



Miejsca Obsługi Podróżnych

Wizja, przypadki użycia

Magdalena Grabowska, Michał Kukuła, Klaudia Laks, Przemysław Perkowski

Nasz system informatyczny powstanie w ramach projektu "Miejsca parkingowe na MOP" realizowanego przez konsorcjum uczelni Uniwersytet Warszawski, Politechnika Warszawska i Instytutu Transportu Samochodowego. Jego poszczególne elementy będą w założeniu wykorzystywane głównie przez pracowników Generalnej Dyrekcji Dróg Krajowych i Autostrad.



UNIWERSYTET
WARSZAWSKI



**Politechnika
Warszawska**



GDDKiA

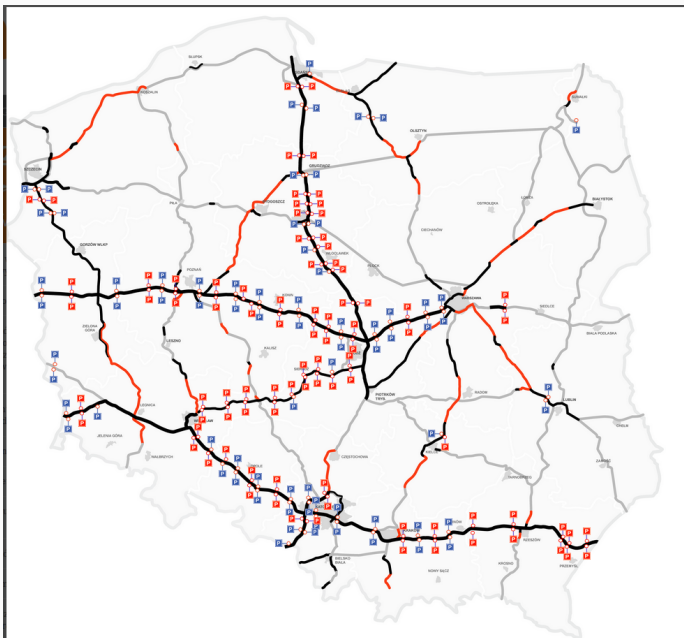
Miejsca obsługi podróżnych - co to?



Teren wydzielony w pasie drogowym poza koroną drogi, wyposażony w miejsca postojowe dla pojazdów oraz w urządzenia służące zaspokajaniu potrzeb podróżnych.

MOP-y są ściśle związane z drogami o ograniczonej dostępności (autostrady i drogi ekspresowe) i znajdują się w pasie drogowym.

Miejsca obsługi podróżnych w Polsce



1. Rozmieszczenie nowych MOP

1. Rozmieszczenie nowych MOP
2. Liczba miejsc parkingowych

1. Rozmieszczenie nowych MOP
2. Liczba miejsc parkingowych
3. Szukanie najemców - tym się nie zajmujemy (ale to poważny problem)

- Sieć drogowa (OpenStreetMap)

- Sieć drogowa (OpenStreetMap)
- Macierze podróży

- Sieć drogowa (OpenStreetMap)
- Macierze podróży
 - Środek transportu
 - Cel (dom-praca, turystyka...)

- Sieć drogowa (OpenStreetMap)
- Macierze podróży
 - Środek transportu
 - Cel (dom-praca, turystyka...)
- Średniodobowe Natężenia Ruchu

- Sieć drogowa (OpenStreetMap)
- Macierze podróży
 - Środek transportu
 - Cel (dom-praca, turystyka...)
- Średniodobowe Natężenia Ruchu
- Informacje o istniejących MOPach.

