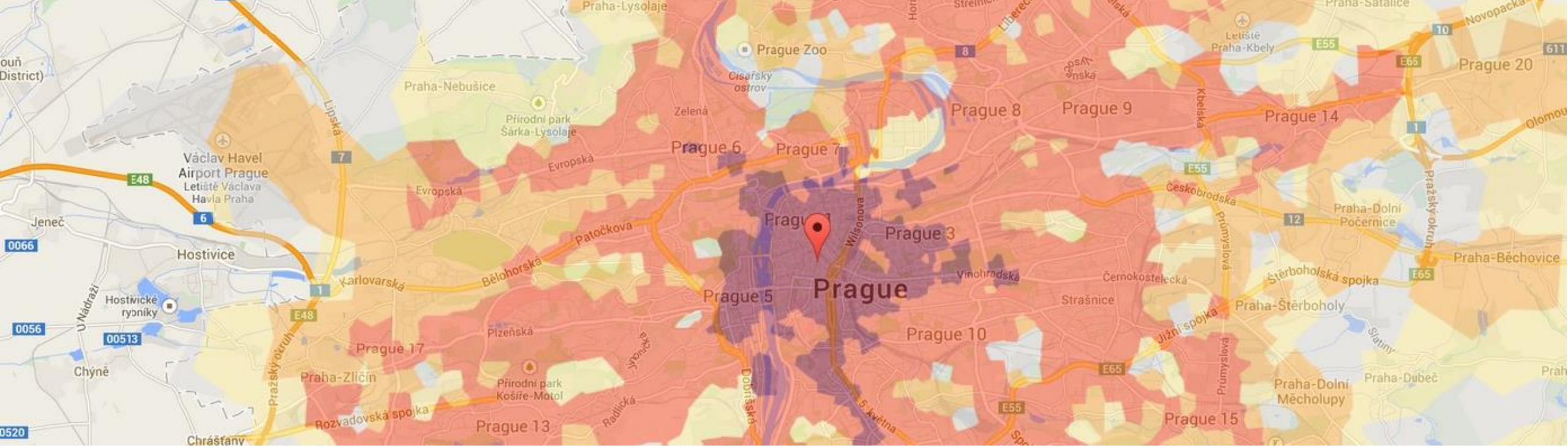
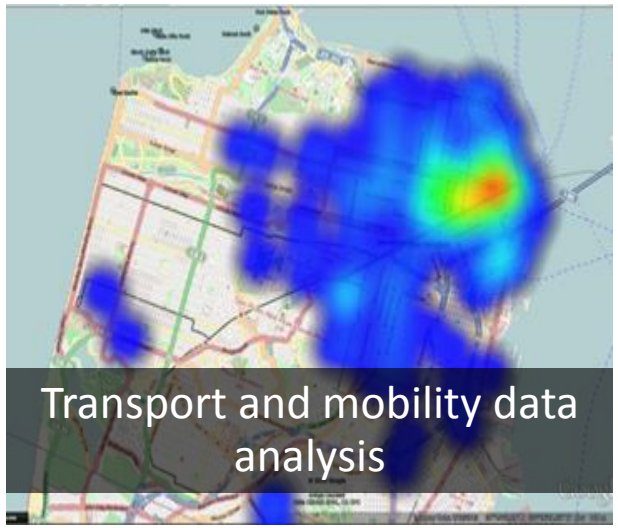


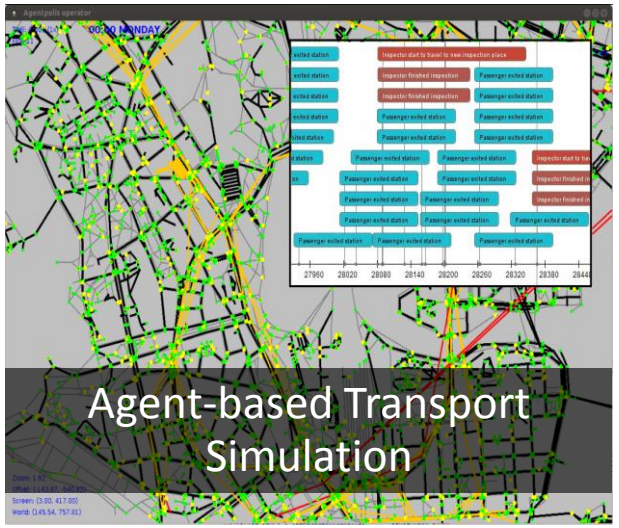
Efficient Fine-grained Analysis of Urban Transport Accessibility

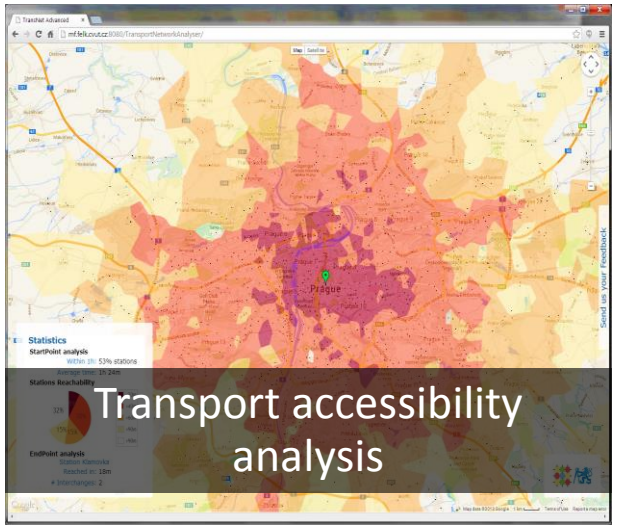
Jan Nykl, Michal Jakob and Jan Hrnčíř
Faculty of Electrical Engineering
Czech Technical University in Prague
http://agents4its.net

