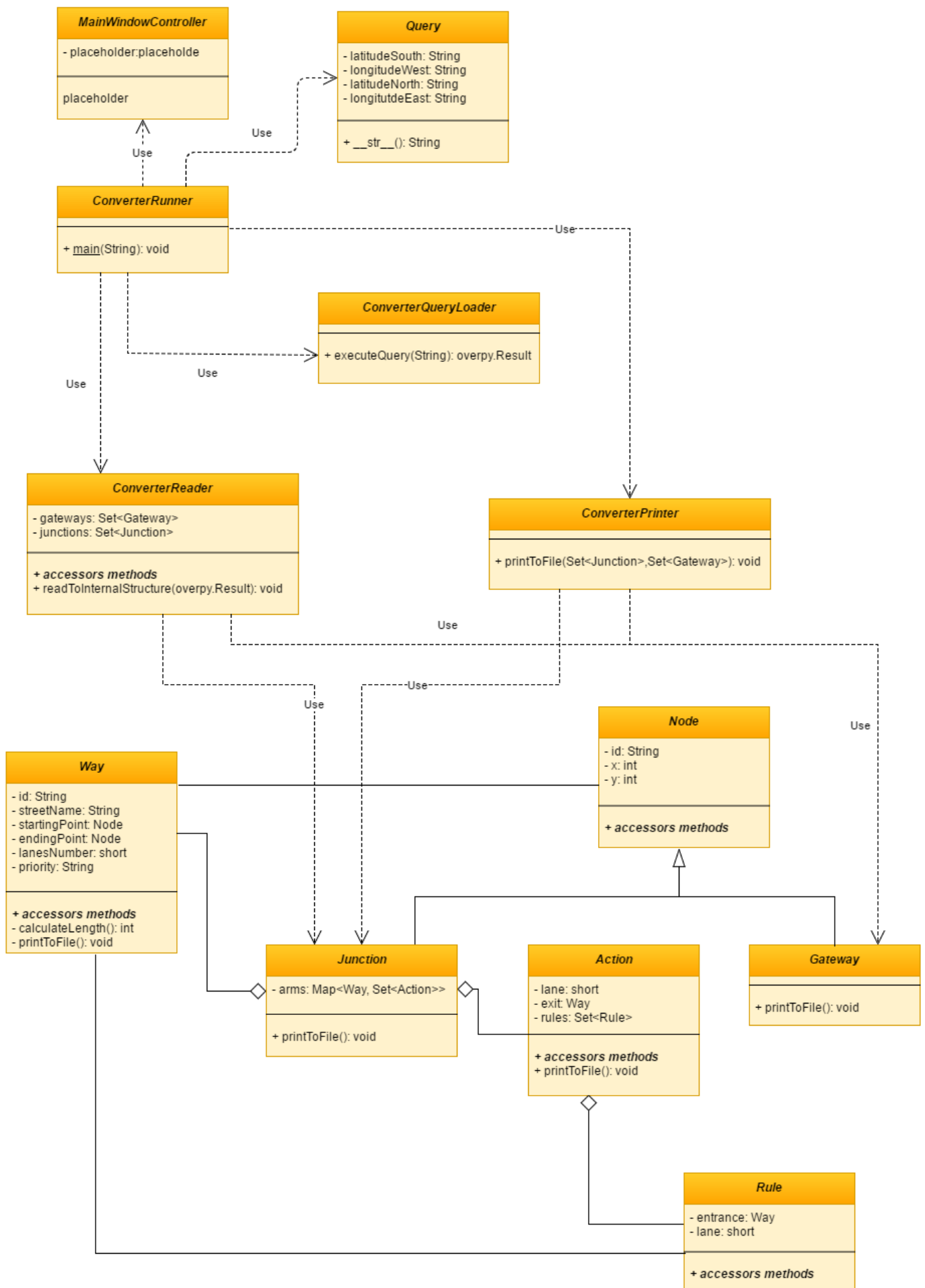
6. Przekazanie odpowiednich pól obiektu ConverterReader do modułu ConverterPrinter

7. Przekazanie informacji o powodzeniu bądź niepowodzeniu wykonania operacji

8. Przekazanie informacji o powodzeniu bądź niepowodzeniu wykonania operacji do modułu GUI

3. Architektura systemu

3.1 Diagram klas



4. Opis algorytmów konwersji do obiektowej struktury danych

4.1 Algorytm wykrywania skrzyżowań

Algorytm na wejście otrzymuje drogi zwrócone z kwerendy. Drogi to lista węzłów, przy czym pierwszy i ostatni węzeł, to potencjalne skrzyżowanie bądź brama. Węzeł zaliczony jest jako skrzyżowanie jeśli co najmniej dwie drogi kończą się lub zaczynają w tym węźle.