- Sieć drogowa (OpenStreetMap)
- Macierze podróży
 - Środek transportu
 - Cel (dom-praca, turystyka...)
- Średniodobowe Natężenia Ruchu
- Informacje o istniejących MOPach.
- Dane o wjazdach i wyjazdach z konkretnych MOPów

1. Mopnik
2. Mopsim
3. Aplikacja Mobilna i strona WWW

Mopnik

1. Wyznaczanie liczby wolnych miejsc na Miejscach Obsługi Podróżnych

1. Wyznaczanie liczby wolnych miejsc na Miejscach Obsługi Podróżnych
2. Aplikacja okienkowa

1. Wyznaczanie liczby wolnych miejsc na Miejscach Obsługi Podróżnych
2. Aplikacja okienkowa
3. Użytkownicy: inżynierowie GDDKiA

Predykcje

Natężenie
ruchu

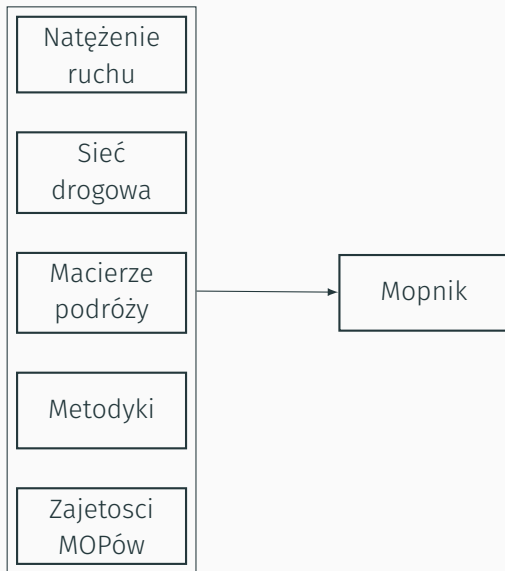
Sieć
drogowa

Macierze
podróży

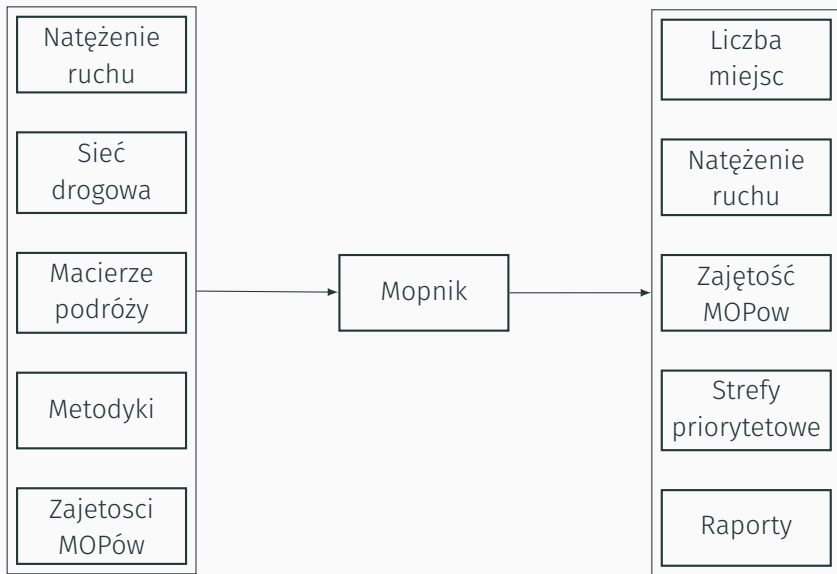
Metodyki

Zajętości
MOPów

Predykcje



Predykcje



- Obecnie: wzory matematyczne

- Obecnie: wzory matematyczne
- Do rozważenia: algorytmy uczenia maszynowego

- Obecnie: wzory matematyczne
- Do rozważenia: algorytmy uczenia maszynowego
- Możliwość ręcznego doboru parametrów

- Wczytywanie danych

- Wczytywanie danych
- Graficzny interfejs

- Wczytywanie danych
- Graficzny interfejs
 - Przeglądanie danych na mapach

- Wczytywanie danych
- Graficzny interfejs
 - Przeglądanie danych na mapach
 - Umieszczenie na mapie nowej drogi lub nowego MOPu

- Java (lub C#, C++) - całość aplikacji wraz z GUI
- OpenStreetMap, JOSM - wprowadzanie danych o strukturze sieci drogowej w postaci plików (.osm), wizualizacja i edycja map