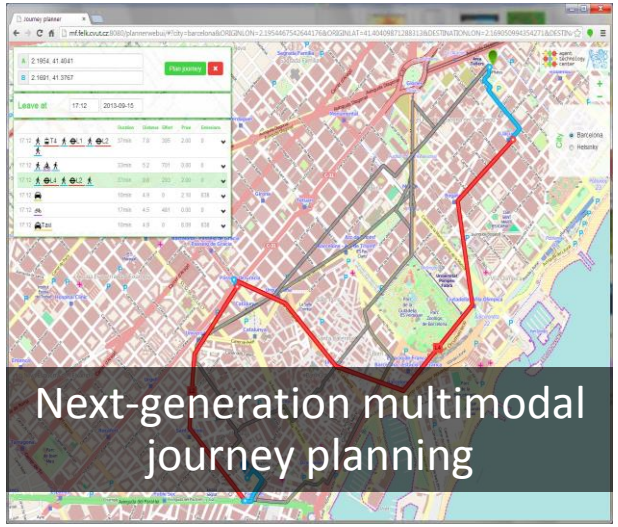
CZECH TECHNICAL UNIVERSITY IN PRAGUE

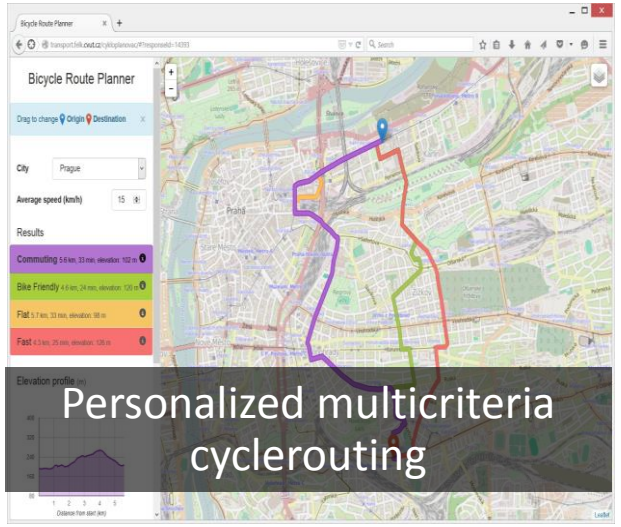
Agent-based Computing for Intelligent Transport Systems Group

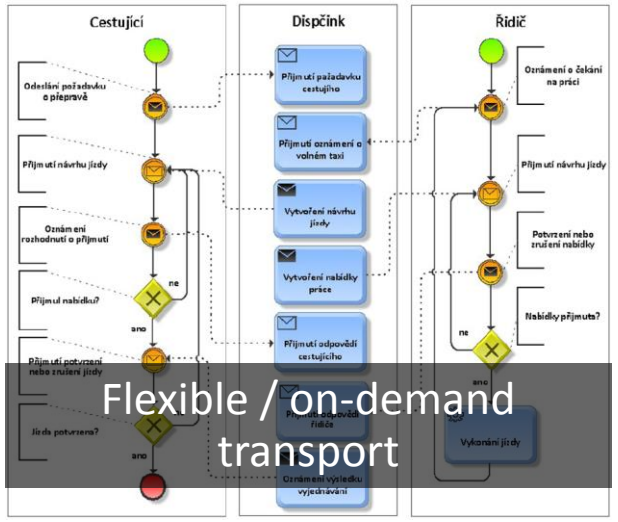




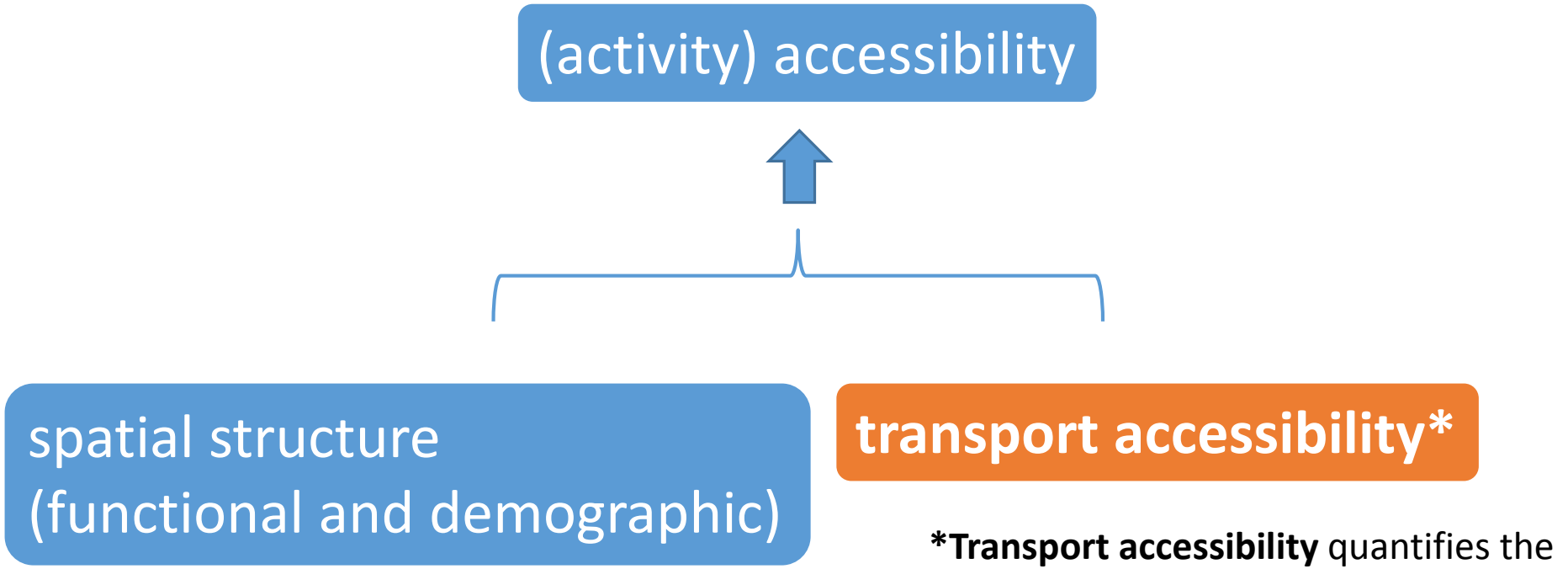








(Transport) Accessibility



***Transport accessibility** quantifies the ease of reaching locations by (different means of) transport.

Related Work

⬅ Prior to ~2005

Simple models of the transport network (core network)
Travel time only
No constraints