Do wykrycia węzłów będących skrzyżowaniami wykorzystano dwie struktury: zbiór oraz listę. Algorytm iteruje po wszystkich drogach i dodaje do obydwu struktur pierwszy i ostatni węzeł każdej drogi. Drugim krokiem algorytmu jest iteracja po zebranych węzłach w zbiorze i usunięcie każdego węzła z listy jeden raz. Te węzły, które zostały w liście są skrzyżowaniami, gdyż powtórzyły się co najmniej dwa razy.

4.2 Algorytm wykrywania bram

Za bramy uznane są węzły, które są początkiem i/lub końcem dokładnie jednej drogi. Mając zbiór węzłów, które są skrzyżowaniami i iterując po pierwszych i ostatnich węzłach każdej drogi wykrycie bram jest proste - są to węzły, które są końcami lub początkami dróg i nie są skrzyżowaniami.

4.3 Algorytm tworzenia akcji

Algorytm na wejściu ma zbiór skrzyżowań, które zawierają przyległe drogi. Dla każdej drogi każdego skrzyżowania wyszukiwane są i tworzone wszystkie możliwe akcje. Potencjalną akcją dla danej drogi jest inna przyległa do danego skrzyżowania droga. Uwzględniane są przypadki dróg jednokierunkowym, które mogą wchodzić bądź wychodzić ze skrzyżowania. Droga

jednokierunkowa, która wchodzi do skrzyżowania może mieć akcje, ale inne drogi przyległe do tego skrzyżowania nie mogą mieć akcji prowadzącej do badanej drogi jednokierunkowej. Z kolei droga jednokierunkowa, która wychodzi ze skrzyżowania nie może mieć akcji na tym skrzyżowaniu, ale inne drogi przyległe do skrzyżowania mogą mieć akcję do badanej drogi jednokierunkowej. Poczyniono również założenie, że drogi nie mogą mieć akcji do samej siebie. W algorytmie uwzględniono również wymaganie kraksima, które polega na tym, że węzły opisujące akcje muszą być inne. Wynika z tego, że węzły, opisujące drogę wchodzącą do skrzyżowania, skrzyżowanie i drogą wychodzącą muszą być inne.

4.4 Algorytm tworzenia reguł nadrzędności

Algorytm na wejściu ma dostęp do obiektu przechowującego całą strukturę obiektową skrzyżowań. Kraksim wymusza istnienie reguł nadrzędności dla wszystkich dróg, które mają takie same "exity" i tylko takich. W celu wykrycia reguł nadrzędności algorytm wykonuje procedurę powtarzaną dla każdego skrzyżowania. Dla każdej przyległej drogi do skrzyżowania, która jest potencjalnym "exitem", wyszukuje wszystkie drogi na skrzyżowaniu wraz z akcją, która dotyczy badanego "exita". Następnie sortuje te drogi wg priorytetów. Pierwszeństwo dla dróg o równych priorytetach ustalane jest w sposób losowy. Mając listę posortowanych dróg w kolejności malejących priorytetów, algorytm począwszy od drugiej drogi w liście, dodaje do badanej akcji danej drogi reguły nadrzędności. Jako reguły nadrzędności dodane zostają wszystkie drogi, które są przed badaną w posortowanej liście. Dla drugiej drogi w liście dodana zostaje 1 droga, dla trzeciej 2 drogi, dla czwartej 3 itd.

5. Plan przedsięwzięcia

5.1 Wprowadzenie

Celem przedsięwzięcia jest stworzenie aplikacji pozwalającej na konwersję mapy z formatu .osm na plik wymagany przez system Kraksim. Czas realizacji przedsięwzięcia jest ograniczony i wynosi 4 tygodnie.

5.2 Organizacja przedsięwzięcia

Projekt jest realizowany przez zespół 2-osobowy.

5.3 Analiza zagrożeń

Zagrożenie	Prawdopodobieństwo	Strategia
Niekompletność danych z OSM potrzebnych do specyfikacji pliku wejściowego do Kraksim	średnie	Stworzenie pseudokodów algorytmu konwersji

5.4 Wykorzystane technologie i oprogramowanie

- Python 2.7
- OpenStreetMap
- Overpass API
- Python Overpass Wrapper
- PyCharm

5.5 Harmonogram

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

Podsumowanie	
Sumaryczny czas na wszystkie zadania	55h
Suma dni pracy	8 dni roboczych