Mopsim

1. Program do symulacji ruchu pojazdów na sieci dróg krajowych i autostrad w Polsce

1. Program do symulacji ruchu pojazdów na sieci dróg krajowych i autostrad w Polsce
2. Aplikacja okienkowa

1. Program do symulacji ruchu pojazdów na sieci dróg krajowych i autostrad w Polsce
2. Aplikacja okienkowa
3. Użytkownicy: inżynierowe GDDKiA

1. Wprowadzanie danych: struktura sieci drogowej, macierze podróży, natężenia ruchu
2. Przeprowadzanie symulacji
3. Prezentacja Danych - wspólne GUI z programem Mopnik

Połączenie modeli mikro- mezo- i makroskopowych:

Połączenie modeli mikro- mezo- i makroskopowych:

- Każdy pojazd to agent (mikro)

Połączenie modeli mikro- mezo- i makroskopowych:

- Każdy pojazd to agent (mikro)
- Model kolejkowy

Połączenie modeli mikro- mezo- i makroskopowych:

- Każdy pojazd to agent (mikro)
- Model kolejkowy
- Symulacje ruchu na pojedynczym MOPie (mikro)

Połączenie modeli mikro- mezo- i makroskopowych:

- Każdy pojazd to agent (mikro)
- Model kolejkowy
- Symulacje ruchu na pojedynczym MOPIe (mikro)
- Skrzyżowania, wjazdy na autostrady (mezo)

Połączenie modeli mikro- mezo- i makroskopowych:

- Każdy pojazd to agent (mikro)
- Model kolejkowy
- Symulacje ruchu na pojedynczym MOPIe (mikro)
- Skrzyżowania, wjazdy na autostrady (mezo)
- Ruch na drogach krajowych i autostradach (makro)

- sieć drogowa Polski - 110120 wierzchołków i 179515 krawędzi

- sieć drogowa Polski - 110120 wierzchołków i 179515 krawędzi
- dane z Krajowego Modelu Ruchu (ŚDRR, Wskaźniki wzrostu ruchu)

- sieć drogowa Polski - 110120 wierzchołków i 179515 krawędzi
- dane z Krajowego Modelu Ruchu (ŚDRR, Wskaźniki wzrostu ruchu)
- dane z Generalnego Pomiaru Ruchu, m.in.:

- sieć drogowa Polski - 110120 wierzchołków i 179515 krawędzi
- dane z Krajowego Modelu Ruchu (ŚDRR, Wskaźniki wzrostu ruchu)
- dane z Generalnego Pomiaru Ruchu, m.in.:
 - Średni Dobowy Ruch Roczny

- sieć drogowa Polski - 110120 wierzchołków i 179515 krawędzi
- dane z Krajowego Modelu Ruchu (ŚDRR, Wskaźniki wzrostu ruchu)
- dane z Generalnego Pomiaru Ruchu, m.in.:
 - Średni Dobowy Ruch Roczny
 - wskaźniki wzrostu ruchu

- Kierowca zazwyczaj wybiera najkrótszą lub najszybszą trasę

- Kierowca zazwyczaj wybiera najkrótszą lub najszybszą trasę
- Algorytm Dijkstry

- Kierowca zazwyczaj wybiera najkrótszą lub najszybszą trasę
- Algorytm Dijkstry
- Algorytm heurystyczny A^* - czas to odległość między punktami *
średnia prędkość na trasie

- Generowanie wykresów

- Generowanie wykresów
- Generowanie raportów

- Generowanie wykresów
- Generowanie raportów
- Eksport danych do pliku

- Generowanie wykresów
- Generowanie raportów
- Eksport danych do pliku
- Eksport do bazy danych (SQLite)

- Generowanie wykresów
- Generowanie raportów
- Eksport danych do pliku
- Eksport do bazy danych (SQLite)

1. Wprowadzanie danych, ustawianie wartości parametrów.

1. Wprowadzanie danych, ustawianie wartości parametrów.
2. Przeglądanie danych na mapie.

1. Wprowadzanie danych, ustawianie wartości parametrów.
2. Przeglądanie danych na mapie.
3. Edycja sieci drogowej.

1. Wprowadzanie danych, ustawianie wartości parametrów.
2. Przeglądanie danych na mapie.
3. Edycja sieci drogowej.
4. Przeprowadzanie symulacji.