Last 10 years ➡

Finer-grained models of the transport network (GTFS timetables)
Still (mostly) travel time only
Slow to calculate

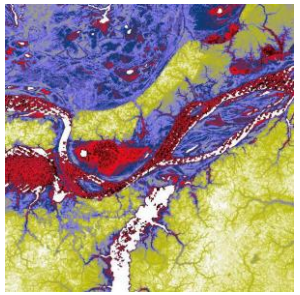
Our approach

- **Fine-grained model** of the transport network (→ generalised time-dependent graphs)
- Realistic **travel constraints** (number of transfers, max walking distance)
- **Efficient** single-origin multi-destination **computation**
- **Multiple metrics**

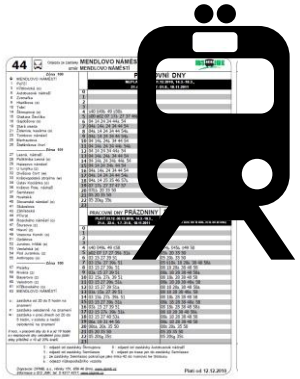
Fine-grained Transport Network Model

Maps (OSM)

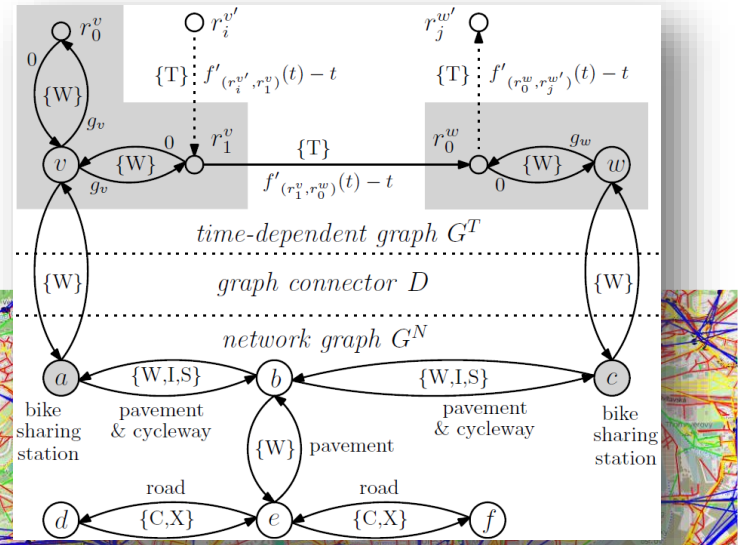
Elevation data

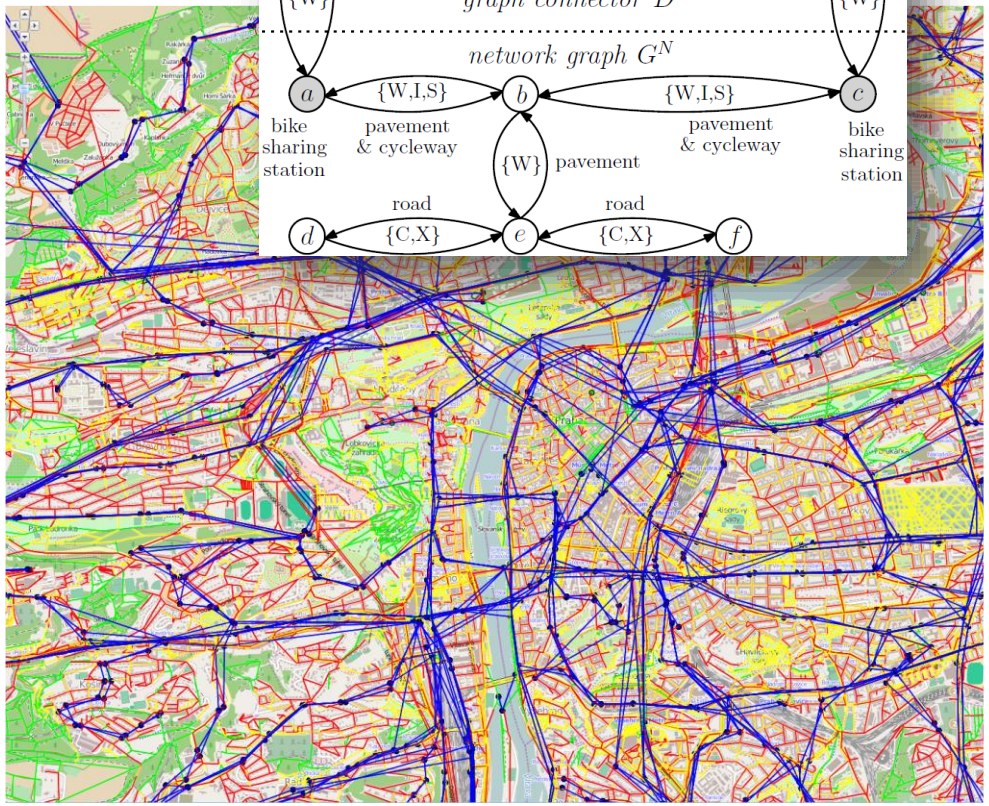


Timetables (JDF/GTFS)



Planning graph builder





Generalised time-dependent graph

(large: 10^5-10^6 nodes and edges)