1. Wprowadzanie danych, ustawianie wartości parametrów.
2. Przeglądanie danych na mapie.
3. Edycja sieci drogowej.
4. Przeprowadzanie symulacji.
5. Zapisywanie danych, wyników edycji i analiz.

1. Wprowadzanie danych, ustawianie wartości parametrów.
2. Przeglądanie danych na mapie.
3. Edycja sieci drogowej.
4. Przeprowadzanie symulacji.
5. Zapisywanie danych, wyników edycji i analiz.
6. Generowanie raportów.

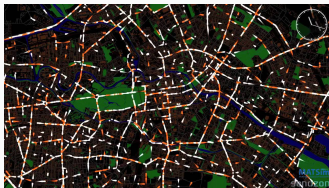
A może jest już coś takiego?

A może jest już coś takiego?

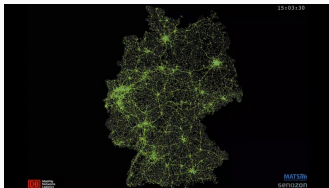
MATSim

Multi-Agent Transport Simulation

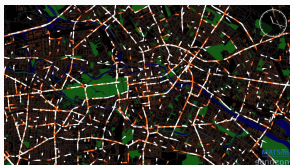
- napisany w Javie opensource'owy framework służący do symulacji ruchu drogowego w dużej skali
- zapoczątkowany w 2004 przez profesora ETH Kaia Nagla
- rozwijany głównie na politechnikach w Zurychu i Berlinie
- składa się z kilku modułów, funkcjonalności każdego mogą być wykorzystywane oddzielnie lub łączone
- łatwo modyfikowalne moduły, publiczne repozytorium



Symulacja ruchu w Berlinie



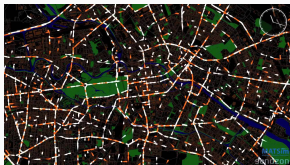
Symulacja ruchu w Niemczech



sieć drogowa



ludzie + ich obowiązki,
plany i wybory



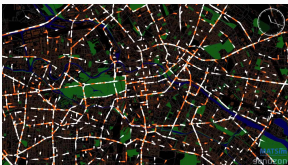
sieć drogowa



ludzie + ich obowiązki,
plany i wybory



symulacja ruchu



sieć drogowa



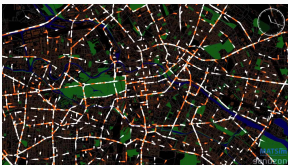
ludzie + ich obowiązki,
plany i wybory



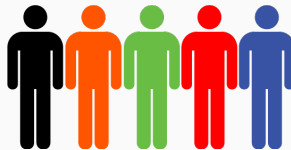
symulacja ruchu



zadowolenie ludzi z podjętych decyzji...



sieć drogowa



ludzie + ich obowiązki,
plany i wybory



symulacja ruchu

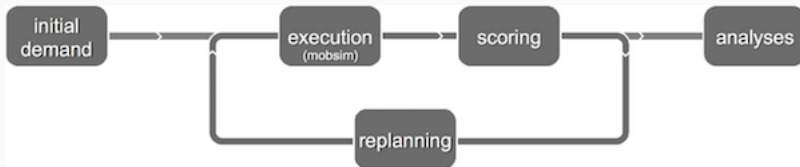


zadowolenie ludzi z podjętych decyzji...



...które wpływa na ich kolejne decyzje, a zatem też kolejne symulacje

MATSim - podział na funkcjonalności



- initial demand - początkowe ustawienia konfiguracyjne
- mobsim - moduł odpowiedzialny za samą symulację
- scoring - obliczanie zadowolenia poszczególnych agentów z przeprowadzonej symulacji
- replanning - modyfikowanie planów niezadowolonych agentów
- analyses - dostarczanie danych analitycznych, możliwych do wykorzystania m.in. w Vio (programie do tworzenia animacji ruchu drogowego)

3 główne pliki konfiguracyjne:

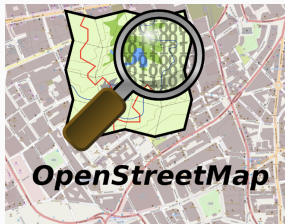
- Metadane o symulacji
- Informacje o populacji
- Informacje o sieci drogowej

OpenStreetMap i JOSM - przydatne narzędzia

Przydatne do tworzenia pliku konfiguracyjnego sieci drogowej mogą się okazać serwis OpenStreetMap i aplikacja JOSM - MATSim posiada wbudowane narzędzia do konwersji map .osm.



JOSM



OpenStreetMaps

1. każdy agent wybiera losowo plan
2. agenci odbywają zajęcia zgodnie z ustalonym planem
3. Ich podróże odbywają się, uwzględniając ruch innych pasażerów i pojemność dróg
4. Na koniec dla każdego z agentów obliczane jest zadowolenie

$$U^{total} = U^{perform} - U^{travel} - U^{wait} - U^{late} \quad (1)$$

1. Dla każdego z agentów obliczana jest użyteczność
2. Brane pod uwagę są wykonane aktywności, od których odejmujemy to co straciliśmy na długich podróżach, czekaniu lub spóźnieniach
3. Uzyskana użyteczność jest podstawą do ponownego planowania

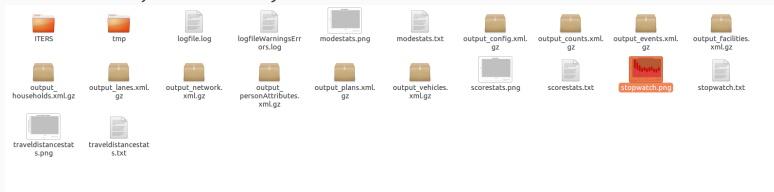
Można wyróżnić 3 główne strategie replaningu:

1. Wybieranie z większym prawdopodobieństwem tych planów, od których oczekujemy wysokiej użyteczności
2. Modyfikowanie planu dnia tak, aby uniknąć spóźnień, korków, itp.
3. Szukanie szybszych dróg



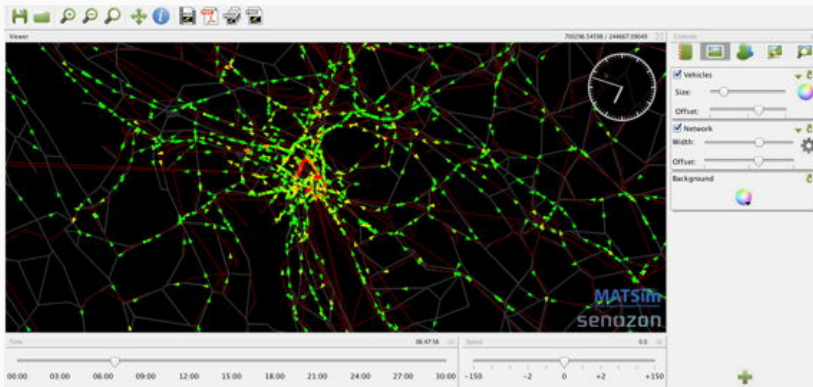
Po fazie ponownego planowania po raz kolejny uruchamiana jest symulacja. Kolejne iteracje zwykle przynoszą zadowalający wzrost wartości funkcji użyteczności.

Wynikiem działania programu jest szereg plików z danymi dotyczącymi każdej iteracji i wykresami. Dane te mogą zostać wykorzystane do wizualizacji lub dalszych analiz.



Via - program do wizualizacji

1. Program firmy Senozon związanej z twórcami MATSim
2. Płatny, bezpłatna wersja posiada ograniczenie do 500 agentów
3. Polega na układaniu odpowiednich warstw na podstawie danych zwróconych przez MATSim



Symulacja na dużą skalę wymaga dużej mocy obliczeniowej. MATSim, oparty na agentach system symulacji, powoli staje się nieefektywny dla mniej wydajnych jednostek obliczeniowych. Rozwiązaniem problemu może być skalowanie scenariuszy, np. poprzez zmniejszenie populacji. Pojawiają się także pomysły makroskopijnych symulacji.