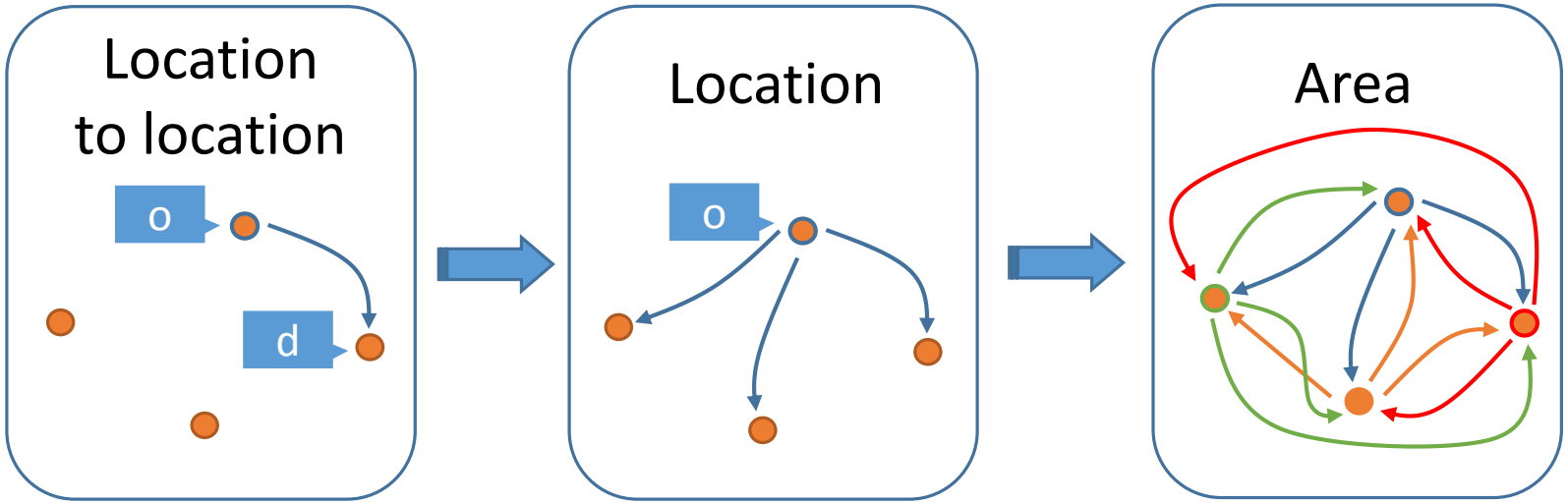
Accessibility Metrics Supported

Public transport	Car	Bike
• Time • Transfers • Frequency • *Overall*	• Time • Distance • Fuel Consumption • *Overall*	• Time • Distance • Physical Effort • Elevation gain • *Overall*

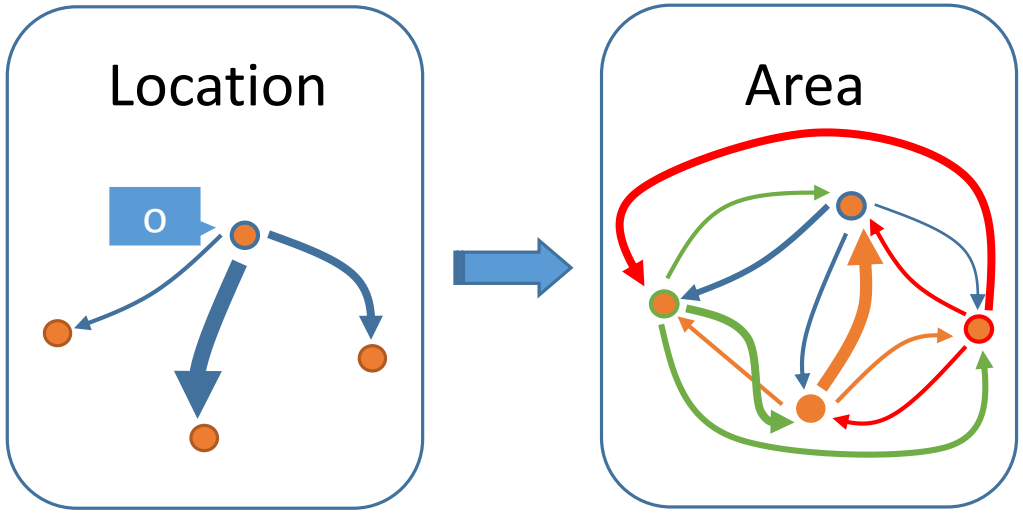
Metrics values vs. levels (discretization)

Accessibility Metrics: Hierarchy

uniform / unweighted



demand- weighted



+ travel demand data
(e.g. from mobile networks)

Algorithm

Modified Dijkstra's algorithm [Dijkstra59]

→ multiple destinations within a single run

Real-world constraints supported:

- number of transfers
- max. walking distance

Key structure: **Contextual Graph View**

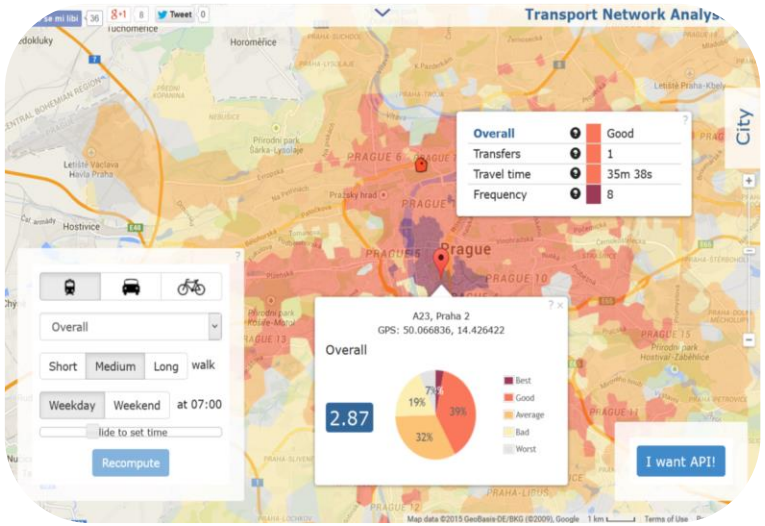
- graph-like structure
- context nodes {node, time, #transfers, walking}
- allows encoding real-world constraints

Algorithm 2: Generating of contextual successors for public transport contextual graph view.