- MatSim + zrównoleglanie obliczeń.
- TensorFlow + ApacheSpark

Aplikacja Mobilna i strona WWW

- Informowanie o zajętości konkretnego MOPa w danej chwili i w niedalekiej przyszłości
- Korzysta z sytemów Mopnik i Mopsim
- Użytkownicy: użytkownicy dróg ekspresowych i autostrad

- Sprawdzenie liczby wolnych miejsc parkingowych na danym MOPie w czasie rzeczywistym

- Sprawdzenie liczby wolnych miejsc parkingowych na danym MOPie w czasie rzeczywistym
- Sprawdzenie przewidywanej liczby wolnych miejsc parkingowych na danym MOPie w danym (niezbyt odległym) czasie

Aplikacja mobilna:

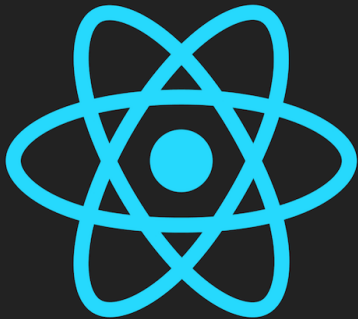
- React Native - framework do tworzenia aplikacji mobilnych

Aplikacja mobilna:

- React Native - framework do tworzenia aplikacji mobilnych

Strona WWW:

- D3.JS - biblioteka javascriptowa do wizualizacji danych
- Django



React Native

- Język: JavaScript
- Aplikacja zbudowana z komponentów, które są renderowane
- Hot Reloading
- Ten sam kod dla wielu platform - przeniesienie aplikacji z Androida na iOS wymaga tylko dodania jednego pliku konfiguracyjnego i zmian w stylowaniu
- Stylowanie w JS, zbliżone do CSS

React Native

```
1 import React, { Component } from 'react';
2 import { Text, View } from 'react-native';
3
4 class WhyReactNativeIsSoGreat extends Component {
5   render() {
6     return (
7       <View>
8         <Text>
9           You just use native components like 'View' and 'Text',
10           instead of web components like 'div' and 'span'.
11         </Text>
12       </View>
13     );
14   }
15 }
```

Listing 1: Przykład

<https://facebook.github.io/react-native/>

- Język: JavaScript
- Nazwy i wartości takie jak w CSS
- Camel case: `backgroundColor` zamiast `background-color`

React Native - style

```
1 import React, { Component } from 'react';
2 import { AppRegistry, StyleSheet, Text, View } from 'react-
  native';
3
4 export default class LotsOfStyles extends Component {
5   render() {
6     return (
7       <View>
8         <Text style={styles.red}>just red</Text>
9         <Text style={styles.bigblue}>just bigblue</Text>
10        <Text style={[styles.bigblue, styles.red]}>bigblue, then
          red</Text>
11        <Text style={[styles.red, styles.bigblue]}>red, then
          bigblue</Text>
12      </View>
13    );
14  }
15 }
```

Listing 2: Przykład stylowania

React Native - style

```
1 const styles = StyleSheet.create({
2   bigblue: {
3     color: 'blue',
4     fontWeight: 'bold',
5     fontSize: 30,
6   },
7   red: {
8     color: 'red',
9   },
10 });
11
12 AppRegistry.registerComponent('
  AwesomeProject', () => LotsOfStyles);
```

Listing 3: Przykład stylowania c.d.

<https://facebook.github.io/react-native/docs/style.html>





Facebook



Facebook



Instagram



Facebook



Airbnb



Instagram



Facebook



Airbnb



Instagram



Skype

Aplikacje w React Native



Facebook



Airbnb



Tesla



Instagram



Skype

Aplikacje w React Native



Facebook



Airbnb



Tesla



Instagram



Skype



UberEATS



<https://d3js.org/>