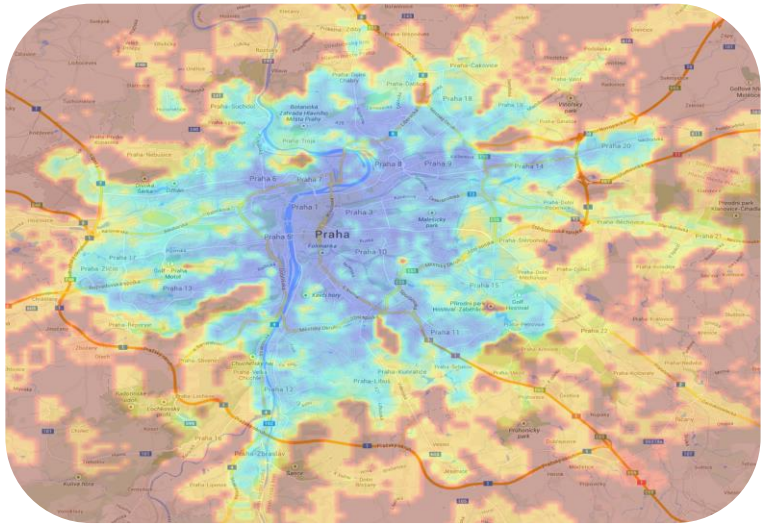
Data: GTD graph $G = (V, E)$, settings S
Input: contextual node n
Output: successors of n

```
1  successors := empty set
2  outgoing := G.getOutgoingEdges(n.GTDNode)
3  foreach edge ∈ outgoing do
4      transfers := n.trasnfers
5      if edgeCausesTransfer(edge) then
6          transfers = transfers + 1
7      end
8      remWalk := n.remWalk − edge.walk
9      arrival := edge.getArrivalToTail(n.arrival)
10     if remWalk > 0 & transfers ≤ S.maxTransfers then
11         successors.add(newContextNode(
           edge.tail,arrival,transfers,remainingWalk))
12     end
13 end
14 return successors
```

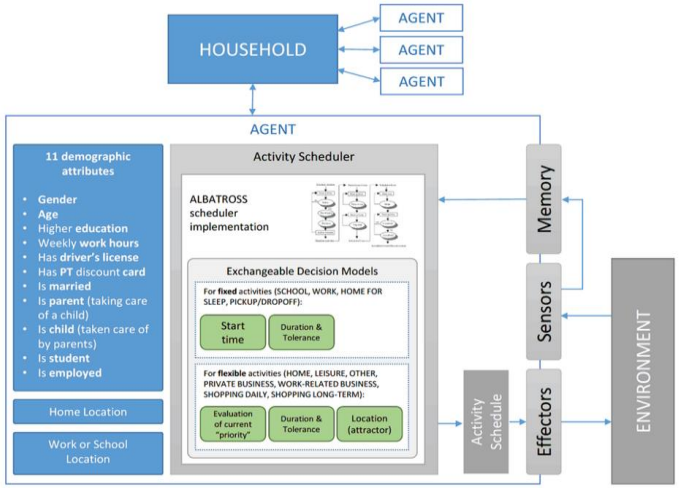
Applications / Use cases



End-user location accessibility analysis

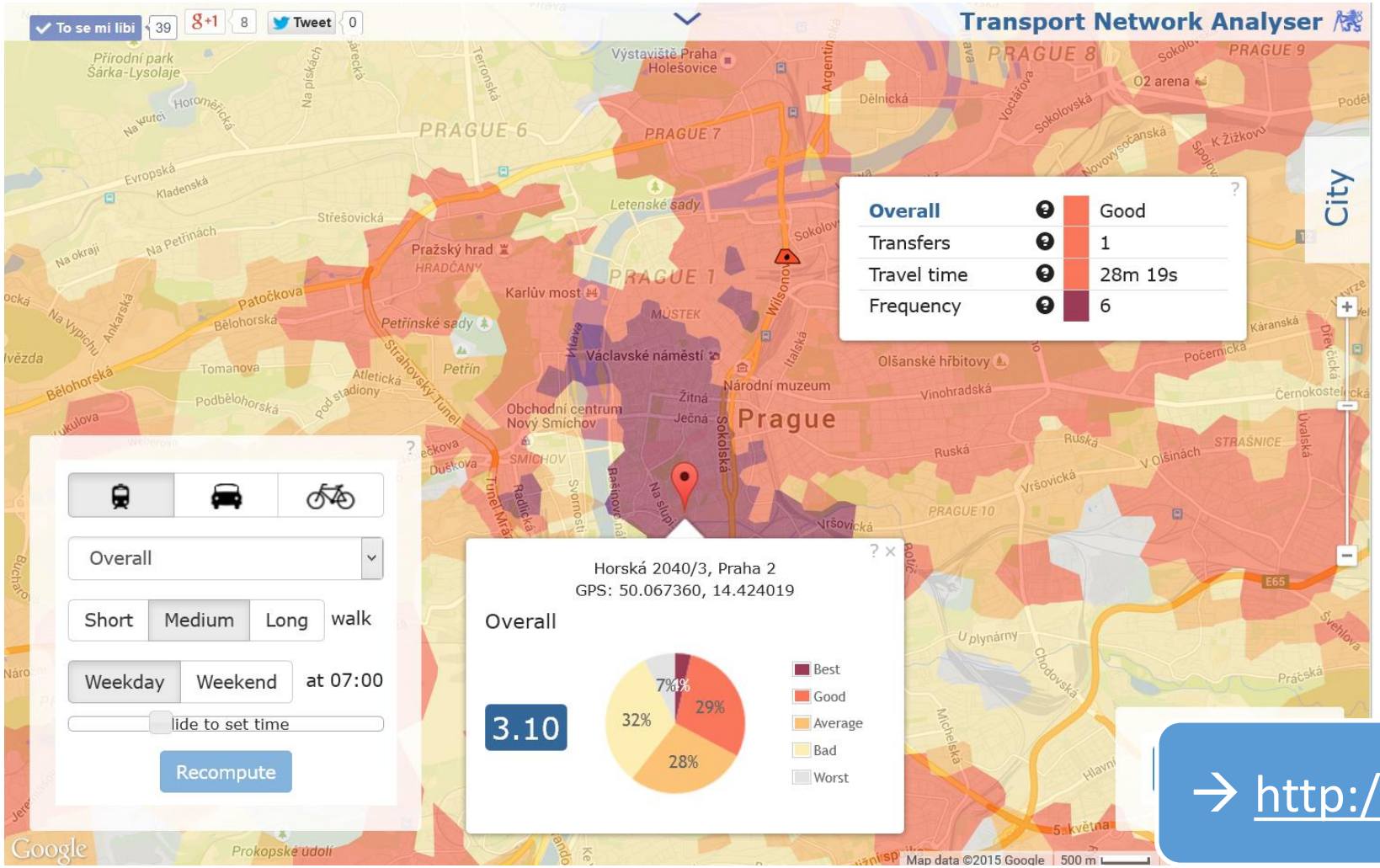


Expert area accessibility analysis tools



Efficient travel time estimation for activity-based models

End-user Location Analysis Web App



Interactive **Location analysis** for general public.

Responsive: calculation time: 1s

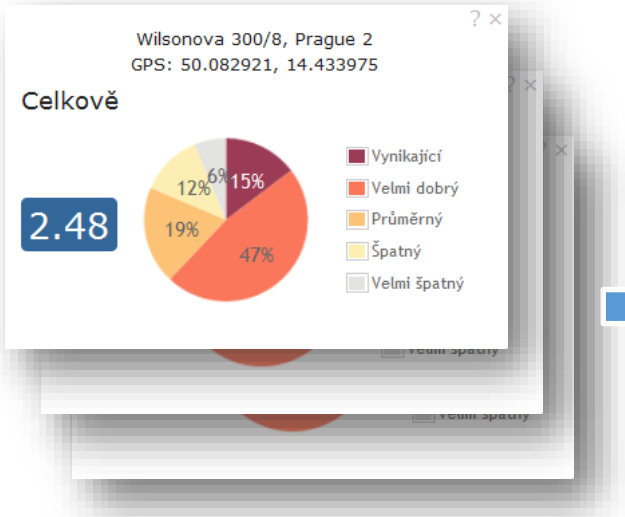
Use cases

- property rental
- facility location
- …

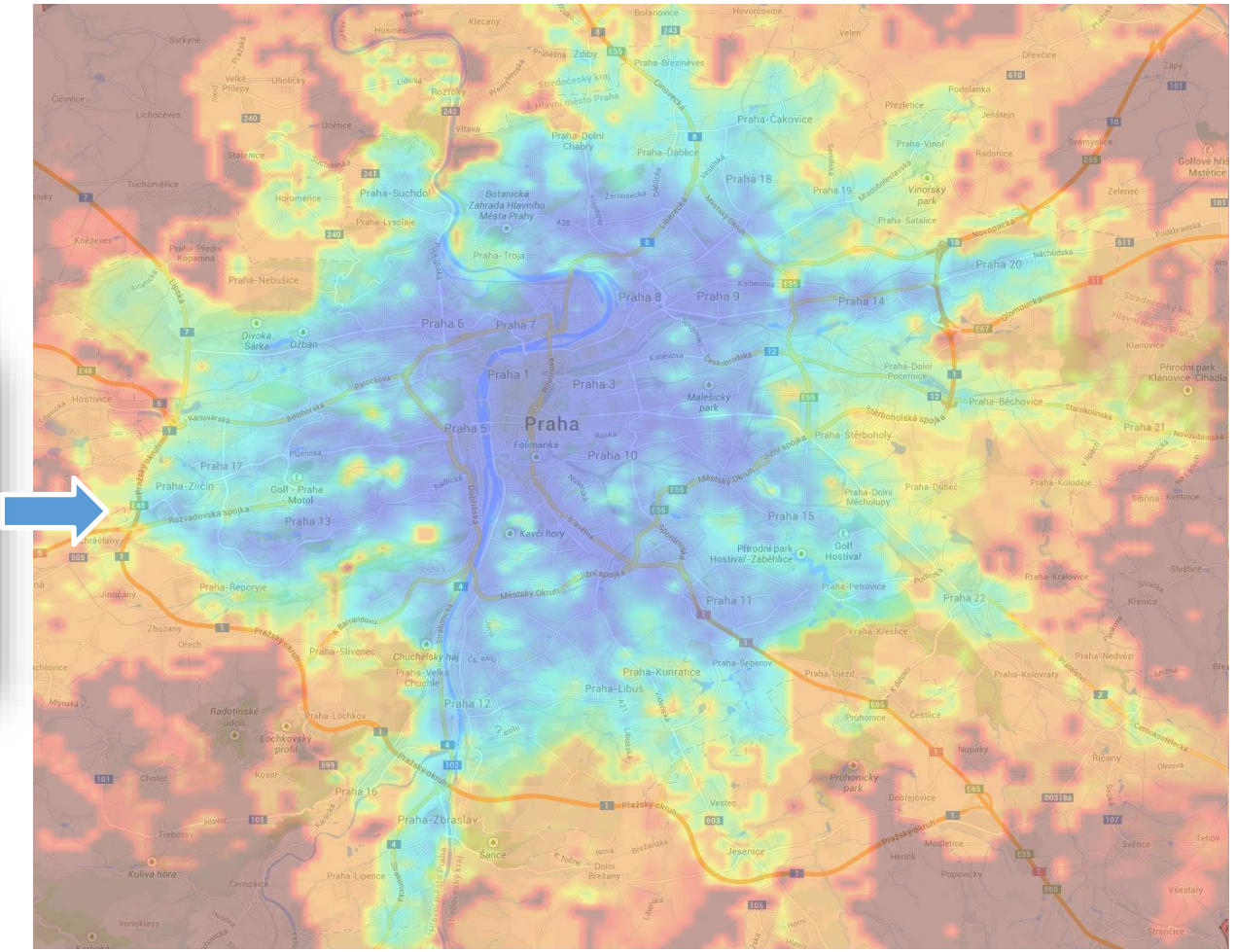
→ http://transportanalyser.com

Timetables from Autumn 2013

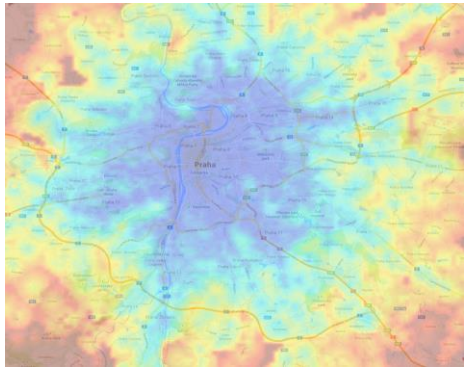
Area Accessibility Maps



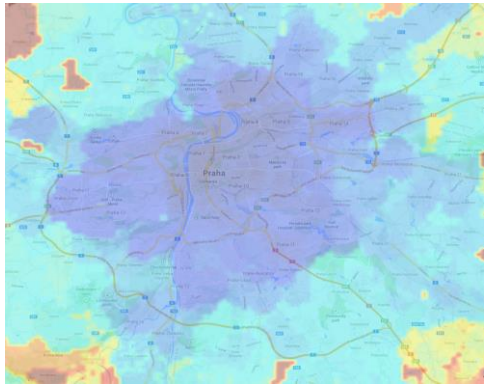
Multiple **location** accessibility analyses



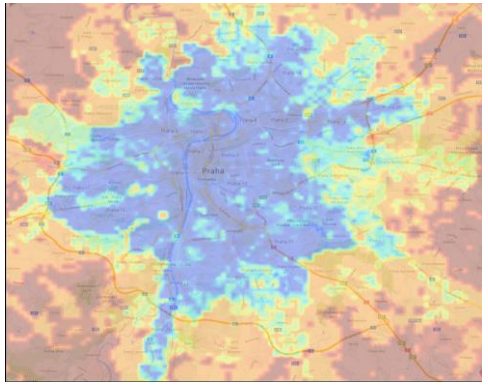
Overall **area** accessibility by public transport



Time



Transfers

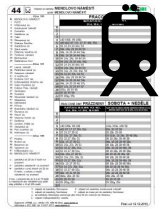


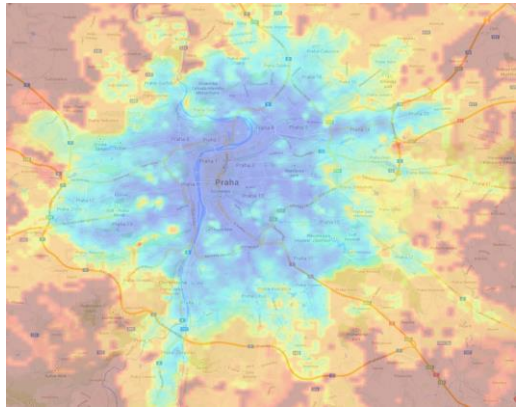
Frequency

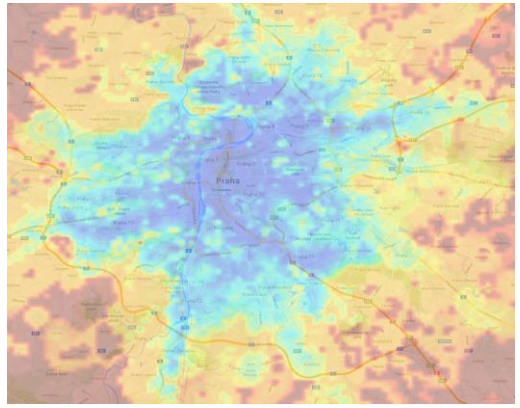
Comparison / What-if Analysis

Scenario 1:

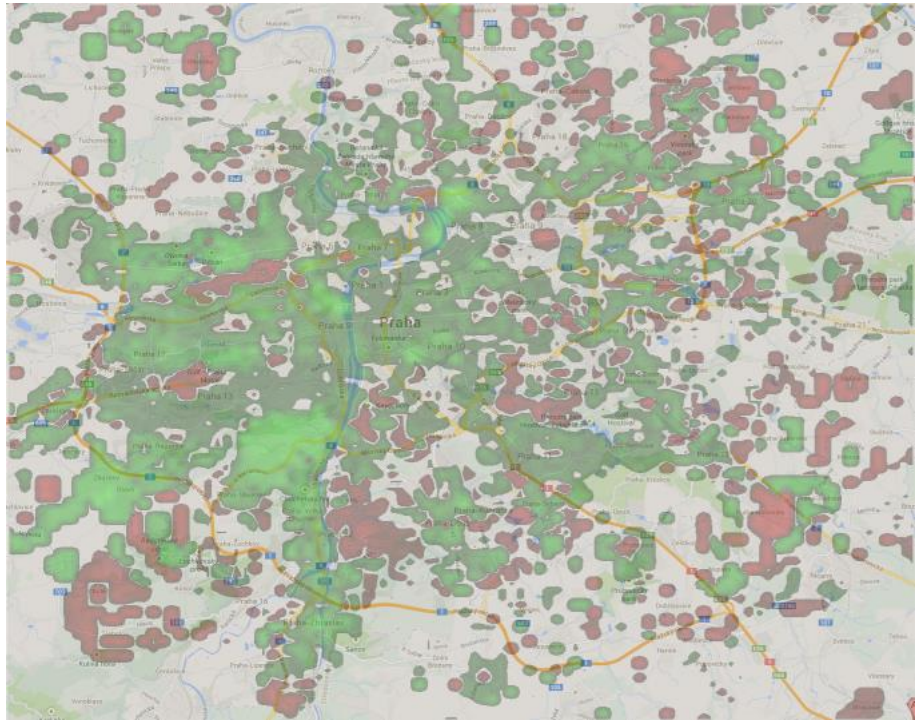
Scenario 2:





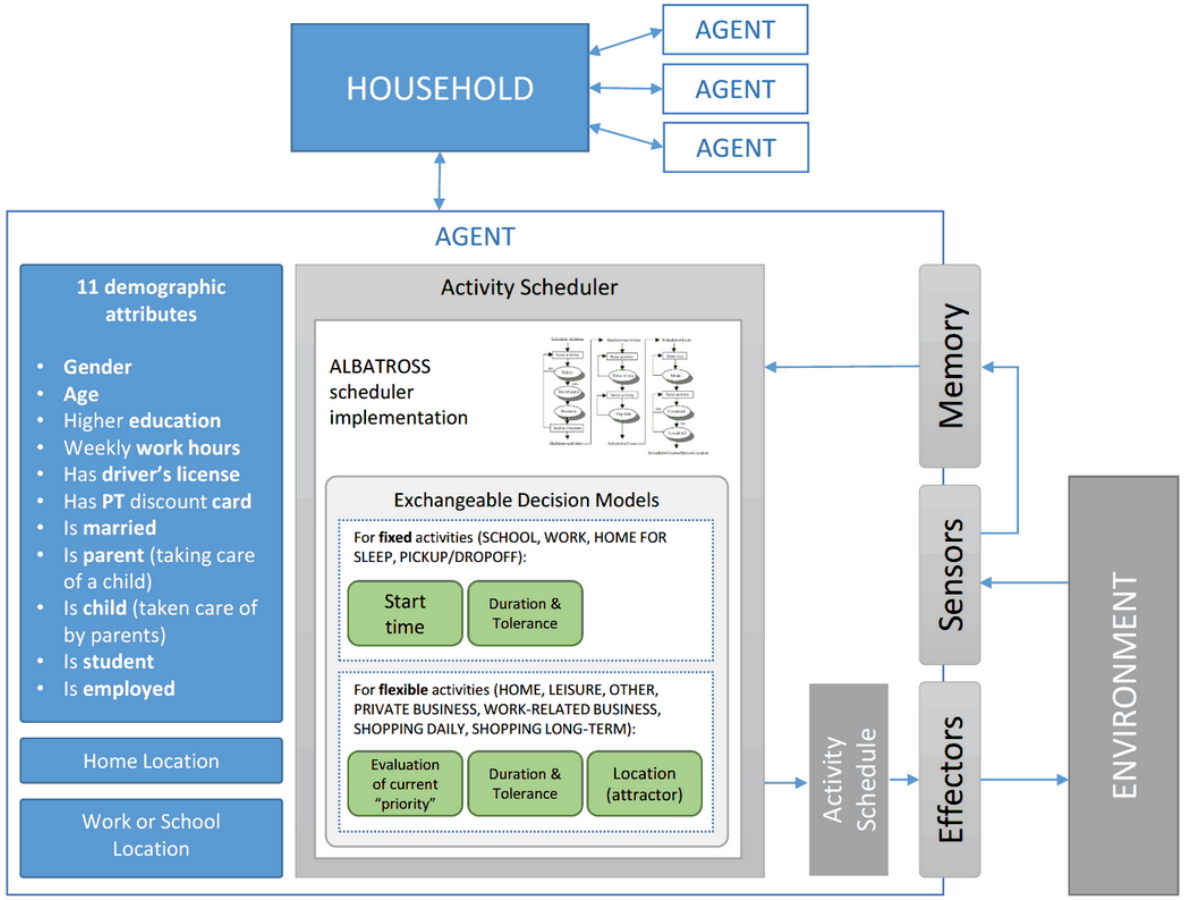


Comparison



Differences

Efficient travel time estimation for activity-based mobility models



Efficient travel time calculation essential for activity **location choice** and **mode choice** in activity models.

Millions+ of travel time estimations required for a single simulation runs.

Future Work

Towards **full multimodality**: P+R and B+R

Additional criteria for **cycling accessibility** (comfort and safety)

Supercomputer deployment + national **scale up**

Closed-loop Integration with public transport network (and timetable) **design process**

Incorporation of **accurate demand** data

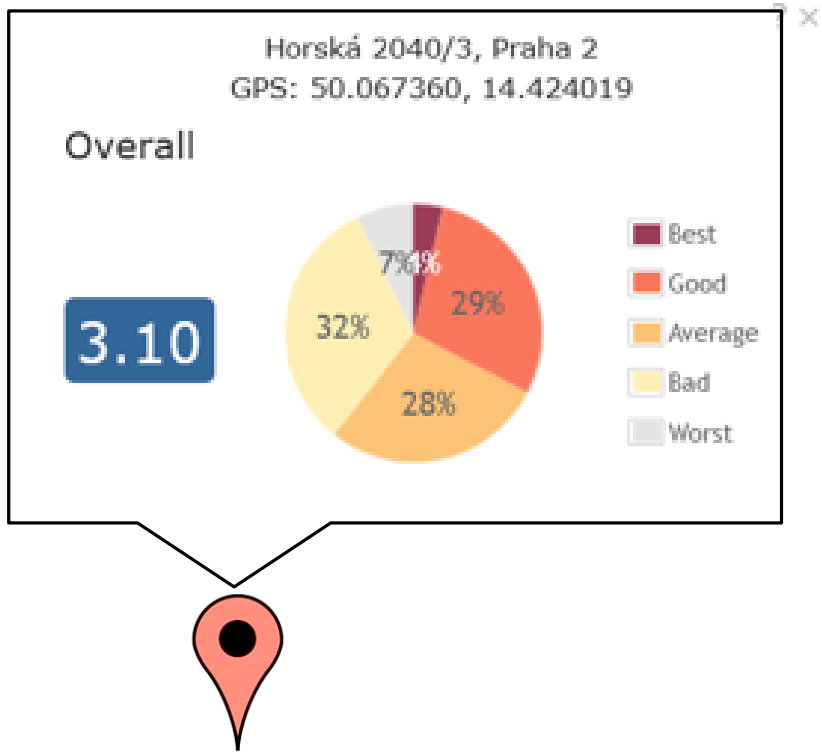
Conclusions

Efficient yet accurate calculation of transport accessibility is a **computationally challenging** problem.

We have developed an **efficient** method utilizing a **fine-grained** model of the **transport network**.

We support accessibility calculation for **multiple transport modes**.

The method can be **easily integrated** into a range of smart-city application in transport and urban planning.



Horská 3, Praha 2

➔ http://transportanalyser.com
http://dopravnidostupnost.cz